

**ΥΠΟΕΡΓΟ 3 «ΔΡΑΣΕΙΣ ΒΕΛΤΙΩΣΗΣ ΠΟΙΟΤΗΤΑΣ ΕΠΙΜΟΡΦΩΤΙΚΩΝ  
ΠΡΟΓΡΑΜΜΑΤΩΝ»**

**της Πράξης «ΔΡΑΣΕΙΣ ΣΥΝΕΧΙΖΟΜΕΝΗΣ ΚΑΤΑΡΤΙΣΗΣ 2014-2018»  
κωδ. ΟΠΣ 5000245**

**ΤΙΤΛΟΣ ΠΡΟΓΡΑΜΜΑΤΟΣ: ΔΙΑΧΕΙΡΙΣΗ LINUX SERVER**

**ΕΚΠΑΙΔΕΥΤΙΚΟ ΥΛΙΚΟ**

**Κωδικός εκπαιδευτικού υλικού:**

**Κωδικός Πιστοποίησης προγράμματος: 618**

**ΥΠΟΕΡΓΟ 3 «ΔΡΑΣΕΙΣ ΒΕΛΤΙΩΣΗΣ ΠΟΙΟΤΗΤΑΣ ΕΠΙΜΟΡΦΩΤΙΚΩΝ  
ΠΡΟΓΡΑΜΜΑΤΩΝ»**

**ΤΙΤΛΟΣ ΠΡΟΓΡΑΜΜΑΤΟΣ:**

**ΔΙΑΧΕΙΡΙΣΗ LINUX SERVER**

**ΟΜΑΔΑ ΕΡΓΑΣΙΑΣ**

**Μέλη Ομάδας**

**Συντονίστρια:** **ΧΑΡΑΛΑΜΠΙΑ ΔΟΥΛΗ** (κωδ. Ο.Π.Σ.: 000570), Υπεύθυνη Προγραμμάτων  
ΙΝ.ΕΠ.

**Συγγραφείς:** **Δρ. ΧΡΗΣΤΟΣ ΓΡΟΜΠΑΝΟΠΟΥΛΟΣ** (κωδ. Ο.Π.Σ.: 011096), ΕΔΙΠ  
Πανεπιστήμιο Δυτικής Μακεδονίας.

**ΒΑΣΙΛΕΙΟΣ ΜΑΤΣΟΥΚΑΣ** (κωδ. Ο.Π.Σ.: 013355), Υπεύθυνος Σπουδών και  
Έρευνας Π.ΙΝ.ΕΠ.Θ./Ε.Κ.Δ.Δ.Α.

**Δρ. ΒΑΣΙΛΕΙΟΣ ΣΤΕΦΑΝΙΔΗΣ** (κωδ. Ο.Π.Σ.: 019938), Καθηγητής  
Πληροφορικής Δευτεροβάθμιας Εκπαίδευσης.

**Αξιολογητές:** **ΑΝΑΣΤΑΣΙΑ ΠΑΠΑΣΤΥΛΙΑΝΟΥ** (κωδ. Ο.Π.Σ.: 001795), Υπεύθυνη Σπουδών  
και Έρευνας Ε.Σ.Δ.Δ.Α.

**Δρ. ΔΗΜΗΤΡΙΟΣ ΚΑΡΑΠΙΠΕΡΗΣ** (κωδ. Ο.Π.Σ.: 007210), Στέλεχος Δήμου  
Θέρμης, Τμήμα Μηχανοργάνωσης.



# ΔΙΑΧΕΙΡΙΣΗ LINUX SERVER

ΕΚΠΑΙΔΕΥΤΙΚΟ ΥΛΙΚΟ ΕΠΙΜΟΡΦΩΤΙΚΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ

ΧΡΗΣΤΟΣ ΓΡΟΜΠΑΝΟΠΟΥΛΟΣ, ΒΑΣΙΛΕΙΟΣ ΜΑΤΣΟΥΚΑΣ, ΒΑΣΙΛΕΙΟΣ ΣΤΕΦΑΝΙΔΗΣ



Ε.Π.  
**ΜΕΤΑΡΡΥΘΜΙΣΗ  
ΔΗΜΟΣΙΟΥ  
ΤΟΜΕΑ**  
NSRF



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

## Περιεχόμενα

|       |  |    |
|-------|--|----|
| 1     | Εγκατάσταση Linux Server .....   | 13 |
| 1.1   | Εισαγωγή στο πρόγραμμα .....   | 13 |
| 1.2   | Βασικά καθήκοντα ενός διαχειριστή συστήματος .....                                   | 14 |
| 1.3   | Εγκατάσταση Linux Server .....   | 15 |
| 1.3.1 | Επιλογές εγκατάστασης .....  | 18 |
| 1.4   | Εκκίνηση, τερματισμός .....  | 20 |
| 1.4.1 | Διαδικασία εκκίνησης .....   | 20 |
| 1.4.2 | Επανεκκίνηση και τερματισμός .....   | 22 |
| 1.4.3 | Διαχείριση του συστήματος κατά την εκκίνηση .....                                    | 24 |
| 1.4.4 | Εκκίνηση σε κατάσταση επίλυσης προβλήματος .....                                     | 25 |
| 2     | Προγραμματισμός στο φλοιό Bash .....   | 28 |
| 2.1   | Εισαγωγή στη δημιουργία σεναρίων (script) του φλοιού bash .....                      | 28 |
| 2.2   | Στοιχεία του προγραμματισμού σε bash .....   | 29 |
| 2.2.1 | Μεταβλητές τοπικές / περιβάλλοντος .....   | 29 |
| 2.2.2 | Παράμετροι θέσης .....   | 33 |
| 2.2.3 | Πίνακες / συναρτήσεις .....  | 35 |
| 2.3   | Δομές προγραμματισμού στο bash .....   | 39 |
| 2.3.1 | Δομή επιλογής .....  | 40 |
| 2.3.2 | Δομή επανάληψης .....  | 45 |
| 2.4   | Μελέτη περιπτώσεων ανάλυσης αρχείων κειμένου με χρήση εργαλείων .....                | 47 |
| 2.4.1 | Κανονικών εκφράσεων .....  | 47 |
| 2.4.2 | Ο επεξεργαστής ροών κειμένου sed .....   | 51 |
| 2.4.3 | Ο επεξεργαστής κειμένου awk .....  | 57 |
| 2.5   | Μελέτη περιπτώσεων scripts που επιλύουν καθημερινά προβλήματα ενός διαχειριστή ..... | 62 |
| 3     | Βασικές λειτουργίες διαχείρισης ενός Linux Server .....                              | 67 |
| 3.1   | Διαχείριση χρηστών / ομάδων (groups) .....   | 67 |
| 3.1.1 | Εισαγωγή, βασικά αρχεία για χρήστες / ομάδες .....                                   | 67 |
| 3.1.2 | Δημιουργία, διαμόρφωση διαγραφή λογαριασμού .....                                    | 70 |
| 3.1.3 | Χρήση γραφικών εργαλείων για τη διαχείριση χρηστών .....                             | 75 |
| 3.1.4 | Διαχείριση συνθηματικών .....  | 76 |
| 3.1.5 | Δικαιώματα πρόσβασης μέσω group .....  | 77 |
| 3.2   | Διαχείριση πακέτων λογισμικών (packages) .....                                       | 79 |



|       |   |     |
|-------|---|-----|
| 3.2.1 | Έλεγχος, ενημέρωση, αναζήτηση και εγκατάσταση πακέτων λογισμικών .....                          | 80  |
| 3.2.2 | Χρήση γραφικών εργαλείων για τη διαχείριση πακέτων λογισμικών.....                              | 85  |
| 3.2.3 | Διαχείριση repositories .....   | 86  |
| 3.3   | Διαχείριση αρχείων / καταλόγων.....   | 88  |
| 3.3.1 | Αντίγραφα ασφαλείας .....   | 88  |
| 3.3.2 | Διαχείριση αντιγράφων ασφαλείας.....  | 90  |
| 3.3.3 | Η δομή του ριζικού δέντρου στο Linux .....  | 96  |
| 3.3.4 | Προσάρτηση συστήματος αρχείων.....  | 98  |
| 3.3.5 | Διαχείριση δικαιωμάτων αρχείων / καταλόγων.....   | 109 |
| 4     | Εγκατάσταση - Διαχείριση Βασικών Εξυπηρετητών - Διαχείριση Υπηρεσιών και System Monitoring..... | 114 |
| 4.1   | Χρήσιμες Εντολές Διαχείρισης .....  | 114 |
| 4.2   | Εγκατάσταση και Διαχείριση X Window Server .....  | 117 |
| 4.2.1 | Εγκατάσταση του X όταν δεν υπάρχει στη διανομή.....   | 118 |
| 4.2.2 | Απενεργοποίηση του X (τεχνική runlevels).....   | 120 |
| 4.2.3 | Πρωτόκολλα απομακρυσμένης πρόσβασης στον X .....  | 122 |
| 4.2.4 | Εγκατάσταση X Window Manager (WM) .....   | 133 |
| 4.3   | Εγκατάσταση και Διαχείριση Samba Server .....   | 134 |
| 4.3.1 | Εγκατάσταση.....  | 135 |
| 4.3.2 | Παραμετροποίηση .....   | 135 |
| 4.3.3 | Χρήση .....   | 139 |
| 5     | Διαχείριση Υπηρεσιών και System Monitoring.....   | 142 |
| 5.1   | Διεργασίες – Υπηρεσίες - Deamons - Sockets.....   | 142 |
| 5.2   | Καταστάσεις διεργασιών .....  | 143 |
| 5.3   | Διεργασίες Συστήματος και Εργαλεία Παρακολούθησης Διεργασιών (process monitoring).....          | 144 |
| 5.3.1 | Εντολή ps .....   | 144 |
| 5.3.2 | Εντολή pstree.....  | 147 |
| 5.3.3 | Εντολή top .....  | 147 |
| 5.3.4 | Εντολή htop .....   | 149 |
| 5.3.5 | Διεργασίες και Τερματικό .....  | 150 |
| 5.3.6 | Τερματισμός διεργασιών (εντολή Kill).....   | 151 |
| 5.4   | Εργαλεία Παρακολούθησης και Διαχείρισης Συστήματος .....  | 152 |
| 5.4.1 | Cockpit.....  | 153 |

|       |   |     |
|-------|---|-----|
| 5.4.2 | Webmin .....  | 158 |
| 6     | Δικτυακές Υπηρεσίες .....   | 164 |
| 6.1   | Το Μοντέλο Αναφοράς O.S.I. ....                                   | 165 |
| 6.2   | IP protocol .....   | 167 |
| 6.3   | Διευθυνσιοδότηση τοπικών δικτύων .....                            | 169 |
| 6.4   | Δικτυακές υπηρεσίες - sockets - ports .....                       | 171 |
| 6.5   | Βασικοί Εξυπηρετητές δικτυακής δραστηριότητας.....                | 172 |
| 6.6   | Γραφικό εργαλείο διαχείρισης δικτυακών στοιχείων .....            | 172 |
| 6.7   | Command line και script Ethernet interface control .....          | 175 |
| 6.8   | Δικτυακά πρωτόκολλα .....   | 176 |
| 6.8.1 | Το πρωτόκολλο TCP .....   | 176 |
| 6.8.2 | Το πρωτόκολλο UDP .....   | 177 |
| 6.8.3 | Το πρωτόκολλο SNMP .....  | 178 |
| 7     | Ασφάλεια Δικτύου.....   | 183 |
| 7.1   | Ασφάλεια Δικτυακών Υπηρεσιών.....                                 | 184 |
| 7.1.1 | SELinux.....  | 184 |
| 7.1.2 | Firewall .....  | 188 |
| 7.1.3 | Port Forwarding .....   | 193 |
| 7.1.4 | Fail2Ban .....  | 197 |
| 7.1.5 | Kali.....   | 200 |
| 7.1.6 | Ενημερώσεις (Updates).....  | 205 |
| 7.2   | Εργαλεία παρακολούθησης δικτυακής δραστηριότητας .....            | 206 |
| 7.2.1 | netstat .....   | 206 |
| 7.2.2 | nmap .....  | 207 |
| 7.2.3 | zenmap .....  | 209 |
| 7.2.4 | Εργαλείο παρακολούθησης fedora .....                              | 212 |
| 7.2.5 | Αρχεία Καταγραφής (log files) .....                               | 214 |
| 7.2.6 | Cacti.....  | 217 |
| 7.2.7 | Cockpit.....  | 222 |
| 7.2.8 | Webmin .....  | 222 |
| 8     | Εγκατάσταση - Διαχείριση Εξυπηρετητών διαδικτυακών υπηρεσιών..... | 223 |
| 8.1   | Εγκατάσταση παραμετροποίηση και διαχείριση Proxy Server.....      | 223 |
| 8.1.1 | Διαχείριση Περιεχομένου.....                                      | 224 |
| 8.2   | Εγκατάσταση παραμετροποίηση και διαχείριση Apache Web Server..... | 227 |

|       |  |     |
|-------|--|-----|
| 8.2.1 | Κατάλογος-διαδρομή διαδικτυακού περιεχομένου .....           | 230 |
| 8.2.2 | Virtual Servers στον Apache .....                            | 230 |
| 8.2.3 | Ο Apache Web Server στο Webmin .....                         | 232 |
| 9     | Διαχείριση εικονικών μηχανών.....                            | 234 |
| 9.1   | Βασικές έννοιες .....  | 234 |
| 9.2   | Προετοιμασία για την υποστήριξη εικονικών μηχανών.....       | 236 |
| 9.3   | Δημιουργία και εκτέλεση μιας εικονικής μηχανής .....         | 238 |
| 9.3.1 | Δημιουργία εικονικής μηχανής με γραφικό τρόπο .....          | 238 |
| 9.3.2 | Δημιουργία εικονικής μηχανής από τη γραμμή εντολών .....     | 241 |
| 9.4   | Υποστήριξη δικτύου στις εικονικές μηχανές.....               | 242 |
| 9.5   | Διαχείριση των εικονικών μηχανών .....                       | 244 |
| 9.5.1 | Διαχείριση εικονικής μηχανής από το γραφικό περιβάλλον ..... | 244 |
| 9.5.2 | Διαχείριση εικονικής μηχανής από τη γραμμή εντολών .....     | 245 |
|       | Παραπομπές .....   | 247 |

## Πίνακας Εικόνων

|   |     |
|---|-----|
| Εικόνα 1- 1: Αρχική οθόνη Εγκατάστασης  | 16  |
| Εικόνα 1- 2: Γραφικό περιβάλλον εγκατάστασης  | 16  |
| Εικόνα 1- 3: Κεντρική οθόνη εγκατάστασης  | 17  |
| Εικόνα 1- 4: Επιλογή μέσου εγκατάστασης   | 18  |
| Εικόνα 1- 5: Καθορισμός διαμόρφωσης δίσκου  | 19  |
| Εικόνα 1- 6: Επιλογή εγκατάστασης λογισμικών  | 20  |
| Εικόνα 1- 7: Στάδια εκκίνησης του συστήματος  | 21  |
| Εικόνα 1- 8: Διαδικασία τερματισμού στο γραφικό περιβάλλον  | 23  |
| Εικόνα 1- 9: Ρύθμιση παραμέτρων του grub2 κατά την εκκίνηση   | 26  |
| <br>  |     |
| Εικόνα 2- 1: Η "οικογένεια" των εντολών if .....  | 40  |
| Εικόνα 2- 2: Παράδειγμα χρήσης της grep.....  | 48  |
| <br>  |     |
| Εικόνα 3- 1: Λειτουργία useradd .....   | 73  |
| Εικόνα 3- 2: Διαχείριση χρηστών με γραφικό τρόπο .....  | 75  |
| Εικόνα 3- 3: Προσθήκη χρήστη με γραφικό τρόπο .....   | 76  |
| Εικόνα 3- 4: Κατηγορίες εντολών του DNF.....  | 84  |
| Εικόνα 3- 5: Το εργαλείο "Software" για τη διαχείριση λογισμικών .....  | 86  |
| Εικόνα 3- 6: Δομή ριζικού δέντρου στο Linux.....  | 97  |
| Εικόνα 3- 7: Το γραφικό εργαλείο Disks .....  | 103 |
| Εικόνα 3- 8: Προσάρτηση δίσκων .....  | 105 |
| Εικόνα 3- 9: Logical Volume Manager.....  | 107 |
| <br>  |     |
| Εικόνα 4- 1: Αποτέλεσμα της εκτέλεσης της εντολής systemctl status της υπηρεσίας httpd .....  | 114 |
| Εικόνα 4- 2: Διακοπή και έλεγχος λειτουργίας της υπηρεσίας httpd.....   | 115 |
| Εικόνα 4- 3: Εμφάνιση των υπηρεσιών των θυρών που είναι ανοικτές στο firewall.....  | 116 |
| Εικόνα 4- 4: Οπτικοποίηση του X Window Server μαζί με τον X Window Manager .....  | 118 |
| Εικόνα 4- 5: Παράδειγμα παραμετροποίησης αρχείων του X Window Server.....   | 119 |
| Εικόνα 4- 6: Στιγμιότυπο εκτέλεσης της εντολής xrandr .....   | 120 |
| Εικόνα 4- 7: Εγκατάσταση και εικόνα του Vinagre κατά το άνοιγμα .....   | 123 |
| Εικόνα 4- 8: Παράθυρο σύνδεσης στην «Απομακρυσμένη Επιφάνεια Εργασίας» .....  | 125 |
| Εικόνα 4- 9: Παράθυρο σύνδεσης στον απομακρυσμένο RDP Server από Windows Client   | 125 |
| Εικόνα 4- 10: Διαμοιρασμένο γραφικό session από τον RDP Server σε Windows Client.....   | 126 |
| Εικόνα 4- 11: Παράθυρο σύνδεσης του Vinagre στον RDP Server .....   | 126 |
| Εικόνα 4- 12: Παράθυρο σύνδεσης στον απομακρυσμένο RDP Server από Linux Client ....   | 127 |
| Εικόνα 4- 13: Διαμοιρασμένο γραφικό session από τον RDP Server σε Linux Client .....  | 127 |
| Εικόνα 4- 14: Δύο συνδέσεις VNC στο ίδιο Display (1) ταυτόχρονα και από Windows και από Linux (UltraVNC και Vinagre αντίστοιχα). Βλέπουν ακριβώς το ίδιο..... | 131 |
| Εικόνα 4- 15: Οθόνη σύνδεσης στο Display 1 στον 192.168.1.11 Linux Fedora Server με VNC από Client σε Windows .....   | 131 |
| Εικόνα 4- 16: Οθόνη με το session στο Display 1 στον 192.168.1.11 Linux Fedora Server με VNC από Client σε Windows.....                                       | 131 |

|   |     |
|---|-----|
| Εικόνα 4- 17: Οθόνη διασύνδεσης του remote desktop viewer (Vinagre) στον Server 192.168.1.11 στο DISPLAY 1 .....      | 132 |
| Εικόνα 4- 18: Οθόνη login του remote desktop viewer (Vinagre).....  | 132 |
| Εικόνα 4- 19: Οθόνη με το session στο Display 1 στον 192.168.1.11 Linux Fedora Server με VNC από Client σε Linux..... | 132 |
| Εικόνα 4- 20: Εικόνα από τα Desktops διαφόρων X Window Managers.....  | 134 |
| Εικόνα 4- 21: Εμπορικό σήμα του Samba Server.....   | 134 |
| Εικόνα 4- 22: Οθόνη των Windows που βλέπει ο χρήστης όταν ο Samba είναι ενεργοποιημένος.....                          | 139 |
| Εικόνα 4- 23: Εικόνα από τον explorer των Windows με το όνομα και το περιεχόμενο του διαμοιραζόμενου φακέλου .....    | 140 |
| Εικόνα 4- 24: Συνολική εικόνα διαμοιραζόμενου αρχείου και φακέλου σε Windows .....                                    | 140 |
| Εικόνα 4- 25: Συνολική εικόνα διαμοιραζόμενου αρχείου και φακέλου σε Linux.....                                       | 141 |
| <br>  |     |
| Εικόνα 5- 1: Έξοδος της εντολής ps.....   | 145 |
| Εικόνα 5- 2: Εμφάνιση διεργασιών χρήστη με την ps -aux.....   | 146 |
| Εικόνα 5- 3: Εμφάνιση διεργασιών χρήστη με την ps -ejf.....   | 146 |
| Εικόνα 5- 4: Έξοδος της εντολής pstree .....  | 147 |
| Εικόνα 5- 5: Εικόνα από την εντολή top.....   | 148 |
| Εικόνα 5- 6: Εικόνα από την εντολή top μετά από μερικά δευτερόλεπτα .....   | 148 |
| Εικόνα 5- 7: Ταξινόμηση της top ως προς το ποσοστό χρήσης της CPU.....  | 148 |
| Εικόνα 5- 8: ταξινόμηση της top ως προς το ποσοστό χρήσης της Μνήμης.....   | 149 |
| Εικόνα 5- 9: Διεπαφή της εφαρμογής htop.....  | 150 |
| Εικόνα 5- 10: Παράδειγμα σύνδεσης διεργασιών - τερματικών.....  | 151 |
| Εικόνα 5- 11: Εισαγωγική οθόνη του Cockpit.....   | 154 |
| Εικόνα 5- 12: Η οθόνη system του cockpit.....   | 154 |
| Εικόνα 5- 13: Η οθόνη logs του cockpit.....   | 155 |
| Εικόνα 5- 14: Η οθόνη storage του cockpit .....   | 155 |
| Εικόνα 5- 15: Η οθόνη networking του cockpit .....  | 156 |
| Εικόνα 5- 16: Η οθόνη containers του cockpit .....  | 156 |
| Εικόνα 5- 17: Η οθόνη accounts του cockpit .....  | 157 |
| Εικόνα 5- 18: Η οθόνη services του cockpit .....  | 157 |
| Εικόνα 5- 19: Η οθόνη terminal του cockpit.....   | 158 |
| Εικόνα 5- 20: Η εισαγωγική οθόνη του webmin .....   | 160 |
| Εικόνα 5- 21: Το Dashboard του webmin με πληροφορίες του Server .....   | 160 |
| Εικόνα 5- 22: Το Dashboard του webmin με πληροφορίες του Server .....   | 161 |
| Εικόνα 5- 23: Το Dashboard του webmin με πληροφορίες του Server .....   | 161 |
| Εικόνα 5- 24: Τα μενού System και Un-used Modules του webmin.....   | 162 |
| Εικόνα 5- 25: Τα μενού Webmin και Servers του webmin.....   | 162 |
| Εικόνα 5- 26: Η οθόνη παραμετροποίησης του Bind DNS Server στο webmin .....   | 163 |
| Εικόνα 5- 27: Η οθόνη Backup των διαφόρων εφαρμογών και υπηρεσιών στο webmin ....                                     | 163 |
| <br>  |     |
| Εικόνα 6- 1: Τα επίπεδα του μοντέλου OSI .....  | 166 |
| Εικόνα 6- 2: Απεικόνιση των δικτυακών διεπαφών του Server .....   | 173 |

|  |     |
|--|-----|
| Εικόνα 6- 3: Διαχείριση δικτυακής διεπαφής Server.....   | 173 |
| Εικόνα 6- 4: Προσδιορισμός απόδοσης IP αυτόματα, μέσω DHCP .....                                 | 174 |
| Εικόνα 6- 5: Προσδιορισμός απόδοσης IP μη αυτόματα, «με το χέρι» .....                           | 174 |
| Εικόνα 6- 6: Αρχιτεκτονική SNMP - εικόνα από το (CONTA_Lab) .....                                | 179 |
| Εικόνα 6- 7: Απεικόνιση πληροφορίας που μεταφέρεται με το πρωτόκολλο UDP .....                   | 180 |
| <br>   |     |
| Εικόνα 7- 1: Λίστα διακοπών SELinux.....   | 185 |
| Εικόνα 7- 2: Διακόπτες που περιέχουν την υπηρεσία httpd.....                                     | 185 |
| Εικόνα 7- 3: Ενεργοποίηση και απενεργοποίηση του SELinux.....                                    | 187 |
| Εικόνα 7- 4: Διαδικασία Relabeling μετά την ενεργοποίηση του SELinux .....                       | 187 |
| Εικόνα 7- 5: Γραφικό περιβάλλον διαχείρισης του εξελιγμένου Firewall .....                       | 189 |
| Εικόνα 7- 6: Διαδρομή για το περιβάλλον διαχείρισης του Firewall στον xfce .....                 | 190 |
| Εικόνα 7- 7: Ενεργοποίηση υπηρεσίας μόνιμα ή για το runtime session.....                         | 190 |
| Εικόνα 7- 8: Ενεργοποίηση πόρτας μόνιμα ή για το runtime session .....                           | 191 |
| Εικόνα 7- 9: Διαχείριση του Firewall από το διαχειριστικό περιβάλλον Webmin .....                | 193 |
| Εικόνα 7- 10: Διαδικασία Masquerading .....  | 194 |
| Εικόνα 7- 11: Port Forwarding .....  | 195 |
| Εικόνα 7- 12: Port Forwarding παράθυρο διαλόγου add .....  | 196 |
| Εικόνα 7- 13: Port Forwarding παράθυρο διαλόγου add συμπληρωμένο .....                           | 196 |
| Εικόνα 7- 14: Port Forwarding με την αντίστοιχη εγγραφή .....                                    | 197 |
| Εικόνα 7- 15: Η διανομή Kali .....   | 200 |
| Εικόνα 7- 16: Εφαρμογή επίθεσης σε τοπικό δίκτυο.....  | 202 |
| Εικόνα 7- 17: Εφαρμογή επίθεσης σε πρωτόκολλο σε συγκεκριμένη IP.....                            | 202 |
| Εικόνα 7- 18: Υπολογιστικά συστήματα πιθανές επιθέσεις σε πρωτόκολλα και πιθανά exploits .....   | 203 |
| Εικόνα 7- 19: Εργαλεία DOS attack από το Kali .....  | 204 |
| Εικόνα 7- 20: Παρακολούθηση μιας επίθεσης DOS από το Cockpit .....                               | 204 |
| Εικόνα 7- 21: Η Εφαρμογή gnome-software .....  | 205 |
| Εικόνα 7- 22: Έξοδος της εντολής netstat με παράμετρο -an .....                                  | 207 |
| Εικόνα 7- 23: Έξοδος της εντολής nmap .....  | 209 |
| Εικόνα 7- 24: Δικτυακή απεικόνιση του Server με το Zenmap .....                                  | 210 |
| Εικόνα 7- 25: Profiles εκτέλεσης του Zenmap .....  | 210 |
| Εικόνα 7- 26: Πόρτες και υπηρεσίες με χαρτογράφηση από το Zenmap.....                            | 211 |
| Εικόνα 7- 27: Τοπολογία δικτύου με χαρτογράφηση από το Zenmap .....                              | 211 |
| Εικόνα 7- 28: Εικόνα των διεργασιών από το gnome-system-monitor .....                            | 212 |
| Εικόνα 7- 29: Εικόνα διαφόρων πόρων συστήματος από το gnome-system-monitor.....                  | 213 |
| Εικόνα 7- 30: Εικόνα του συστήματος αρχείων από το gnome-system-monitor.....                     | 213 |
| Εικόνα 7- 31: Οθόνη σύνδεσης Cacti .....   | 217 |
| Εικόνα 7- 32: Απεικόνιση του device που θέλει να παρακολουθεί ο διαχειριστής στο Cacti .....     | 218 |
| Εικόνα 7- 33: Παραμετροποίηση του device που θέλει να παρακολουθεί ο διαχειριστής στο Cacti..... | 218 |
| Εικόνα 7- 34: Προσδιορισμός του Data Collector στο Cacti.....                                    | 219 |
| Εικόνα 7- 35: Προσδιορισμός Μεθόδων Εισαγωγής Δεδομένων στο Cacti .....                          | 219 |

|  |     |
|--|-----|
| Εικόνα 7- 36: Στοιχεία για το networking στο Cacti.....  | 220 |
| Εικόνα 7- 37: Προσδιορισμός Διαθέσιμων Γραφημάτων στο Cacti .....                              | 220 |
| Εικόνα 7- 38: Επιλογή δημιουργίας διαγραμμάτων στο Cacti .....                                 | 220 |
| Εικόνα 7- 39: Παρακολούθηση του Disk Space του Linux Server στο Cacti.....                     | 221 |
| Εικόνα 7- 40: Παρακολούθηση του Disk Space του Linux Server στο Cacti .....                    | 221 |
| Εικόνα 7- 41: Παρακολούθηση του Load Average του Linux Server στο Cacti .....                  | 221 |
| Εικόνα 7- 42: Παρακολούθηση της Χρήσης Μνήμης του Linux Server στο Cacti.....                  | 222 |
| <br>   |     |
| Εικόνα 8- 1: Εμπορικό σήμα του squid .....   | 223 |
| Εικόνα 8- 2: Ορισμός Proxy στον Browser του χρήστη .....                                       | 226 |
| Εικόνα 8- 3: Μήνυμα αποκλεισμού διεύθυνσης URL από τον squid proxy Server .....                | 227 |
| Εικόνα 8- 4: Αναζήτηση ιστοσελίδας με πλήρη διαδρομή.....                                      | 229 |
| Εικόνα 8- 5: Αναζήτηση ιστοσελίδας μόνο με κατάλογο .....                                      | 229 |
| Εικόνα 8- 6: Διαδικτυακή διεύθυνση στην οποία βρίσκεται σελίδα που φιλοξενεί ο Apache<br>..... | 230 |
| Εικόνα 8- 7: Διαδικτυακή διεύθυνση με τη χρήση του λεκτικού localhost .....                    | 230 |
| Εικόνα 8- 8: Διαδικτυακή διεύθυνση με τη χρήση domain .....                                    | 230 |
| Εικόνα 8- 9: URL εικονικού Server στον Apache.....   | 231 |
| Εικόνα 8- 10: URL εικονικού Server στον Apache.....  | 231 |
| Εικόνα 8- 11: Διαχείριση του Apache μέσα από το διαχειριστικό εργαλείο Webmin .....            | 232 |
| Εικόνα 8- 12: Virtual Hosts στον Apache έτσι όπως φαίνονται από το Webmin .....                | 232 |
| <br>   |     |
| Εικόνα 9- 1: Χρήση Εικονικών Μηχανών στους Servers .....                                       | 235 |
| Εικόνα 9- 2: Τύποι Hypervisor .....  | 236 |
| Εικόνα 9- 3: Το εργαλείο Virtual Machine Manager .....   | 238 |
| Εικόνα 9- 4: Βήματα 1 & 2 του οδηγού δημιουργίας VM .....                                      | 239 |
| Εικόνα 9- 5: Βήματα 3 & 4 του οδηγού δημιουργίας VM .....                                      | 239 |
| Εικόνα 9- 6: Βήμα 5 του οδηγού δημιουργίας VM.....   | 240 |
| Εικόνα 9- 7: Λειτουργία εικονικής μηχανής .....  | 241 |
| Εικόνα 9- 8: Ρύθμιση εικονικών δικτύων .....   | 242 |
| Εικόνα 9- 9: Πληροφορίες υλικού εικονικής μηχανής .....  | 245 |

## Πίνακας Πινάκων

|   |     |
|---|-----|
| Πίνακας 2- 1: Βασικοί έλεγχοι της εντολής test .....                      | 42  |
| Πίνακας 2- 2: Έλεγχοι της εντολής test σε αρχεία .....                    | 45  |
| Πίνακας 2- 3: Οι μεταχαρακτήρες wildcards .....                           | 49  |
| Πίνακας 2- 4: Οι μεταχαρακτήρες modifier .....                            | 50  |
| Πίνακας 2- 5: Οι μεταχαρακτήρες ancor .....                               | 51  |
| <br>  |     |
| Πίνακας 3- 1: Παράμετροι της εντολής useradd .....                        | 71  |
| Πίνακας 3- 2: Παράμετροι της εντολής usermod .....                        | 73  |
| Πίνακας 3- 3: Παράμετροι της εντολής chage .....                          | 77  |
| Πίνακας 3- 4: Παράμετροι της εντολής tar .....                            | 90  |
| Πίνακας 3- 5: Παράμετροι εμφάνισης της εντολής tar .....                  | 91  |
| Πίνακας 3- 6: Παράμετροι της εντολής rsync .....                          | 93  |
| Πίνακας 3- 7: Ειδικές παράμετροι της εντολής rsync .....                  | 94  |
| Πίνακας 3- 8: Παράμετροι της εντολής dd .....                             | 95  |
| Πίνακας 3- 9: Τύποι αρχείων στο Linux .....                               | 98  |
| Πίνακας 3- 10: Εντολές του εργαλείου parted .....                         | 100 |
| Πίνακας 3- 11: Εντολές διαχείρισης του LVM .....                          | 107 |
| <br>  |     |
| Πίνακας 4- 1: Runlevels και λειτουργίες (WikiRunlevels, n.d.) .....       | 121 |
| <br>  |     |
| Πίνακας 5- 1: Κωδικοί και σήματα της εντολής Kill .....                   | 152 |
| <br>  |     |
| Πίνακας 6- 1: Γνωστές Πόρτες Λειτουργικού Συστήματος (Common Ports) ..... | 172 |
| Πίνακας 6- 2: Μοντέλο Διαστρωμάτωσης (ή Αναφοράς) του Internet .....      | 177 |



# 1 Εγκατάσταση Linux Server

## 1.1 Εισαγωγή στο πρόγραμμα

Η τεχνολογία **Server – Client** (Εξυπηρετητή –πελάτη) χρησιμοποιείται σε διάφορες περιοχές της πληροφορικής όπως το διαδίκτυο, η αποστολή email, η εκτέλεση προγραμμάτων κ.α.. Σύμφωνα με αυτή την τεχνολογία (Wikipedia:Client-Server, n.d.) υπάρχει ένας μικρός σε υπολογιστική ισχύ υπολογιστής πελάτης (client) που αποστέλλει αιτήματα σε έναν ισχυρό υπολογιστή εξυπηρετητή (Server). Ο Server επεξεργάζεται τις αιτήσεις πολλών clients ταυτόχρονα και τους επιστρέφει πίσω απαντήσεις. Ο ίδιος όρος Server, χρησιμοποιείται επίσης και για να περιγράψει τα προγράμματα που εκτελούνται στον υπολογιστή Server τα οποία έχουν σαν σκοπό την επεξεργασία των αιτήσεων των client και τη δημιουργία απαντήσεων.

Για τη λειτουργία ενός οποιαδήποτε υπολογιστικού συστήματος, ακόμα και ενός Server, είναι απαραίτητη η εγκατάσταση σε αυτό ενός **λειτουργικού συστήματος (operating system)**. Αυτή τη στιγμή, υπάρχουν αρκετά λειτουργικά συστήματα όπως: τα Windows, το UNIX, το Mac OS κτλ., καθένα από τα οποία παρουσιάζει τα δικά του πλεονεκτήματα και μειονεκτήματα. Το **Linux** είναι ο πυρήνας (**kernel**) ενός λειτουργικού συστήματος που εκτελείται σε επεξεργαστές αρχιτεκτονικής Intel και έχει σαν βασικό συστατικό τη σύνδεση του λογισμικού με το υλικό του υπολογιστή. Η χρήση του Linux σε υπολογιστές Server παρουσιάζει αρκετά πλεονεκτήματα, με ένα πολύ σημαντικό από αυτά είναι το ότι μπορεί να χρησιμοποιηθεί χωρίς κόστος απόκτησης (Jayden, 2018). Το λειτουργικό σύστημα Linux εκτός από τον πυρήνα περιέχει και μία ποικιλία άλλων εφαρμογών που ρυθμίζουν διάφορα θέματα όπως: την επικοινωνία με το χρήστη, τον τρόπο εγκατάστασης καινούργιων εφαρμογών κ.α. Ο συνδυασμός ενός πυρήνα Linux με συγκεκριμένες περιφερειακές εφαρμογές ονομάζεται **διανομή (distribution)**. Υπάρχουν αρκετές διανομές του Linux όπως οι Ubuntu, Red Hat, Open Suse, Debian κτλ. Η επιλογή μιας διανομής για την υποστήριξη ενός συστήματος είναι μια αρκετά σημαντική απόφαση που πιθανώς να επηρεάσει τη δυνατότητα λειτουργίας του συστήματος. Μερικά κριτήρια για την επιλογή μιας διανομής για την υποστήριξη ενός υπολογιστή Server είναι τα παρακάτω (Morelo, 2020):

- Σταθερότητα. Οι εκδόσεις ενός λειτουργικού συστήματος για Server έχουν μεγαλύτερο χρόνο ζωής από τις εκδόσεις για κανονικούς χρήστες. Επιπρόσθετα πρέπει να ληφθεί υπόψη το γεγονός ότι ένα μηχάνημα Server θα είναι σχεδόν συνέχεια σε λειτουργία.
- Ασφάλεια. Θα πρέπει να υπάρχει δημιουργία και διανομή ενημερώσεων ασφαλείας καθώς και υποστηρίζονται πολιτικές που να περιορίζουν τις επιπτώσεις από την πιθανή εμφάνιση λαθών.
- Ευελιξία. Είναι απαραίτητη η δημιουργία και υποστήριξη μιας ποικιλίας λογισμικών που να μπορούν να επιτελέσουν το ρόλο που πρέπει να διαδραματίζει ο Server.
- Υποστήριξη. Να υπάρχει η δυνατότητα παροχής υποστήριξης για τη λειτουργία της διανομής καθώς επίσης και μια σημαντική εγκατεστημένη βάση για την αναζήτηση βοήθειας.



Η διανομή που επιλέχτηκε στο συγκεκριμένο πρόγραμμα είναι η **Fedora 31** η έκδοση **Server**. Ένα από τα πλεονεκτήματα της συγκεκριμένης διανομής είναι ότι βασίζεται στην RedHat η οποία έχει μεγάλη εγκατεστημένη βάση και υποστήριξη, αλλά επίσης τόσο το ίδιο το λειτουργικό σύστημα όσο και οι εφαρμογές που διατίθενται για αυτό, είναι δωρεάν και ανοιχτού κώδικα. Επιπρόσθετα, η διανομή της Fedora φροντίζει να καινοτομεί ενσωματώνοντας πρώτη όλες τις καινούργιες τεχνολογικές εξελίξεις.

## 1.2 Βασικά καθήκοντα ενός διαχειριστή συστήματος

Ένας **διαχειριστής συστήματος (system administrator / admin)** είναι επιφορτισμένος με μία πληθώρα εργασιών. Φυσικά όλες αυτές οι εργασίες δεν είναι ανάγκη να εκτελούνται από ένα άτομο όμως ο διαχειριστής του συστήματος θα πρέπει να γνωρίζει τη λειτουργία τους και να επιβλέπει και να επεμβαίνει σε περίπτωση που χρειάζεται. Οι εργασίες λοιπόν για τις οποίες είναι υπεύθυνος ο διαχειριστής είναι οι παρακάτω: (Nemeth, Snyder, Hein, & Whaley, 2017):

- **Διαχείριση των χρηστών.** Ο διαχειριστής είναι υπεύθυνος για τη δημιουργία λογαριασμών για τους νέους χρήστες καθώς και την κατάργηση των λογαριασμών των χρηστών που δεν είναι πλέον ενεργοί. Επιπρόσθετα, χειρίζεται και όλες τις ενέργειες σχετικά με τους λογαριασμούς που μπορούν να προκύψουν στο ενδιάμεσο (π.χ. ξεχασμένα passwords). Ακόμα όταν ένας χρήστης δεν είναι πλέον ενεργός στο σύστημα, ο διαχειριστής πρέπει να δημιουργήσει αντίγραφα ασφαλείας για όλα τα αρχεία του χρήστη και στη συνέχεια να τα διαγράψει από την τρέχουσα ρύθμιση του συστήματος.
- **Προσθήκη και κατάργηση υλικού (hardware).** Η αγορά καινούργιου υλικού ή η μετακίνηση υπάρχοντος από ένα μηχάνημα σε ένα άλλο, επιβάλλει τη ρύθμιση του συστήματος από τον διαχειριστή. Η διαδικασία αυτή μπορεί να είναι κάτι απλό, όπως η προσθήκη ενός εκτυπωτή ή κάτι πιο πολύπλοκο, όπως η προσθήκη μιας συστοιχίας δίσκων. Επιπλέον, η χρήση εικονικών μηχανών (virtualization) προσθέτει περισσότερη πολυπλοκότητα στις παραπάνω ενέργειες επειδή ο διαχειριστής πρέπει να αποφασίσει αν οι εικονικές μηχανές θα έχουν πρόσβαση και πώς σε αυτό το καινούργιο υλικό.
- **Εκτέλεση αντιγράφων ασφαλείας.** Η λήψη αντιγράφων ασφαλείας είναι μια από τις πιο σημαντικές δουλειές που πρέπει να εκτελεί ένας διαχειριστής μα συχνά υποτιμάται. Η δημιουργία αντιγράφων ασφαλείας μπορεί να αυτοματοποιηθεί, μα σε κάθε περίπτωση, ο διαχειριστής πρέπει να ελέγχει ότι λαμβάνονται σωστά αντίγραφα ασφαλείας και ότι είναι δυνατή η επαναφορά των δεδομένων από τα μέσα στα οποία έχει γίνει.
- **Εγκατάσταση και ενημέρωση λογισμικών.** Ένα από τα καθήκοντα του διαχειριστή είναι η εγκατάσταση των καινούργιων λογισμικών καθώς και η ενημέρωση του. Επιπρόσθετα, ο διαχειριστής πρέπει να ενημερώσει και τους χρήστες για την ύπαρξη των λογισμικών και τη σωστή χρήση τους. Υπάρχει ακόμα η περίπτωση ο οργανισμός να αναπτύσσει ο ίδιος λογισμικά τα οποία πρέπει να εγκαθίστανται και να ενημερώνονται σύμφωνα τις πολιτικές του οργανισμού. Στις περισσότερες περιπτώσεις αυτό σημαίνει ότι θα πρέπει να δοκιμάζονται προτού γίνουν διαθέσιμα σε όλο τον οργανισμό.

- **Επίβλεψη του συστήματος.** Η σωστή λειτουργία ενός συστήματος, ειδικά εάν αυτό είναι μεγάλου μεγέθους, απαιτεί συστηματική και αναλυτική επίβλεψη. Μερικά παραδείγματα επίβλεψης είναι η σωστή λειτουργία του ηλεκτρονικού ταχυδρομείου και του Web Server, η συνδεσιμότητα στο δίκτυο όλων των μηχανημάτων και η διαθεσιμότητα των πόρων. Η πραγματοποίηση της επίβλεψης συχνά επιτελείται με την ανάλυση των **αρχείων καταγραφής (log files)** τα οποία είναι αυτά που θα εμφανίσουν ένα επικείμενο πρόβλημα. Στο καθήκον του αυτό ο διαχειριστής μπορεί να βοηθηθεί από εργαλεία αυτοματισμού που μπορούν να επιτελέσουν αντί για αυτόν τη βαρετή και επαναλαμβανόμενη διαδικασία ελέγχου.

- **Επιδιόρθωση βλαβών.** Ακόμα και τα καλύτερα συστήματα εμφανίζουν βλάβες. Είναι καθήκον του διαχειριστή ενός συστήματος να κάνει διάγνωση του προβλήματος και αν μπορεί να το επιλύσει, αλλιώς να ζητήσει τη βοήθεια ενός ειδικού. Συχνά όμως η διάγνωση του προβλήματος είναι δυσκολότερη διαδικασία από την επίλυση του.

- **Διαχείριση τοπικών οδηγιών.** Κατά τη διάρκεια λειτουργίας του ένα σύστημα μεταβάλλεται έτσι ώστε να ικανοποιεί τις ανάγκες του οργανισμού από τον οποίο χρησιμοποιείται. Αφού ο διαχειριστής είναι υπεύθυνος για αυτές τις αλλαγές είναι υπεύθυνος και για την καταγραφή τους. Αυτό σημαίνει ότι, για παράδειγμα, διαχειριστής του συστήματος θα πρέπει να κρατάει αρχείο με τα καλώδια που έχει τοποθετήσει καθώς και με τις συσκευές που έχει εγκαταστήσει όπως και τα αντίγραφα ασφαλείας που έχει δημιουργήσει. Τέλος είναι σημαντικό για έναν διαχειριστή να καταγράψει τις πολιτικές που χρησιμοποιεί ο οργανισμός για την εκτέλεση ορισμένων ενεργειών σημαντικών στο σύστημα (π.χ. εγγραφή ενός καινούργιου χρήστη).

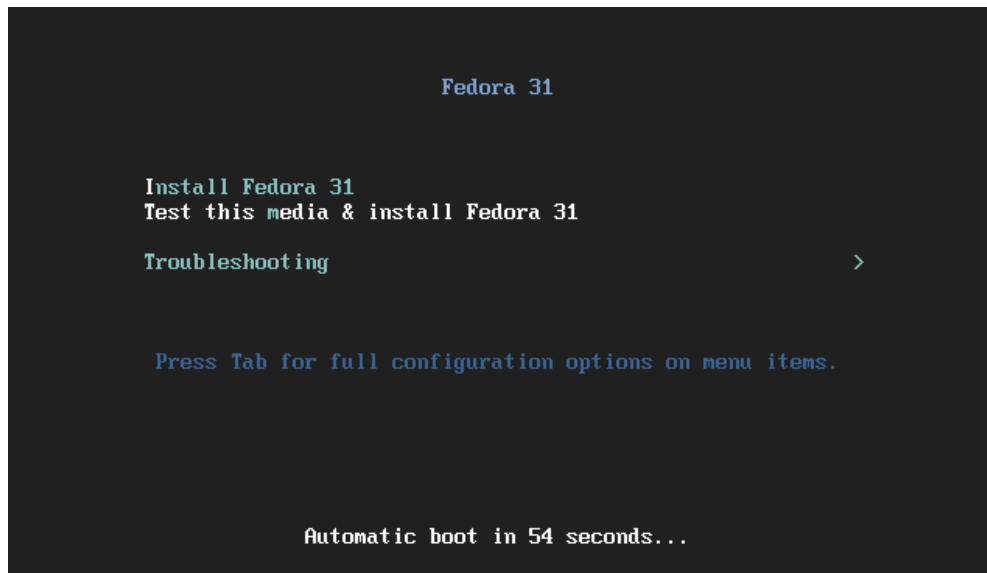
- **Εξασφάλιση της ασφάλειας του συστήματος.** Ο διαχειριστής ενός συστήματος πρέπει να εφαρμόζει μια πολιτική ασφάλειας και να ελέγχει περιοδικά ότι αυτή η πολιτική δεν παραβιάζεται. Σε ένα σύστημα χαμηλής ασφάλειας αυτό προϋποθέτει ότι ο διαχειριστής επιτελεί ένα βασικό σετ ελέγχων για μη εξουσιοδοτημένη πρόσβαση στο σύστημα. Σε ένα σύστημα υψηλής ασφάλειας η λειτουργία αυτή σημαίνει τη δημιουργία παγίδων δικτύου (network traps) και αναλυτική αναζήτηση σε λογισμικά ελέγχου (auditing programs).

Φυσικά στις παραπάνω εργασίες που πρέπει να επιτελεί ένας διαχειριστής προστίθεται και η καθημερινή υποστήριξη των χρηστών του συστήματος. Έτσι ο διαχειριστής θα κληθεί να δώσει λύση σε προβλήματα όπως η καταστροφή συσκευών (π.χ. καφές στο πληκτρολόγιο) ή η επεξήγηση της λειτουργίας του συστήματος («μα χτες δούλευε μια χαρά, σήμερα δεν μπορώ να κάνω τίποτα»).

### 1.3 Εγκατάσταση Linux Server

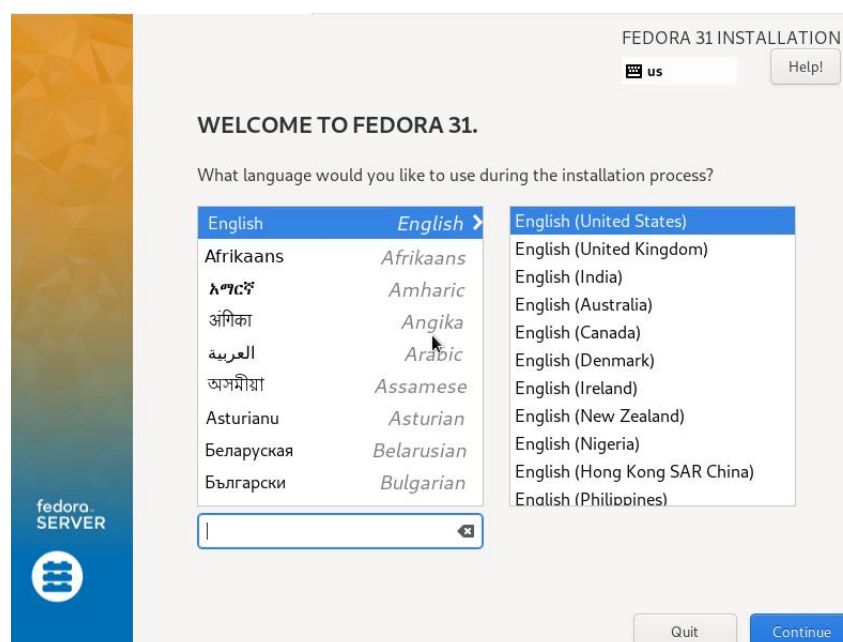
Το πρώτο βήμα για να εγκαταστήσουμε τη διανομή Fedora σε ένα υπολογιστικό σύστημα είναι η δημιουργία του κατάλληλου μέσου εγκατάστασης (cd, usb stick) (FedoraDocs-InstImage, n.d.). Στη συνέχεια, και αφού πρώτα έχουμε εισάγει το μέσο εγκατάστασης στον υπολογιστή πραγματοποιούμε εκκίνηση του υπολογιστή. Ιδιαίτερη προσοχή θα πρέπει να δοθεί έτσι ώστε να γίνει η κατάλληλη ρύθμιση στο λογισμικό εκκίνησης του υπολογιστή (BIOS/UEFI) ώστε αυτός να ξεκινήσει από το μέσο εγκατάστασης.

Η πρώτη εικόνα που εμφανίζεται κατά τη διαδικασία εγκατάστασης της διανομής Fedora είναι η Εικόνα 1- 1 που μας δίνει τρεις επιλογές. Η πρώτη ξεκινάει την εγκατάσταση άμεσα, ενώ η δεύτερη αφού πρώτα κάνει μια επαλήθευση του μέσου εγκατάστασης. Η τρίτη επιλογή “*Troubleshooting*” περιέχει διάφορες άλλες επιλογές (π.χ. έλεγχος των μνημών του συστήματος, επιδιόρθωση σφαλμάτων κτλ. ).



Εικόνα 1- 1: Αρχική οθόνη Εγκατάστασης

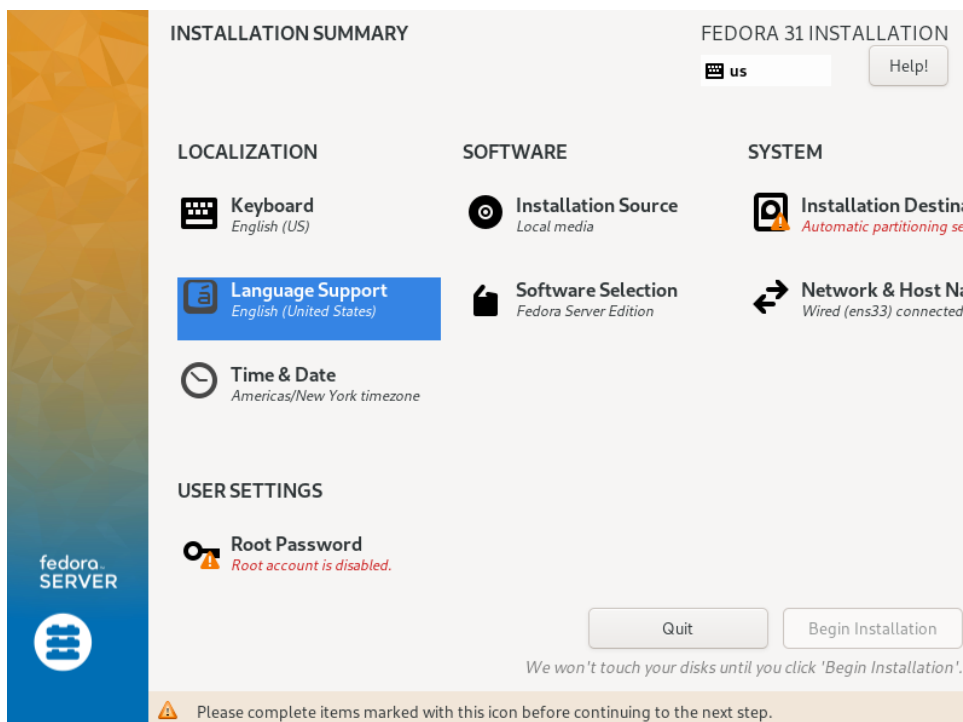
Στη συνέχεια ξεκινάει το γραφικό περιβάλλον εγκατάστασης της διανομής Fedora το οποίο ονομάζεται **Anaconda** (FedoraDocs-Install, n.d.) εμφανίζοντας την Εικόνα 1- 2. Στο παράθυρο της Εικόνα 1- 1 δίνεται η δυνατότητα επιλογής της γλώσσας εγκατάστασης στην οποία θα γίνει η εγκατάσταση η οποία καθορίζει και την επιλογή των τοπικών παραμέτρων (μονάδες μέτρησης, νόμισμα, κτλ.). Η συνέχιση της εγκατάστασης πραγματοποιείται με την επιλογή του πλήκτρου “*Continue*”.



Εικόνα 1- 2: Γραφικό περιβάλλον εγκατάστασης

Η διαδικασία της εγκατάστασης συνεχίζεται με την παρουσίαση της Εικόνα 1- **3Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε..** Αυτή παρουσιάζει το κύριο παράθυρο της εγκατάστασης, που περιέχει όλες τις βασικές επιλογές σχετικά με την εγκατάσταση και οι οποίες είναι οι παρακάτω:

- Ρύθμιση τοπικών χαρακτηριστικών του λειτουργικού συστήματος όπως οι γλώσσες εμφάνισης, οι γλώσσες εισόδου από το πληκτρολόγιο καθώς και η τοπική ώρα του συστήματος
- Εγκατάσταση λογισμικών στο σύστημα. Εδώ μπορούμε να επιλέξουμε αν θα γίνει η εγκατάσταση από το τοπικό μέσο που έχουμε δημιουργήσει ή αν θέλουμε να «κατεβάσουμε» την τελευταία έκδοση της Fedora από το δίκτυο. Επίσης μπορούμε να καθορίσουμε ποια «πακέτα» λογισμικών θέλουμε να εγκατασταθούν στο σύστημα μας ανάλογα με τη λειτουργικότητα που θέλουμε να επιτελεί ο Server που δημιουργούμε
- Επιλογές συστήματος. Εδώ καθορίζουμε σε ποιο αποθηκευτικό μέσο του συστήματος θα γίνει η εγκατάσταση του λειτουργικού και ποια θα είναι η διαμόρφωση των **καταμήσεων<sup>1</sup> (partitions)** του μέσου αυτού. Επίσης μπορούμε να ρυθμίσουμε και τις συνδέσεις του δικτύου που θα υποστηρίζει το λειτουργικό σύστημα που πρόκειται να εγκατασταθεί.
- Επιλογές σχετικές με τους λογαριασμούς χρηστών που θέλουμε να καθορίσουμε. Αυτό που πρέπει να ορίσουμε είναι ένα συνθηματικό (password) για τον χρήστη root καθώς και το όνομα και συνθηματικό ενός κανονικού χρήστη.



Εικόνα 1- 3: Κεντρική οθόνη εγκατάστασης

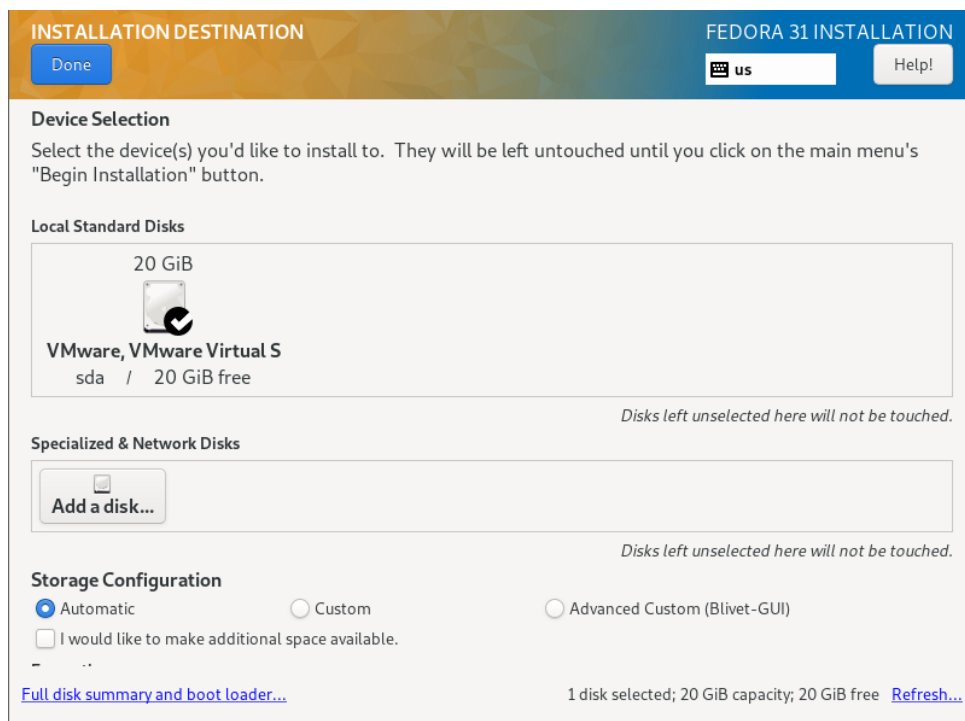
<sup>1</sup> Οι όροι κατάτμηση και partition θα χρησιμοποιούνται με την ίδια έννοια στην συνέχεια του εγγράφου

Αξίζει να σημειωθεί ότι η ρύθμιση των παραπάνω επιλογών δεν πρέπει να έχει κάποια συγκεκριμένη σειρά αλλά ο χρήστης μπορεί να επιλέξει ότι θέλει. Υπάρχει όμως ένα γραφικό σύστημα ειδοποίησης για την ανάγκη παρέμβασης του χρήστη. Συγκεκριμένα, αν κάποιο από τα εικονίδια έχει ένα κίτρινο θαυμαστικό, αυτό σημαίνει ότι πρέπει ο χρήστης να παρέμβει και να κάνει σε αυτή την επιλογή κάποια ρύθμιση γιατί διαφορετικά δεν μπορεί να ξεκινήσει η εγκατάσταση. Σε περίπτωση που δεν υπάρχει κανένα θαυμαστικό τότε ενεργοποιείται το πλήκτρο “*Begin Installation*” το οποίο ξεκινάει την εγγραφή των αρχείων από το μέσο εγκατάστασης στον σκληρό δίσκο του συστήματος (πραγματική εγκατάσταση). Επίσης ανά πάσα στιγμή ο χρήστης μπορεί να επιλέξει το πλήκτρο “*Quit*” και να γίνει έξοδος από το πρόγραμμα εγκατάστασης χωρίς να γίνει η εγκατάσταση του λειτουργικού συστήματος. Περισσότερες λεπτομέρειες για τις επιλογές εγκατάστασης μπορούν να βρεθούν στη σελίδα τεκμηρίωσης της διανομής Fedora (FedoraDocs-Install, n.d.).

Αφού ολοκληρωθεί η διαδικασία εγκατάστασης εμφανίζεται η επιλογή έναρξης του καινούργιου λειτουργικού συστήματος. Προσοχή, αν έχει γίνει αλλαγή στο BIOS για να γίνεται εκκίνηση από άλλο μέσο, π.χ. usb, τότε αυτή θα πρέπει να τροποποιηθεί έτσι ώστε η εκκίνηση να πραγματοποιηθεί από το μέσο εγκατάστασης.

### 1.3.1 Επιλογές εγκατάστασης

Μία από τις επιλογές που πρέπει να πραγματοποιήσουμε όταν εγκαθιστούμε το λειτουργικό σύστημα είναι η επιλογή του αποθηκευτικού μέσου στο οποίο θα γίνει η εγκατάσταση. Η Εικόνα 1- 4 παρουσιάζει τις επιλογές που έχουμε σε αυτή την περίπτωση.

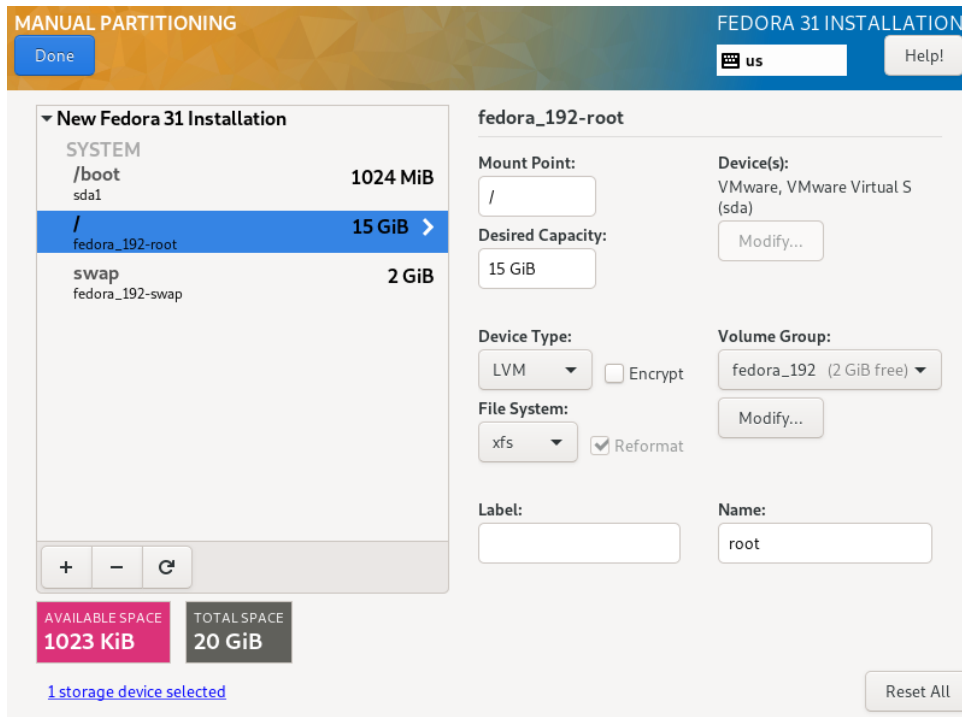


Εικόνα 1- 4: Επιλογή μέσου εγκατάστασης

Αρχικά πρέπει να επιλέξουμε το αποθηκευτικό μέσο, τον σκληρό δίσκο δηλαδή, στον οποίο θα γίνει η εγκατάσταση. Όπως παρουσιάζεται και στην Εικόνα 1- 4 το σύστημα μας διαθέτει



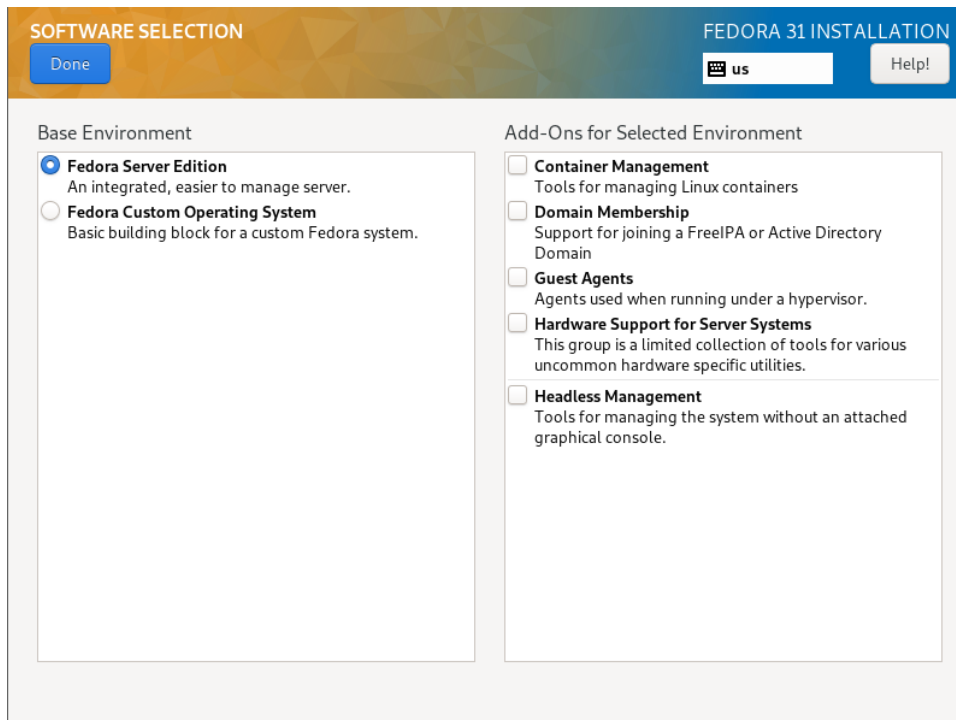
ένα δίσκο και οποίος έχει επιλεγεί (αυτό παρουσιάζεται με το σύμβολο *v* στο εικονίδιο του δίσκου). Επόμενο βήμα είναι η διαμόρφωση του μέσου αποθήκευσης και οι δημιουργία κατατμήσεων σε αυτόν. Σε αυτή την περίπτωση οι επιλογές που έχουμε είναι οι *“automatic”*, *“custom”* και *“advanced custom”*. Η επιλογή *“automatic”* θα κάνει αυτόματη διαμόρφωση του δίσκου ενώ η *“custom”* θα μας δώσει τη δυνατότητα να επιλέξουμε εμείς ακριβώς ποιες κατατμήσεις θα δημιουργηθούν και τι θα περιέχει η κάθε μία τους. Επιλέγοντας *“custom”* εμφανίζεται το παράθυρο *“MANUAL PARTITIONING”* όπου μπορούμε να επιλέξουμε τον σύνδεσμο *“click here to create them automatically”* με αποτέλεσμα την παρουσίαση του παραθύρου της Εικόνα 1- 5.



Εικόνα 1- 5: Καθορισμός διαμόρφωσης δίσκου

Στην Εικόνα 1- 5 παρουσιάζονται οι αυτόματες επιλογές της διανομής Fedora για τη διαμόρφωση του δίσκου και αυτές είναι η δημιουργία τριών κατατμήσεων α) μία για τον κατάλογο εκκίνησης */boot* β) μία για τον κατάλογο */* (*root*) και γ) μία για το **αρχείο εναλλαγής (swap file)**. Παρατηρούμε τόσο τον κατάλογο *root* όσο και το αρχείο εναλλαγής δεν τα αναθέτει σε κανονικές κατατμήσεις αλλά τα αποδίδει για διαχείριση μέσω του Logical Volume Manager (LVM). Περισσότερες πληροφορίες για τις κατατμήσεις και τη λειτουργικότητα τους θα παρουσιάσουμε στην Ενότητα 3.3.

Μία άλλη πολύ συχνή επιλογή που χρειάζεται να πραγματοποιήσουμε κατά την αρχική εγκατάσταση ενός λειτουργικού συστήματος, είναι να επιλέξουμε τα απαραίτητα λογισμικά που πρέπει να εγκατασταθούν στο σύστημα, έτσι ώστε αυτό να μπορεί να ανταποκρίνεται στη λειτουργικότητα για την οποία έχει σχεδιαστεί.



Εικόνα 1- 6: Επιλογή εγκατάστασης λογισμικών

Η Εικόνα 1- 6 παρουσιάζει τις επιλογές που έχουμε για την εγκατάσταση διαφόρων πακέτων λογισμικού που εξειδικεύουν τη λειτουργία του υπολογιστικού μας συστήματος.

## 1.4 Εκκίνηση, τερματισμός

Μία πολύ συχνή διαδικασία που θα κληθεί να εκτελέσει κάθε διαχειριστής συστήματος είναι η εκκίνηση και ο τερματισμός του συστήματος. Η εκκίνηση είναι μια πολύ «ευαίσθητη» διαδικασία στην οποία μερικές φορές παρουσιάζονται προβλήματα τα οποία πρέπει να αντιμετωπίσει ο διαχειριστής συστήματος. Η κατανόηση της διαδικασίας εκκίνησης και οι δυνατότητες παρέμβασης που έχει είναι γνώσεις που πρέπει να διαθέτει ένας διαχειριστής συστήματος.

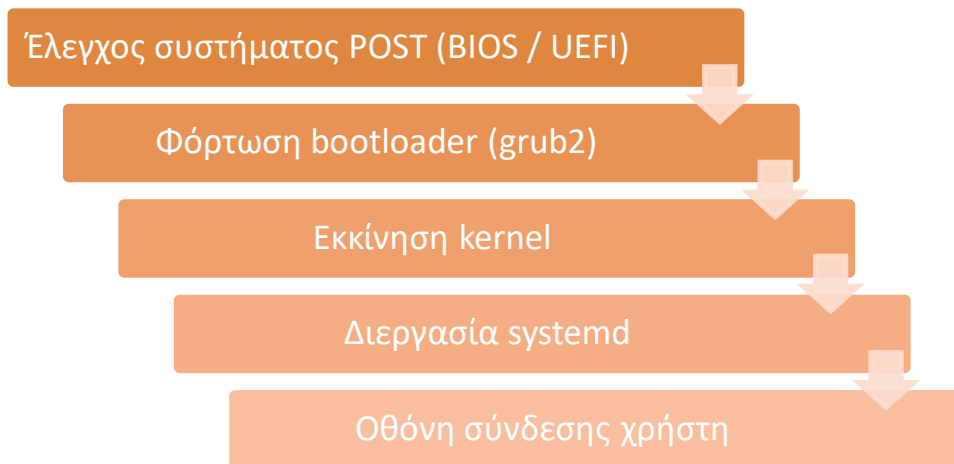
### 1.4.1 Διαδικασία εκκίνησης

Τα βασικά στάδια από τα οποία αποτελείται η διαδικασία εκκίνησης ενός προσωπικού υπολογιστή όπως παρουσιάζονται στην Εικόνα 1- 7 είναι τα παρακάτω (FedoraDocs-Boot, n.d.):

1. Με την εκκίνηση ενός υπολογιστικού συστήματος εκτελείται από ένα ειδικό πρόγραμμα που βρίσκεται αποθηκευμένο στη μνήμη ROM του συστήματος (**firmware**) με σκοπό την αναγνώριση των περιφερειακών του συστήματος και τον έλεγχο της ορθής λειτουργίας τους. Επιπρόσθετα, «φορτώνεται» από τις περιφερειακές συσκευές αποθήκευσης στη μνήμη ένα ειδικό πρόγραμμα που ονομάζεται **bootloader** και ξεκινά η εκτέλεση του.
2. Ο **bootloader** έχει σαν σκοπό την επιλογή του λειτουργικού συστήματος που θα εκτελεστεί. Στην περίπτωση του Linux φορτώνει τον kernel μαζί με ένα μικρό σύστημα αρχείων (**initramfs**) στην μνήμη του συστήματος. Το σύστημα αρχείων



- αυτό είναι μόνο για ανάγνωση και περιέχει όλα τα εργαλεία που χρειάζονται για να συνεχίσει τη διαδικασία εκκίνησης ο kernel.
3. Ο kernel με τη σειρά του ξεκινάει την πρώτη διεργασία του συστήματος (**systemd**) και μεταφέρει τον έλεγχο της διαδικασίας εκκίνησης σε αυτή.
  4. Το systemd έχει σαν σκοπό την εγκατάσταση των συστημάτων αρχείων που περιέχονται στο αρχείο **/etc/fstab** καθώς και την εκκίνηση συγκεκριμένων λογισμικών ανάλογα με τη λειτουργικότητα του συστήματος.
  5. Τελευταίο στάδιο στην εκκίνηση ενός συστήματος είναι η παρουσίαση της οθόνης σύνδεσης του χρήστη στο Linux σύστημα είτε με γραφικό είτε με περιβάλλον γραμμής εντολής.



Εικόνα 1- 7: Στάδια εκκίνησης του συστήματος

Πιο αναλυτικά, στο πρώτο στάδιο της εκκίνησης ο υπολογιστής ψάχνει στο τέλος της μνήμης του συστήματος, η οποία αποτελείται από κυκλώματα που δεν χάνουν το περιεχόμενο τους όταν ο υπολογιστής δεν έχει ρεύμα, για ένα πρόγραμμα που ονομάζεται Basic Input/Output System (**BIOS**) για τα πιο παλιά συστήματα ή το Unified Extensible Firmware Interface (**UEFI**) για τα πιο σύγχρονα. Και τα δύο αποτελούν τον σύνδεσμο ανάμεσα στο hardware του μηχανήματος και το λειτουργικό σύστημα που θα φορτωθεί σε αυτό. Σε γενικές γραμμές αυτό που κάνουν είναι μια σειρά από ελέγχους (POST – Power On Self Tests) που έχουν σαν σκοπό να ελέγξουν αν υπάρχουν όλα τα απαραίτητα συστατικά του συστήματος (επεξεργαστής, μνήμη, δίσκοι κτλ.) και αν λειτουργούν σωστά. Σε περίπτωση που δεν υπάρχει λάθος θα φορτωθεί το πρόγραμμα του bootloader από τον δίσκο εκκίνησης. Αν αναγνωριστεί λάθος, π.χ. πρόβλημα στη μνήμη RAM, τότε εμφανίζεται ένα μήνυμα στην οθόνη και παράλληλα ακούγονται ήχοι (μπιπ) από το ηχείο του συστήματος.

Υπάρχουν διάφορα προγράμματα bootloader αλλά το πιο διαδεδομένο στις σύγχρονες εκδόσεις του Linux είναι το **grub2**. Το grub2 αποτελείται από δύο στάδια τα οποία εκτελούνται διαδοχικά. Το μέρος στον σκληρό δίσκο στο οποίο βρίσκεται το πρώτο στάδιο του grub2 εξαρτάται από το εάν ο υπολογιστής μας έχει BIOS ή UEFI και από το εάν ο δίσκος εκκίνησης είναι διαμορφωμένος με το σύστημα Master Boot Record (**MBR**) ή το GUID Partition Table (**GPT**). Τελικό αποτέλεσμα της εκτέλεσης του πρώτου σταδίου του grub2 είναι να βρεθεί και να «φορτωθεί» στη μνήμη το δεύτερο στάδιο του grub2.

Σκοπός του δεύτερου σταδίου του grub2 είναι επιλογή του λειτουργικού συστήματος που θα «φορτωθεί» και θα εκτελεστεί στο σύστημα μας. Ένα πλεονέκτημα του grub2 είναι το γεγονός ότι μπορεί να υποστηρίξει πολλά συστήματα αρχείων ακόμα και εικονικά όπως το LVM καθώς και ότι μπορεί να υποστηρίξει πολλά λειτουργικά συστήματα (όπως π.χ. τα Windows). Η λειτουργία του grub2 εξειδικεύεται με βάση το περιεχόμενο του αρχείου διαμόρφωσης του (`/boot/grub2/grub.cfg` για τα BIOS συστήματα και `/boot/efi/EFI/fedora/grub.cfg` για τα UEFI συστήματα). Με την εκτέλεση του δεύτερου σταδίου του grub2 εμφανίζεται στον χρήστη μια οθόνη γραφικών που περιέχει όλα τα λειτουργικά συστήματα τα οποία είναι σε θέση να εκκινήσει ο υπολογιστής. Από την οθόνη αυτή ο χρήστης με τα πλήκτρα κατεύθυνσης (βελάκια) και Enter επιλέγει το λειτουργικό σύστημα που θέλει να εκκινήσει, αν δεν επιλέξει μετά από λίγο χρόνο επιλέγεται ένα προεπιλεγμένο. Στη συνέχεια το grub2 βρίσκει το αντίστοιχο εκτελέσιμο αρχείο του kernel που υπάρχει στο κατάλογο `/boot` και το φορτώνει στη μνήμη. Τα εκτελέσιμα αρχεία έχουν τη μορφή `/boot/vmlinuz-<kernel-version>`. Κατά τη διάρκεια φόρτωσης του kernel ο bootloader μπορεί να περάσει παραμέτρους όπως π.χ. να λειτουργεί το σύστημα μας με συγκεκριμένες δυνατότητες (runlevel). Παράλληλα με τη φόρτωση του kernel το grub2 φορτώνει το κατάλληλο προσωρινό σύστημα αρχείων (initramfs images) στη μνήμη που περιέχει οδηγούς συσκευών (**drivers**) και modules απαραίτητα για την εκκίνηση του συστήματος και περνάει τον έλεγχο στο **systemd**.

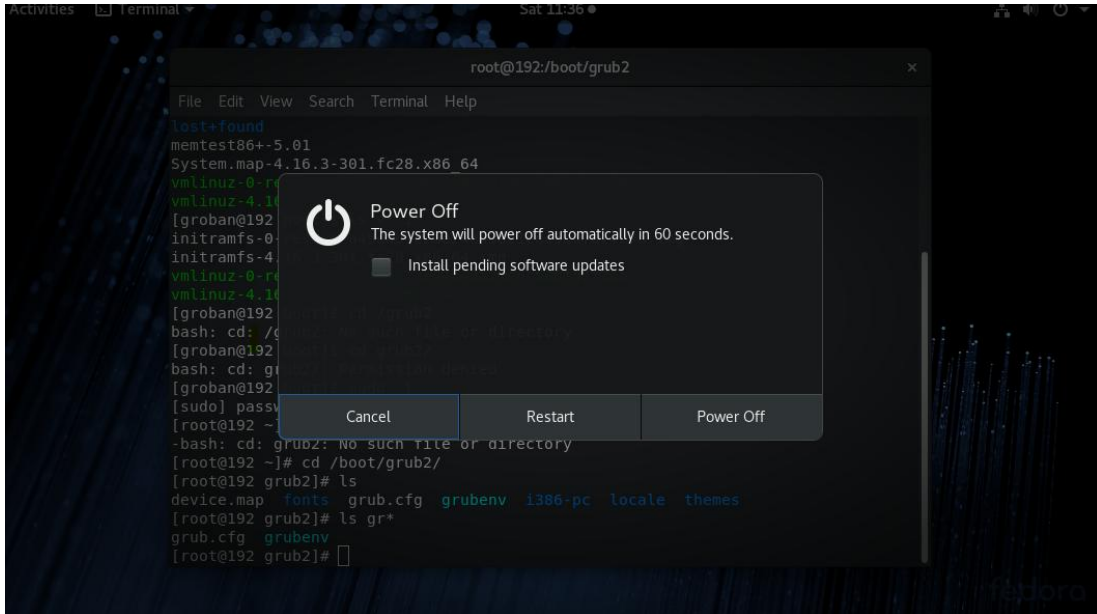
Το systemd είναι η πρώτη διεργασία που δημιουργείται από τον Kernel. Το systemd έχει αντικαταστήσει τη διεργασία init που χρησιμοποιούταν σε παλαιότερα συστήματα (systemV). Σκοπός του systemd είναι να φορτώσει όλες τις υπηρεσίες που πρέπει να παρέχει το λειτουργικό σύστημα. Στο τέλος αυτής της διαδικασίας εμφανίζεται η οθόνη στην οποία ο χρήστης καλείται να εισαχθεί στο σύστημα.



Η μετάβαση από το system V στο system ήταν μια από τις μεγάλες αλλαγές που έχουν πραγματοποιηθεί στο Linux κατά τη διάρκεια της εξέλιξης του. Η Fedora ήταν η πρώτη διανομή που υποστήριξε το systemd, όπως και πολλές άλλες καινοτομίες, και τώρα πλέον υποστηρίζεται από σχεδόν όλες τις διανομές. Για περισσότερες πληροφορίες για τις διαφορές τους και την ανάγκη εξέλιξης του system V μπορείτε να δείτε στο (TechMint, 2017) .

#### 1.4.2 Επανεκκίνηση και τερματισμός

Ο τερματισμός σε γραφικό περιβάλλον γίνεται με την επιλογή του εικονιδίου τερματισμού που βρίσκεται στο πάνω και δεξιά μέρος της επιφάνειας εργασίας και στη συνέχεια την επιλογή μιας από τις ενέργειες του τερματισμού της επανεκκίνησης ή της ακύρωσης όπως παρουσιάζονται στην Εικόνα 1- 8. Αν θέλουμε να θέσουμε μέσω του γραφικού περιβάλλοντος την αδρανοποίηση (sleep) του συστήματος θα επιλέξουμε πάλι το εικονίδιο τερματισμού πάνω και δεξιά της επιφάνειας εργασίας μόνο που θα έχουμε και πατημένο το πλήκτρο alt.



Εικόνα 1- 8: Διαδικασία τερματισμού στο γραφικό περιβάλλον

Στις σύγχρονες εκδόσεις του Linux που χρησιμοποιούν το systemd ο τερματισμός της λειτουργίας του Linux γίνεται με τη χρήση της εντολής **systemctl**. Παρόλα αυτά, για να υπάρχει συμβατότητα με τις προηγούμενες εκδόσεις του Linux, διατηρείται και η ύπαρξη των παρακάτω εντολών (Moon, 2020):

- **halt** σταματά το σύστημα χωρίς να διακόπτει την παροχή ρεύματος
- **poweroff** σταματά την παροχή ρεύματος
- **reboot** πραγματοποιεί μία επανεκκίνηση του συστήματος
- **shutdown** μπορεί να επιτελέσει όλα τα παραπάνω

Θα παρουσιάσουμε τη λειτουργικότητα της εντολής shutdown. Επίσης, δεν πρέπει να ξεχνάμε, ότι ένα Linux σύστημα μπορεί να χρησιμοποιείται από πολλούς χρήστες την ίδια στιγμή. Επομένως αν θέλουμε να τερματίσουμε το σύστημα πρέπει να ενημερώσουμε γι' αυτό τους χρήστες που βρίσκονται στο σύστημα. Η σύνταξη της εντολής shutdown είναι η παρακάτω:

***shutdown [παραμέτροι] [χρόνος] [μήνυμα στους χρήστες]***

Μερικές από τις παραμέτρους που μπορούμε να χρησιμοποιήσουμε είναι η *P* (*Poweroff*) που εκτελεί τερματισμό και η *H* (*Halt*) που θέτει το σύστημα σε κατάσταση αδράνειας. Επίσης ο χρόνος στον οποίο θα γίνει η ενέργεια που επιλέγουμε μπορεί να είναι είτε στη μορφή ΩΩ:ΛΛ (μορφή 24 ωρών) είτε με τη χρήση του χαρακτήρα + πριν οπότε εννοεί τόσο χρόνο μετά από τώρα. Επίσης αν υπάρχει η λέξη now στη θέση του χρόνου τότε εκτελείται η ενέργεια άμεσα. Τελευταίο είναι ένα μήνυμα το οποίο θα δουν όλοι οι χρήστες που χρησιμοποιούν το σύστημα εκείνη τη στιγμή.

Για παράδειγμα η εντολή που θα τερματίσει το σύστημα στις 10 το βράδυ και θα ενημερώσει για αυτό όλους τους χρήστες είναι η παρακάτω.

```
[root@linux ~]# shutdown -P 22:00 "The system is terminated at
10:00 pm"
Shutdown scheduled for Sun 2020-11-22 10:00:00 EET, use 'shutdown
-c' to cancel.
```

Στις παραπάνω εντολές παρατηρούμε ότι αν θέλουμε ακυρώσουμε έναν προγραμματισμένο τερματισμό εκτελούμε την εντολή `shutdown` με την παράμετρο `c`.

Ένα άλλο παράδειγμα είναι αν θέλουμε να εκτελέσουμε αδρανοποίηση του συστήματος σε 20 λεπτά από τώρα οπότε εκτελούμε την παρακάτω εντολή.

```
[root@linux ~]# shutdown -H +20 "The system is halting in 20
minutes"
Shutdown scheduled for Sun 2020-11-22 10:02:44 EET, use 'shutdown
-c' to cancel.
```

### 1.4.3 Διαχείριση του συστήματος κατά την εκκίνηση

Όπως αναφέραμε και προηγουμένως το πρόγραμμα που αναλαμβάνει την εκκίνηση του λειτουργικού συστήματος είναι ο bootloader `grub2`. Η λειτουργία του `grub2` βασίζεται στα περιεχόμενα του αρχείου `/boot/grub2/grub.cfg` (Both, 2017). Τα περιεχόμενα αυτού του αρχείου δημιουργούνται αυτόματα από το `grub2` και δεν πρέπει να επέμβουμε απευθείας στα περιεχόμενα του. Αν θέλουμε να τροποποιήσουμε τη λειτουργία του `grub2` μπορούμε να το κάνουμε χρησιμοποιώντας το αρχείο `/etc/default/grub`. Τα περιεχόμενα αυτού του αρχείου εμφανίζονται με τη χρήση της εντολής `cat` στο παρακάτω πλαίσιο.

```
[root@ekdda user]# cat /etc/default/grub
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR="$(sed 's, release .*$,,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="resume=/dev/mapper/fedora_localhost--live-
swap rd.lvm.lv=fedora_localhost-live/root
rd.lvm.lv=fedora_localhost-live/swap rhgb quiet"
GRUB_DISABLE_RECOVERY="true"
GRUB_ENABLE_BLSCFG=true
```

Όπως παρατηρούμε παραπάνω το αρχείο `/etc/default/grub` αποτελείται από μια σειρά από μεταβλητές η τιμή των οποίων καθορίζει τη λειτουργία του `grub2`. Η πρώτη μεταβλητή `GRUB_TIMEOUT=5` καθορίζει σε 5 δευτερόλεπτα τον χρόνο που μας αφήνει το `grub2` για να επιλέξουμε ποιο kernel θέλουμε να ξεκινήσουμε. Η μεταβλητή `GRUB_DEFAULT=saved` καθορίζει ότι θα είναι προεπιλεγμένη επιλογή η τελευταία που επιλέχθηκε. Ενώ πολύ βασική είναι και η μεταβλητή `GRUB_CMDLINE_LINUX` επειδή καθορίζει τις παραμέτρους που θα περαστούν στο kernel για την εκτέλεση του όπως θα δούμε και παρακάτω.

Για παράδειγμα αν θέλουμε να δώσουμε χρονικό περιθώριο 10 δευτερολέπτων στον χρήστη έτσι ώστε να επιλέξει το λειτουργικό που θέλει να εκτελέσει αρχικά θα πρέπει να επεμβούμε στο αρχείο `/etc/default/grub` και να αλλάξουμε την τιμή της μεταβλητής `GRUB_TIMEOUT` σε 10. Στη συνέχεια καλούμε την εντολή **grub2-mkconfig** για να δημιουργήσει το καινούργιο αρχείο το οποίο και καθορίζουμε ότι θα είναι το `/boot/grub2/grub.cfg`. Η διαδικασία φαίνεται στο παρακάτω πλαίσιο:

```
[root@linux ~]# grub2-mkconfig > /boot/grub2/grub.cfg
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-4.16.3-301.fc28.x86_64
Found initrd image: /boot/initramfs-4.16.3-301.fc28.x86_64.img
Found linux image: /boot/vmlinuz-0-rescue-
a45dab422ae24d43a21e6c5e6fd7f25a
Found initrd image: /boot/initramfs-0-rescue-
a45dab422ae24d43a21e6c5e6fd7f25a.img
```

Αυτό που βλέπουμε είναι ότι το `grub2-mkconfig` δημιουργεί το καινούργιο `/boot/grub2/grub.cfg` αρχείο. Για να το κάνει χρειάζεται τόσο τις καινούργιες επιλογές που έχουμε κάνει καθώς και τα `linux kernels` όπως και τα προσωρινά συστήματα αρχείων (`initramfs`) που υπάρχουν στο σύστημα μας.



Για μία εκτεταμένη παρουσίαση του `grub2` και των ρυθμίσεων που μπορούμε να κάνουμε σε αυτόν μπορείτε να επισκεφτείτε την ιστοσελίδα αναφοράς του (`GNU, GNU GRUB Manual 2.04, n.d.`).

Επειδή η ρύθμιση αλλαγής της `default` επιλογής του λειτουργικού είναι πολύ συχνή υπάρχει και εξειδικευμένη εντολή που εκτελεί την αλλαγή και είναι η `grub2-set-default`. Προσοχή η αρίθμηση των λειτουργικών που υπάρχουν αρχίζει από το μηδέν. Μία άλλη ρύθμιση που πρέπει να κάνουμε, αν έχουμε έκδοση `Fedora 30` και νεότερη, είναι να εκτελέσουμε την παρακάτω ρύθμιση που μας εμφανίζει το μενού επιλογής του `grub`.

```
[root@linux ~]# grub2-editenv - unset menu_auto_hidedone
```

#### 1.4.4 Εκκίνηση σε κατάσταση επίλυσης προβλήματος

Μία από τις δυσάρεστες καταστάσεις που μπορεί να βιώσουμε σαν διαχειριστές ενός συστήματος είναι όταν το σύστημα μας δεν μπορεί να εκκινήσει κανονικά και εμφανίζει μήνυμα λάθους. Σε αυτή την περίπτωση, αυτό που μπορούμε να κάνουμε, είναι να εκκινήσουμε το σύστημα σε μία κατάσταση περιορισμένων λειτουργιών έτσι ώστε να εκτελέσουμε κάποιους ελέγχους ή διορθώσεις για να επαναφέρουμε το σύστημα σε κανονική λειτουργία (`FedoraDocsBasic-Recovery, n.d.`). Την εκκίνηση σε αυτή την ειδική κατάσταση θέλουμε να την κάνουμε μόνο για αυτή τη φορά και όχι μόνιμα όπως θα κάναμε με τη διαδικασία τροποποίησης του `/etc/default/grub.cfg` αρχείου που περιγράψαμε στην προηγούμενη ενότητα. Η διαδικασία που πρέπει να εκτελέσουμε είναι η ακόλουθη: Αρχικά

εκκινούμε κανονικά το σύστημα και όταν εμφανιστεί η οθόνη του grub2 όπου πρέπει να γίνει η επιλογή του kernel τότε πιέζουμε το πλήκτρο e από το πληκτρολόγιο και εμφανίζεται η Εικόνα 1-9.

```
load_video
set gfx_payload=keep
insmod gzio
linux ($root)/vmlinuz-5.3.7-301.fc31.x86_64 root=/dev/mapper/fedora_localhost-
-live-root ro resume=/dev/mapper/fedora_localhost--live-swap rd.lvm.lv=fedora_
localhost-live/root rd.lvm.lv=fedora_localhost-live/swap rhgb quiet
initrd ($root)/initramfs-5.3.7-301.fc31.x86_64.img

Press Ctrl-x to start, Ctrl-c for a command prompt or Escape to
discard edits and return to the menu. Pressing Tab lists
possible completions.
```

Εικόνα 1-9: Ρύθμιση παραμέτρων του grub2 κατά την εκκίνηση

Σε αυτό το κείμενο κατεβαίνουμε με το πλήκτρο κατεύθυνσης (βελάκι) και βρίσκουμε τη γραμμή που ξεκινάει με το κείμενο "linux". Αυτή είναι η γραμμή του grub2 που δίνει εντολή στο αντίστοιχο kernel να ξεκινήσει. Στο τέλος αυτής της γραμμής μπορούμε να περάσουμε παραμέτρους που να εξειδικεύουν την εκκίνηση του kernel. Οι προκαθορισμένες επιλογές είναι οι παρακάτω

```
linux ($root)/vmlinuz-5.3.7-301.fc31.x86_64 root=/dev/mapper/fed-
ora_localhost-live-root ro resume=/dev/mapper/fedora_localhost-
-live/root rd.lvm.lv=fedora_localhost-live/swap rhgb quiet
```

Στην παραπάνω γραμμή παρατηρούμε ότι καθορίζονται οι κατατμήσεις που περιέχουν το root directory καθώς και το αρχείο εναλλαγής καθώς καθορίζονται και οι λογικές κατατμήσεις των δίσκων που δημιουργεί το LVM. Τέλος υπάρχουν δύο οδηγίες η πρώτη (*rhgb*) καθορίζει την εμφάνιση του γραφικού logo της fedora κατά την εκκίνηση του συστήματος και η δεύτερη (*quiet*) επιβάλλει την εμφάνιση μόνο των σημαντικών μηνυμάτων κατά την εκκίνηση. Όλες οι διαθέσιμες παράμετροι που μπορούν να δοθούν στην εκκίνηση του kernel εμφανίζονται στο manual page της dracut.cmdline (7).

Μία σημαντική ρύθμιση που μπορούμε να καθορίσουμε σε αυτό το σημείο είναι η «κατάσταση» στην οποία θέλουμε να εκκινήσει το σύστημα μας. Με τον όρο «κατάσταση» εννοούμε τα προγράμματα και τις υπηρεσίες που θέλουμε να είναι διαθέσιμες κατά τη λειτουργία του. Ο όρος κατάσταση λειτουργίας του συστήματος περιγράφεται στη βιβλιογραφία **runlevel** για τις παλαιότερες διανομές της Fedora και **target** για τις νεότερες<sup>2</sup>. Όταν το σύστημα μας δεν μπορεί να εκκινήσει σε κανονική λειτουργία μπορούμε να

<sup>2</sup> Περισσότερα για τα runlevels και τα targets σε επόμενο κεφάλαιο

ζητήσουμε την εκκίνηση του σε μια πιο «περιορισμένη» λειτουργία όπου μόνο τα βασικά προγράμματα θα ξεκινήσουν κατά την εκκίνηση του υπολογιστή. Αυτή η κατάσταση λειτουργίας περιγράφεται σαν **rescue.target**. Για να ξεκινήσει το σύστημα μας σε **rescue.target** πηγαίνουμε και προσθέτουμε το τέλος της γραμμής **linux** την παράμετρο **systemd.unit=rescue.target**. Για να ενημερώσουμε το **grub2** ότι τελειώσαμε με τις αλλαγές και θέλουμε ξεκινήσει η εκτέλεση του **kernel**, θα πρέπει να πατήσουμε τον συνδυασμό πλήκτρων **control+x** στο πληκτρολόγιο. Στη συνέχεια, θα μας ζητήσει το **root password** (προσοχή! σε εκδόσεις που δεν ορίζεται αυτόματα **root password** πρέπει να έχουμε ορίσει εμείς προηγουμένως) και θα ξεκινήσει το λειτουργικό σύστημα σε περιβάλλον γραμμής εντολών. Επιπρόσθετα, το σύστημα αρχείων θα είναι σε κατάσταση μόνο ανάγνωσης και θα εκτελούνται μόνο τα απαραίτητα λογισμικά και υπηρεσίες για τη λειτουργία του συστήματος.

Αν το σύστημα μας δεν ξεκινάει ούτε στην **rescue.target** μπορούμε ή αν δεν γνωρίζουμε το **root password**, μπορούμε να ξεκινήσουμε το σύστημα μας σε μία ακόμα πιο «περιορισμένη» κατάσταση λειτουργίας η οποία δεν απαιτεί γνώση του συνθηματικού του **root**. Για να γίνει αυτή η εκκίνηση, ακολουθούμε την παραπάνω διαδικασία που περιγράψαμε στο **rescue.target**, αλλά αυτή τη φορά προσθέτουμε στο τέλος την παράμετρο **rd.break**. Αυτή η παράμετρος έχει σαν αποτέλεσμα την εμφάνιση της γραμμής εντολών όπου μπορούμε να εκτελέσουμε κάποιες εντολές διαχείρισης. Μία από τις δυνατότητες που μας δίνει αυτή η επιλογή μας είναι να εκτελέσουμε κάποιες ενέργειες που θα έχουν σαν αποτέλεσμα την τοποθέτηση ενός καινούργιου συνθηματικού για τον **root** στο σύστημα μας (Warfield, 2018). Όπως καταλαβαίνουμε από τα παραπάνω, η πρόσβαση στο αρχείο ρύθμισης παραμέτρων του **grub2** κατά την εκκίνηση, είναι μια σημαντική δυνατότητα που μπορεί να δημιουργήσει ένα κενό ασφαλείας στο σύστημα μας επειδή παρέχει δυνατότητα αλλαγής στο συνθηματικό του **root**. Μία από τις λύσεις είναι να προστατέψουμε την πρόσβαση στο αρχείο αυτό με κωδικό. Η εντολή που το κάνει αυτό είναι η **grub2-set-password** όπως φαίνεται στο παρακάτω πλαίσιο.

```
[root@ekdda ~]# grub2-set-password
Enter password:
Confirm password:
[root@ekdda ~]# grub2-mkconfig >/boot/grub2/grub.cfg
Generating grub configuration file ...
done
```



## 2 Προγραμματισμός στο φλοιό Bash

Ένα από τα βασικά προσόντα που πρέπει να διαθέτει ένας διαχειριστής Linux συστήματος είναι η γνώση προγραμματισμού στο φλοιό (**shell**) του συστήματος (Kili, 2019). Φλοιός ονομάζεται το πρόγραμμα που δέχεται τις εντολές που δίνει ο χρήστης και τις μεταφέρει στο λειτουργικό σύστημα για να τις εκτελέσει. Εκτός από τη δυνατότητα εκτέλεσης εντολών οι φλοιοί διαθέτουν και κάποια χαρακτηριστικά γλωσσών προγραμματισμού όπως οι μεταβλητές, οι δομές επιλογής, επανάληψης, οι συναρτήσεις κ.α.. Υπάρχουν αρκετοί φλοιοί μερικοί από αυτούς είναι ο sh, ο csh και ο **bash** (Bourne Again Shell) (OpenSource, n.d.). Στη συνέχεια του κειμένου θα ασχοληθούμε με τον bash επειδή αυτός είναι ο προεπιλεγμένος φλοιός στη διανομή Fedora.

Τις εντολές που πρόκειται να εκτελέσει ο φλοιός μπορεί να τις δέχεται είτε από το πληκτρολόγιο είτε από ένα αρχείο κειμένου που ονομάζεται σενάριο (**script**). Μερικοί από τους λόγους για τους οποίους ο διαχειριστής ενός συστήματος μπορεί να χρησιμοποιήσει ένα bash script είναι οι ακόλουθοι:

- Αυτοματισμό ενεργειών που πραγματοποιούνται συχνά όλες μαζί
- Επεξεργασία αρχείων κειμένου μεγάλου μεγέθους
- Δημιουργία απλών εφαρμογών για εξειδικευμένες επεξεργασίες

Η γνώση προγραμματισμού σε bash είναι ένα από τα απαραίτητα προσόντα που πρέπει να έχει ένας διαχειριστής Linux Server. Ακόμα και αν δεν δημιουργεί δικά του script, πολύ συχνά θα πρέπει να κατανοήσει και προσαρμόσει στις ανάγκες του συστήματος του, script που υπάρχουν έτοιμα.



Ο προγραμματισμός στο φλοιό bash είναι ένα εκτενές πεδίο με πολλά αντικείμενα. Σκοπός του παρακάτω κεφαλαίου είναι να αποτελέσει μια εισαγωγή σε αυτόν με συνοπτική παρουσίαση των κυριότερων αντικειμένων του. Για μια αναλυτική παρουσίαση των δυνατοτήτων του bash μπορείτε να επισκεφθείτε τη σελίδα αναφορά του (GNU, Bash Reference Manual, n.d.).

### 2.1 Εισαγωγή στη δημιουργία σεναρίων (script) του φλοιού bash

Ένα αρχείο script είναι ένα αρχείο κειμένου όπου κάθε μία του γραμμή είναι και μία εντολή που θα εκτελέσει ο bash. Αν μια εντολή είναι μεγάλη και χρειάζεται να περάσει στην επόμενη γραμμή μπορεί να χρησιμοποιηθεί ο χαρακτήρας \. Η πρώτη γραμμή ενός script καθορίζει ποιο πρόγραμμα θα εκτελέσει τις εντολές που ακολουθούν. Γραμμές που ξεκινούν με τον χαρακτήρα # (εκτός από την πρώτη) είναι σχόλια και δεν εκτελούνται. Για να δημιουργήσουμε το πρώτο μας script, ανοίγουμε έναν επεξεργαστή κειμένου και εισάγουμε το παρακάτω κείμενο



```

1 #!/bin/bash
2
3 # echo "this is a comment"
4 echo "Welcome to the System!"
5 date
6 pwd

```

Αρχικά αποθηκεύουμε το παραπάνω αρχείο με το όνομα *first.sh*. Η επέκταση *sh* δεν είναι απαραίτητη για το Linux αλλά είναι μια ένδειξη για εμάς τους χρήστες ότι το αρχείο αυτό περιέχει ένα bash script. Για να μπορέσει ένας χρήστης να εκτελέσει αυτό το script θα πρέπει να έχει σε αυτό δικαιώματα ανάγνωσης και εκτέλεσης. Επιπρόσθετα, για να το εκτελέσει, είτε θα πρέπει να γράψει το πλήρες όνομα του, είτε θα πρέπει να υπάρχει ο τρέχων κατάλογος στη μεταβλητή περιβάλλοντος *PATH*. Η μεταβολή των δικαιωμάτων και η εκτέλεση του script εμφανίζονται στο παρακάτω πλαίσιο.

```

[user@ekdda scripts]$ chmod +x first.sh
[user@ekdda scripts]$ first.sh
bash: first.sh: command not found...
[user@ekdda scripts]$ ./first.sh
Welcome to the System!
Mon 23 Nov 2020 05:56:38 PM EET
/home/user/scripts

```

Ενδιαφέρον παρουσιάζει η πρώτη γραμμή που ονομάζεται shebang (*#!/bin/bash*). Αυτή καθορίζει το πρόγραμμα το οποίο θα εκτελέσει το script που ακολουθεί. Στο script που δημιουργήσαμε καθορίζουμε ότι θα εκτελεστεί από το bash. Αν αγνοήσουμε το shebang οι περισσότεροι φλοιοί θα προσπαθήσουν να το εκτελέσουν με τον εαυτό τους.

## 2.2 Στοιχεία του προγραμματισμού σε bash

Όπως έχουμε ήδη αναφέρει η βασική λειτουργία του φλοιού bash είναι η εκτέλεση των εντολών και η εμφάνιση των αποτελεσμάτων. Παράλληλα όμως ο bash διαθέτει όμως και βασικά χαρακτηριστικά μιας γλώσσας προγραμματισμού. Ένα από αυτά είναι και η δημιουργία και η χρήση μεταβλητών (Budzar, 2020). Οι μεταβλητές είναι ένας μηχανισμός χρήσης της μνήμης του υπολογιστή για την αποθήκευση δεδομένων και πληροφοριών.

### 2.2.1 Μεταβλητές τοπικές / περιβάλλοντος

Η εντολή του bash με την οποία ορίζουμε μια μεταβλητή μαζί με την τιμή της έχει την παρακάτω σύνταξη:

***Όνομα\_μεταβλητής=τιμή\_μεταβλητής***

Σε αντίθεση με άλλες γλώσσες προγραμματισμού δεν χρειάζεται να δηλώσουμε μια μεταβλητή αλλά μπορούμε να κάνουμε κατευθείαν ανάθεση τιμής σε αυτή. Επίσης ο τύπος όλων των τιμών στο bash είναι κείμενο (text). Προσοχή! Η παραπάνω δήλωση τιμή σε μεταβλητή είναι και αυτή μια εντολή του φλοιού bash. Γι' αυτό το λόγο δεν θα πρέπει να

έχει κενά σε κανένα σημείο γιατί τότε το bash θα νομίσει ότι πρόκειται για μία εντολή με παραμέτρους. Επιπρόσθετα το όνομα της μεταβλητής δεν πρέπει να έχει κενά ή περιέργους χαρακτήρες (π.χ. τον χαρακτήρα δέση #) ενώ ακόμα το bash κάνει διάκριση μεταξύ πεζών και κεφαλαίων χαρακτήρων. Αν η τιμή που θέλουμε να αναθέσουμε στη μεταβλητή περιέχει κενά τότε θα πρέπει να χρησιμοποιήσουμε μονά η διπλά εισαγωγικά<sup>3</sup>. Στο παρακάτω παράδειγμα παρατηρούμε μερικές δηλώσεις μεταβλητών μαζί με συχνά λάθη που γίνονται σε αυτές.

```
[user@ekdda scripts]$ city=thessaloniki
[user@ekdda scripts]$ age=24
[user@ekdda scripts]$ age = 44
bash: age: command not found...
[user@ekdda scripts]$ comment="Value with spaces"
```

Αν θέλουμε να χρησιμοποιήσουμε την τιμή μιας μεταβλητής τότε γράφουμε το όνομα της μεταβλητής με τον χαρακτήρα δολάριο (\$) πριν από αυτό. Για να εμφανίσουμε ένα μήνυμα μαζί με την τιμή μιας μεταβλητής στην οθόνη χρησιμοποιούμε την εντολή echo όπως παρακάτω:

```
[user@ekdda scripts]$ echo $city
thessaloniki
[user@ekdda scripts]$ echo $year

[user@ekdda scripts]$ echo i live in $city
i live in thessaloniki
[user@ekdda scripts]$ file=/home/user/data
[user@ekdda scripts]$ echo ${file}.txt
/home/user/data.txt
```

Στο παραπάνω παράδειγμα παρατηρούμε ότι μπορούμε να παραθέσουμε μαζί σταθερό κείμενο μαζί με την τιμή μιας μεταβλητής. Επίσης παρατηρούμε ότι αν προσπαθήσουμε να χρησιμοποιήσουμε την τιμή μιας μεταβλητής που δεν έχει οριστεί τότε δε θα υπάρξει μήνυμα λάθους αλλά απλώς δε θα τυπωθεί τίποτα. Ακόμα αν θέλουμε να προσθέσουμε κείμενο δίπλα στην τιμή μιας μεταβλητής τότε θα πρέπει να βάλουμε την τιμή της μεταβλητής μέσα σε άγκιστρα (\${file}) έτσι ώστε το bash να μπορεί να ξεχωρίσει που τελειώνει η τιμή της μεταβλητής και που ξεκινάει το κανονικό κείμενο.



Όταν τοποθετώ το όνομα μιας μεταβλητής μέσα σε άγκιστρα μπορώ να εκτελέσω μια σειρά από ενέργειες πάνω στην τιμή της, που ονομάζονται string manipulators. Για μια πιο λεπτομερή παράθεση των δυνατοτήτων τους μπορείτε να επισκεφθείτε το *Advanced Bash-Scripting Guide* (Cooper, 2014)

<sup>3</sup> Τη διάκριση μεταξύ μονών και διπλών εισαγωγικών θα τη δούμε παρακάτω

## Άλλοι τρόποι ανάθεσης τιμής σε μεταβλητή

Μια μεταβλητή στο bash μπορεί να πάρει τιμή, εκτός από την απευθείας ανάθεση, με ακόμα δύο τρόπους. Ο πρώτος είναι με την ανάγνωση μιας τιμής από το πληκτρολόγιο με τη χρήση της εντολής `read` (LandOfLinux, n.d.). Η σύνταξη της εντολής `read` είναι η παρακάτω:

```
read [-p 'μήνυμα προς χρήστη'] Ονόματα_μεταβλητών_με_κενά
```

Στο τέλος της εντολής `read` υπάρχουν τα ονόματα ενός ή περισσότερων ονομάτων μεταβλητών που χωρίζονται μεταξύ τους με κενά. Η εντολή `read` αναμένει από τον χρήστη την είσοδο μιας γραμμής κειμένου από το πληκτρολόγιο (μέχρι τον χαρακτήρα `enter`). Στη συνέχεια αναθέτει το κείμενο μέχρι το πρώτο κενό<sup>4</sup>, σαν τιμή στην πρώτη μεταβλητή, το επόμενο κείμενο μέχρι το δεύτερο κενό, στη δεύτερη μεταβλητή κ.τ.λ.. Αν εξαντληθεί το κείμενο και υπάρχουν μεταβλητές, αυτές δεν έχουν τιμή, ενώ αν υπάρχουν περισσότερα κενά στο κείμενο η τελευταία μεταβλητή παίρνει όλο το κείμενο που περισσεύει μαζί με τα κενά. Επιπρόσθετα, η εντολή `read` μπορεί να πάρει την παράμετρο `r` με ένα μήνυμα που εμφανίζει στον χρήστη. Μερικά παραδείγματα χρήσης της εντολής `read` εμφανίζονται στο παρακάτω πλαίσιο:

```
[user@ekdda scripts]$ read -p 'Give Employess:' NAME1 NAME2
Give Employess:Nikolaou Petrou
[user@ekdda scripts]$ read -p 'Give Employess:' NAME1 NAME2
Give Employess:Nikolaou
[user@ekdda scripts]$ echo $NAME1 $NAME2
Nikolaou
[user@ekdda scripts]$ read -p 'Give Employess:' NAME1 NAME2
Give Employess:Nikolaou Petrou Charisis
[user@ekdda scripts]$ echo $NAME2 $NAME1
Petrou Charisis Nikolaou
```

Επίσης τα δεδομένα που λαμβάνει η `read` μπορεί να είναι όχι από το πληκτρολόγιο αλλά από ένα αρχείο κειμένου, αν χρησιμοποιηθεί η ανακατεύθυνση εισόδου όπως στο παρακάτω παράδειγμα:

```
[user@ekdda scripts]$ cat values.dat
22 21 45 32 55
[user@ekdda scripts]$ read mon tue wed thu fri < values.dat
[user@ekdda scripts]$ echo Value of Wednesday: $wed
Value of Wednesday: 45
```

Ο δεύτερος και πολύ σημαντικός τρόπος με τον οποίο μπορεί να γίνει ανάθεση τιμής σε μία μεταβλητή είναι με την εκτέλεση μιας εντολής. Η τεχνική αυτή ονομάζεται αντικατάσταση

<sup>4</sup> Για την ακρίβεια μέχρι τον χαρακτήρα που είναι πρώτος στην μεταβλητή συστήματος `IFS`

εντολής (**command substitution** (Newham & Rosenblett, 2005)) έχει την παρακάτω σύνταξη:

**όνομα\_μεταβλητής=\$(εντολή *bash*)**

Σε αυτή την περίπτωση η μεταβλητή παίρνει σαν τιμή το αποτέλεσμα εκτέλεσης της εντολής. Η εντολή δηλαδή δεν εμφανίζει τα αποτελέσματα στην οθόνη αλλά τα «περνάει» σαν τιμή στη μεταβλητή που ορίζουμε. Η τεχνική αυτή είναι πολύ χρήσιμη στον προγραμματισμό στο bash. Μερικά παραδείγματα παρουσιάζονται στο παρακάτω πλαίσιο.

```
[user@ekdda scripts]$ now=$(date)
[user@ekdda scripts]$ echo $now
Tue 24 Nov 2020 10:35:09 AM EET
[user@ekdda scripts]$ year=$(date +%Y)
[user@ekdda scripts]$ echo $year
2020
[user@ekdda scripts]$ files=$(ls)
[user@ekdda scripts]$ echo $files
first.sh values.dat
```

### Εισαγωγικά (μονά – διπλά – ανάποδα)

Ένα πρόβλημα που αντιμετωπίζουν όσοι ξεκινούν προγραμματισμό στο bash είναι η διάκριση χρήσης μεταξύ των μονών και των διπλών εισαγωγικών. Η διαφοροποίηση στη λειτουργικότητα των εισαγωγικών είναι η ακόλουθη:

- Τα μονά εισαγωγικά χρησιμοποιούνται όταν θέλουμε αυτό που υπάρχει μέσα τους να χρησιμοποιηθεί ΟΠΩΣ ακριβώς είναι γραμμένο. Δηλαδή δεν θα γίνει καμία αντικατάσταση.
- Τα διπλά εισαγωγικά χρησιμοποιούνται όταν θέλουμε να γίνει αντικατάσταση των μεταβλητών με τις τιμές τους αν φυσικά υπάρχουν μεταβλητές μέσα σε αυτά. Επίσης «δραπετεύονται» ειδικοί χαρακτήρες που ακολουθούν τον χαρακτήρα `
- Ανάποδα εισαγωγικά χρησιμοποιούνται αν χρειάζεται να εκτελεστεί μια εντολή και να χρησιμοποιηθεί το αποτέλεσμα της εκτέλεσης της. Είναι ισοδύναμα με το \$(command). Δεν προτείνεται η χρήση τους γιατί δημιουργεί δυσανάγνωστο κώδικα.

Ένα παράδειγμα χρήσης των μονών, διπλών και ανάποδων εισαγωγικών παρουσιάζεται στο παρακάτω πλαίσιο. Δημιουργούμε μία μεταβλητή *file* η τιμή της οποίας είναι το όνομα ενός αρχείου που υπάρχει στον τρέχων κατάλογο (*values.dat*). Στην συνέχεια χρησιμοποιούμε την εντολή echo με ένα μήνυμα που βρίσκεται σε μονά, διπλά και ανάποδα εισαγωγικά και περιέχει την τιμή της μεταβλητής *file*. Παρατηρούμε ότι στα μονά εισαγωγικά δεν έγινε αντικατάσταση της τιμής της μεταβλητής *file* ενώ στα διπλά έγινε. Επιπρόσθετα, στα ανάποδα εισαγωγικά έγινε εκτέλεση της εντολής και τυπώθηκαν τα αποτελέσματα της.

```
[user@ekdda scripts]$ ls
first.sh  values.dat
[user@ekdda scripts]$ file=values.dat
[user@ekdda scripts]$ echo 'wc $file'
wc $file
[user@ekdda scripts]$ echo "wc $file"
wc values.dat
[user@ekdda scripts]$ echo `wc $file`
1 5 15 values.dat
```

### Μεταβλητές τοπικές, περιβάλλοντος

Οι μεταβλητές που ορίζουμε σε ένα script ονομάζονται τοπικές γιατί έχουν ισχύ μόνο στο φλοιό στο οποίο τις ορίζουμε. Αν θέλουμε μια μεταβλητή να έχει ισχύ και στους φλοιούς που δημιουργούνται από το shell που ορίζεται (απόγονοι φλοιοί), τότε θα πρέπει να τις ορίσουμε σαν μεταβλητές περιβάλλοντος. Η ιδιότητα αυτή είναι πολύ χρήσιμη γιατί κάθε script που εκτελούμε δημιουργεί έναν δικό του φλοιό στον οποίο στο οποίο και εκτελείται. Επομένως, αν θέλουμε να «περνάμε» μια μεταβλητή, από το εξωτερικό φλοιό που καλέσαμε το script στο φλοιό τον οποίο εκτελείται το script, αυτή η μεταβλητή θα πρέπει να είναι μεταβλητή περιβάλλοντος. Η δημιουργία μιας μεταβλητής περιβάλλοντος γίνεται με τη χρήση της εντολής `export` όπως παρακάτω:

***export όνομα\_μεταβλητής***

Στο παρακάτω πλαίσιο ακολουθεί ένα παράδειγμα όπου μια μεταβλητή περιβάλλοντος «περνάει» την τιμή της στο script ενώ μια τοπική μεταβλητή δεν την περνάει.

```
[user@ekdda scripts]$ local=one #τοπική μεταβλητή
[user@ekdda scripts]$ environ=two
[user@ekdda scripts]$ export environ #μεταβλητή περιβάλλοντος
[user@ekdda scripts]$ cat simple.sh
#!/bin/bash

echo local variable: $local
echo environment variable: $environ
[user@ekdda scripts]$ ./simple.sh
local variable:
enviromental variable: two
```

### 2.2.2 Παράμετροι θέσης

Τα περισσότερα scripts που εκτελούνται στο φλοιό bash δέχονται, μετά το όνομα του script, μία ή περισσότερες παραμέτρους που εξειδικεύουν την εκτέλεση του script. Αυτές οι παράμετροι λέγονται παράμετροι θέσης και χρησιμοποιούνται για το πέρασμα τιμών σε ένα script (BashHackersWiki, n.d.). Για παράδειγμα αν έχουμε δημιουργήσει ένα script με το όνομα `test_parameter` τότε ο χρήστης μπορεί να το καλέσει με την αντίστοιχη εντολή

```
./test_parameter.sh 22 Tsimiski Thessaloniki
```

Αυτό που γίνεται είναι ότι αυτόματα ο φλοιός bash δημιουργεί τις μεταβλητές, θα αναφέρονται σαν παράμετροι θέσης από εδώ και πέρα, με όνομα \$1, \$2, \$3 και τιμές τις 22, Tsimiski και Thessaloniki αντίστοιχα. Ο φλοιός bash φροντίζει να δημιουργήσει τόσες παραμέτρους θέσης όσες είναι και οι τιμές που δίνει ο χρήστης όταν καλεί το script. Προσοχή, για τις παραμέτρους θέσης πάνω από το δέκα θα πρέπει να χρησιμοποιούνται άγκιστρα (π.χ. \${10}) και αν τιμή μιας παραμέτρου θέση έχει κενά θα πρέπει να περαστεί σε διπλά εισαγωγικά. Στο παρακάτω πλαίσιο παρουσιάζεται ένα script που τυπώνει τις παραμέτρους θέσης.

```
1 #!/bin/bash
2
3 echo first parameter: $1
4 echo second parameter: $2
5 echo third parameter: $3
```

Και η εκτέλεση του παρουσιάζεται ακολούθως

```
[user@ekdda scripts]$ ./test_parameter.sh 22 Tsimiski
Thessaloniki
first parameter: 22
second parameter: Tsimiski
third parameter: Thessaloniki
```

Δεν μπορούμε να κάνουμε ανάθεση ή να αλλάξουμε την τιμή σε μια παράμετρο θέσης. Επιπρόσθετα, υπάρχουν και κάποιες ειδικές παράμετροι θέσης με τιμές που αναφέρονται παρακάτω:

- \$0 το όνομα του script (*./test\_parameter.sh*)
- \$# όλες τις παραμέτρους σε ξεχωριστά string ("*22*" "*Tsimiski*" "*Thessaloniki*")
- \$\* όλες οι παράμετροι σε ένα string με κενά ("*22 Tsimiski Thessaloniki*")
- \$# ο αριθμός των παραμέτρων με τις οποίες καλέστηκε το script

### Χρήση αριθμητικών τιμών

Έχει ήδη αναφερθεί ότι ο τύπος όλων των τιμών στο bash είναι κείμενο. Παρόλα αυτά, υπάρχει η δυνατότητα εκτέλεσης αριθμητικών πράξεων αν οι μεταβλητές περιέχουν αριθμητικές τιμές. Ένας τρόπος με τον οποίο μπορεί να εκτελεστεί πράξη μεταξύ δύο αριθμητικών τιμών είναι με τη χρήση της εντολής **let**. Η σύνταξη της εντολής let είναι η παρακάτω:

**let μεταβλητή=μεταβλητή/σταθερά (τελεστής) μεταβλητή/σταθερά**

Η λειτουργία της παραπάνω εντολής είναι η πραγματοποίηση της πράξης μεταξύ των δύο αριθμητικών τιμών και η τοποθέτηση του αποτελέσματος σαν τιμή στη μεταβλητή που

βρίσκεται αριστερά του χαρακτήρα ίσον. Πρέπει να προσέξουμε να μην υπάρχουν κενά ανάμεσα στον αριθμητικό τελεστή και στις μεταβλητές ή τις σταθερές. Επιπρόσθετα, αν και χρησιμοποιούμε την τιμή των μεταβλητών, οι μεταβλητές δεν πρέπει να έχουν τον χαρακτήρα δολάριο πριν από το όνομα τους. Στο παρακάτω πλαίσιο παρουσιάζεται ένα script που υπολογίζει το σύνολο των γραμμών που έχουν δύο αρχεία (*values.dat* και *data.dat*).

```
1 #!/bin/bash
2
3 line1=$(wc -l "values.dat"|cut -f1 -d ' ')
4 line2=$(wc -l "data.dat"|cut -f1 -d ' ')
5
6 let final=line1+line2
7 echo There are $final lines
```

Στο παραπάνω παράδειγμα η εντολή που εκτελεί την αριθμητική πράξη είναι η *let*. Επίσης αν αναλύσουμε και την εντολή στην γραμμή 3 (και η εντολή στη γραμμή 4 είναι παρόμοια). Η εντολή *wc -l "values.dat"* εμφανίζει τον αριθμό μαζί με το όνομα του αρχείου. Επομένως το περνάμε με σωλήνωση (*pipe*) στην εντολή *cut* για να πάρουμε μόνο το πρώτο αριθμό (πεδίο *f1*) χρησιμοποιώντας σαν διαχωριστικό το χαρακτήρα κενό (*-d ' '*).

Εκτός από την εντολή *let* μπορούμε να χρησιμοποιήσουμε και την ακόλουθη σύνταξη για να εκτελέσουμε αριθμητικές πράξεις.

**μεταβλητή=\$( (μεταβλητή/σταθερά (τελεστής) μεταβλητή/σταθερά) )**

Και σε αυτή την περίπτωση βλέπουμε ότι δεν χρειάζεται ο χαρακτήρας δολάριο πριν τα ονόματα των μεταβλητών αλλά μπορούμε να έχουμε κενά ανάμεσα στον αριθμητικό τελεστή και στα ονόματα των μεταβλητών. Το προηγούμενο script θα μπορούσε να δημιουργηθεί τοποθετώντας στη θέση της εντολής *let* την παρακάτω εντολή

```
6 final=$((line1 + line2))
```



Οι αριθμητικές πράξεις που εκτελούνται με τις δύο παραπάνω επιλογές γίνονται μόνο σε ακεραίους. Ωστόσο υπάρχουν επιλογές για όλες τις πράξεις που θέλουμε να εκτελέσουμε ακόμα και για δεκαδικούς αριθμούς (ShellTips, 2020).

### 2.2.3 Πίνακες / συναρτήσεις

#### Πίνακες

Μία από τις προγραμματιστικές δυνατότητες που παρέχει ο φλοιός *bash* είναι και η υποστήριξη πινάκων (Aboukhalil, 2018). Ο ορισμός μιας μεταβλητής πίνακα πραγματοποιείται με την παρακάτω σύνταξη:

**μεταβλητή=( "τιμή1", "τιμή2", ..., "τιμήn" )**

Ενώ μπορεί να γίνει ανάθεση τιμής σε ένα στοιχείο του πίνακα που έχει τη θέση «δείκτης» με την ακόλουθη σύνταξη:

***μεταβλητή[δείκτης]=τιμή***

Αν ο δείκτης υπάρχει τότε γίνεται αντικατάσταση της τιμής που είχε με την καινούργια τιμή. Αν ο δείκτης δεν υπάρχει τότε δημιουργείται με τιμή αυτή που του ανατίθεται. Επίσης, αν ο δείκτης είναι μεγαλύτερος από το τελευταίο δείκτη του πίνακα δημιουργούνται και όλοι οι προηγούμενοι με τιμή κενό. Αν θέλουμε να το προσθέσουμε ακριβώς μετά το τελευταίο στοιχείο του πίνακα χρησιμοποιούμε την ακόλουθη σύνταξη:

***μεταβλητή+=("τιμή")***

Για τη χρήση ενός στοιχείου πίνακα του bash χρησιμοποιούμε την ακόλουθη σύνταξη:

***\${μεταβλητή[δείκτης]}***

Υπάρχουν και δύο ειδικόι τρόποι σύνταξης. Ο πρώτος μας επιστρέφει όλα τα στοιχεία ενός πίνακα ενώ ο δεύτερος των αριθμό των στοιχείων ενός πίνακα και είναι οι παρακάτω:

***\${μεταβλητή[@]} \${#μεταβλητή[@]}***

Στο παρακάτω πλαίσιο παρουσιάζεται το παράδειγμα ενός script που χρησιμοποιεί πίνακες. Αρχικά δημιουργείται ένας κενός πίνακας στον οποίο προστίθενται στοιχεία. Θα μπορούσε η πρόσθεση αυτή να γίνει δυναμικά από τα περιεχόμενα αρχείου και όχι στατικά μέσω κώδικα (θα δούμε παρακάτω στην ανάγνωση αρχείου). Στη συνέχεια παρουσιάζονται όλα τα στοιχεία του πίνακα και τέλος με βάση την επιλογή του χρήστη παρουσιάζεται ένα στοιχείο από όλα αυτά. Παρατηρούμε ότι το πρώτο στοιχείο σε πίνακα στο bash έχει δείκτη μηδέν.

```

1  #!/bin/bash
2
3  mes=()
4  mes+=("1.Insert")
5  mes+=("2.Delete")
6  mes+=("3.Edit")
7  mes+=("4.Exit")
8
9  clear
10 echo ${mes[@]}
11 read -p 'Give selection [1-4]:' sel
12 sel=$((sel-1))
13 echo Selected ${mes[$sel]}lines

```



Οι συναρτήσεις στο bash είναι ομάδες κώδικα που έχουν ένα όνομα το οποίο πρέπει να είναι μοναδικό (Linuxize, 2020). Η δημιουργία μιας συνάρτησης μπορεί να γίνει με ένα από τους παρακάτω τρόπους σύνταξης:

```
function όνομα_συνάρτησης {εντολές;}
```

```
όνομα_συνάρτησης () {εντολές;}
```

Η δημιουργία μιας συνάρτησης δεν συνεπάγεται την εκτέλεση της. Η εκτέλεση του κώδικα μιας συνάρτησης πραγματοποιείται αν γράψουμε το όνομα της. Δεν δημιουργείται νέος φλοιός, όπως π.χ. στα script, αλλά οι εντολές της συνάρτησης εκτελούνται στο ίδιο φλοιό με το script που την κάλεσε. Στο παρακάτω πλαίσιο εμφανίζεται ένα script το οποίο ορίζει μία συνάρτηση **showinfo** που εμφανίζει πληροφορίες σχετικές με το υπολογιστικό σύστημα στο οποίο εκτελείται:

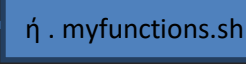
```
1 #!/bin/bash
2
3 function showinfo {
4     cpu=$(lscpu | sed -n '14 p' | cut -f2 -d: | xargs)
5     mem=$(lsmem | sed -n '12 p' | cut -f2 -d: | xargs)
6     echo "** System Info **"
7     echo CPU: $cpu
8     echo Memory: $mem
9     echo OS: $(uname -s)
10 }
11
12 showinfo
```

Προσοχή πρέπει να υπάρχει κενό

Βλέπουμε ότι η συνάρτηση περιλαμβάνει τις εντολές που αναγράφονται στις γραμμές 3 με 10. Ακόμα η κλήση της συνάρτησης γίνεται με την απλή αναγραφή του ονόματός της. Μια μικρή εξήγηση χρειάζεται η γραμμή 4 όπου αυτό που πραγματοποιείται είναι το εξής. Αρχικά εκτελείται η εντολή **lscpu**. Αυτή η εντολή παράγει πολλές γραμμές σαν έξοδο από τις οποίες διαλέγουμε και κρατάμε μόνο τη γραμμή 14 με τη χρήση του **sed**. Στη συνέχεια από τη γραμμή 14 επιλέγουμε μόνο το δεύτερο μέρος με διαχωριστικό τον χαρακτήρα άνω κάτω τελεία (:). Τέλος η εντολή **xargs** διαγράφει τα κενά που έχει το τμήμα του κειμένου που κρατήσαμε.

Μία συνάρτηση μπορεί να οριστεί στο ίδιο αρχείο με το script που την καλεί ή μπορεί να οριστεί και σε διαφορετικό αρχείο. Στο προηγούμενο παράδειγμα θα μπορούσαμε να ορίσουμε τη συνάρτηση **showinfo** σε ένα αρχείο που περιέχει διάφορες συναρτήσεις και ονομάζεται **myfunctions.sh**. Στη συνέχεια θα μπορούσαμε να χρησιμοποιήσουμε τη συνάρτηση από οποιοδήποτε script με τη χρήση της εντολής **source** όπως στο παρακάτω πλαίσιο:

```
1 #!/bin/bash
2
3 source myfunctions.sh
4
5 showinfo
```



Οι μεταβλητές που δημιουργούμε στο φλοιό bash είναι καθολικές (**global**) σε ότι αφορά την εμβέλεια τους. Δηλαδή κάθε μεταβλητή που ορίζουμε μπορούμε να τη χρησιμοποιήσουμε σε οποιαδήποτε επόμενη εντολή ανεξάρτητα αν αυτή είναι μέσα σε συνάρτηση ή όχι. Αν θέλουμε οι μεταβλητές που ορίζουμε μέσα σε μία συνάρτηση να έχουν εμβέλεια μόνο τη συνάρτηση αυτή τότε πριν τον ορισμό τους θα πρέπει να προηγείται η λέξη **local** (**τοπική**). Στο παρακάτω πλαίσιο εμφανίζεται ένα παράδειγμα με τη δημιουργία και χρήση τοπικών και καθολικών μεταβλητών.

```
[user@ekdda scripts]$ cat locvar.sh
#!/bin/bash
function inner {
    local locvar=hidden
    glovar=visible
}
inner
echo Local variable: $locvar
echo Global variable: $glovar

[user@ekdda scripts]$ ./locvar.sh
Local variable:
Global variable: visible
```

Η εκτέλεση μιας συνάρτησης γίνεται όπως η εκτέλεση ενός script με τη διαφορά ότι πραγματοποιείται στο ίδιο φλοιό με το script που την κάλεσε και όχι σε καινούργιο. Για να «περάσουμε» δεδομένα από το εξωτερικό script στη συνάρτηση που καλούμε μπορούμε είτε να χρησιμοποιήσουμε μεταβλητές του script είτε να χρησιμοποιήσουμε τον μηχανισμό των παραμέτρων θέσης που περιγράψαμε σε προηγούμενες σελίδες. Το παρακάτω πλαίσιο περιγράφει ένα script που καλεί μία συνάρτηση που δέχεται σαν παράμετρο θέσης το όνομα ενός χρήστη (π.χ. user) και χρησιμοποιεί από το script που την κάλεσε τον όνομα ενός καταλόγου (π.χ. scripts). Η συνάρτηση υπολογίζει και τυπώνει τον αριθμό των κανονικών αρχείων που περιέχει αυτός ο κατάλογος του χρήστη (δηλαδή ο κατάλογος /home/user/scripts).

```
1 #!/bin/bash
2 function showfiles {
3     user=$1
4     mypath="/home/$user/$folder"
5     files=$(find $mypath -type f | wc -l)
6     echo There are $files regular files in $mypath
7 }
8 folder=scripts
9 showfiles user
```

Για να «επιστρέψουμε» τιμές από μια συνάρτηση που καλούμε σε ένα script στο ίδιο το script μπορούμε να χρησιμοποιήσουμε διάφορες τεχνικές. Μία τεχνική είναι να αναθέσουμε τις τιμές που θέλουμε να επιστρέψουμε στο script σε καθολικές μεταβλητές. Ενώ η μία δεύτερη τεχνική είναι να τυπώσουμε την τιμή που θέλουμε να επιστρέψουμε και στο script να χρησιμοποιήσουμε την τεχνική της αντικατάστασης εντολής<sup>5</sup>.

```
1 #~/bin/bash
2 function userinfo {
3     lastuser=$( tail -1 /etc/passwd | cut -f1 -d: )
4 }
5 userinfo
6 echo Last user is $lastuseruser

1 #~/bin/bash
2 function userinfo {
3     local lastuser=$( tail -1 /etc/passwd | cut -f1 -d: )
4     echo $lastuser
5 }
6 user=$(userinfo)
7 echo Last user is $user
```

Στο παραπάνω πλαίσιο βλέπουμε μια συνάρτηση που επιστρέφει το όνομα του χρήστη που δημιουργήθηκε τελευταίος στο σύστημα μας. Την πρώτη φορά το επιστρέφει μέσω καθολικής μεταβλητής ενώ τη δεύτερη με τη χρήση του μηχανισμού της αντικατάστασης εντολής.

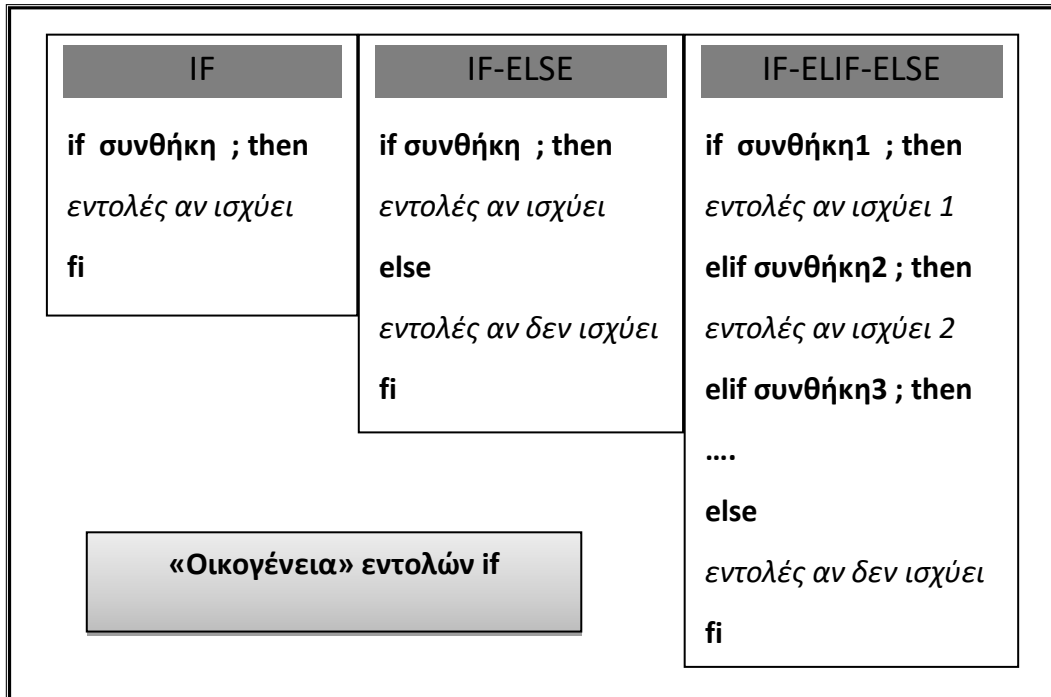
### 2.3 Δομές προγραμματισμού στο bash

Δύο βασικά χαρακτηριστικά μιας γλώσσας προγραμματισμού που υποστηρίζονται και από το bash είναι η δομή επιλογής και η δομή επανάληψης. Με τη χρήση της δομής επιλογής η εκτέλεση ή μη μιας ομάδας εντολών, εξαρτάται από την αποτίμηση μιας συνθήκης. Επιπρόσθετα, με τη δομή επανάληψης δίνεται η δυνατότητα επανάληψης μιας ομάδας εντολών είτε για κάποιο συγκεκριμένο αριθμό επαναλήψεων είτε για όσο καιρό ισχύει μία συνθήκη.

<sup>5</sup> Προσοχή δεν μπορούμε να χρησιμοποιήσουμε τις δύο τεχνικές μαζί

### 2.3.1 Δομή επιλογής

Το bash υποστηρίζει αρκετές εντολές που ανήκουν στη δομή επιλογής με βασικό εκπρόσωπο την «οικογένεια» των εντολών if όπως παρουσιάζονται στην Εικόνα 2- 1. Όπως παρατηρούμε βασικό στοιχείο στις εντολές επιλογής είναι ο ορισμός και η χρήση της συνθήκης.



Εικόνα 2- 1: Η "οικογένεια" των εντολών if

Τα στοιχεία που μπορούν να χρησιμοποιηθούν σαν συνθήκη στο bash είναι τα παρακάτω (James, 2012):

- Επιτυχής εκτέλεση εντολής
- Έλεγχος τιμής (αριθμητικής ή κειμένου)
- Έλεγχος ιδιοτήτων αρχείου / καταλόγου

Στη συνέχεια γίνεται μια λεπτομερέστερη αναφορά στις κατηγορίες συνθηκών που περιγράφονται

#### Επιτυχής εκτέλεση εντολής

Μετά την εκτέλεσή της, κάθε μία εντολή του φλοιού bash, επιστρέφει μέσω της μεταβλητής **\$?** έναν αριθμό που χαρακτηρίζει το εάν εκτελέστηκε επιτυχημένα ή όχι. Αυτός ο αριθμός ονομάζεται **κατάσταση εξόδου (exit status)** της εντολής και είναι μηδέν (0) αν η εντολή έχει εκτελεστεί επιτυχημένα ενώ είναι ένας αριθμός διαφορετικός του μηδενός ανάλογα με τον λόγο της αποτυχίας εκτέλεσης. Στα scripts που δημιουργούμε έχουν σαν κατάσταση εξόδου την κατάσταση εξόδου της τελευταίας εντολής εκτός αν χρησιμοποιήσουμε την εντολή **exit**. Μια παρουσίαση της κατάστασης εξόδου διάφορων εντολών εμφανίζεται στο παρακάτω πλαίσιο:

```
[user@ekdda scripts]$ ls values.dat
values.dat
[user@ekdda scripts]$ wc -l values.dat
4 values.dat
[user@ekdda scripts]$ echo $?
0
[user@ekdda scripts]$ ls noexist.dat
ls: cannot access 'noexist.dat': No such file or directory
[user@ekdda scripts]$ echo $?
2
```

Όπως παρατηρούμε, στην τελευταία εντολή `ls` όπου δεν υπήρχε το αρχείο που προσπαθήσαμε να εμφανίσουμε, εκτός από το να εμφανιστεί μήνυμα λάθους, ανατέθηκε και μη μηδενική τιμή στη μεταβλητή της κατάστασης εξόδου (`$?`).

Το αποτέλεσμα εκτέλεσης μιας εντολής μπορεί να το χρησιμοποιήσει στη θέση της συνθήκης η εντολή `if`. Στο παρακάτω παράδειγμα η εντολή `if` ελέγχει αν υπάρχει ο χρήστης που βρίσκεται στη μεταβλητή `client`. Αν ο χρήστης υπάρχει τότε εμφανίζει τα περιεχόμενα που έχει στον κατάλογο `/home/$client`. Αν ο χρήστης δεν υπάρχει τότε δεν υπάρχει εμφανίζει μήνυμα.

```
1 #!/bin/bash
2
3 read -p 'Give user:' client
4
5 if id $client >/dev/null 2>&1 ; then
6     ls /home/$client
7 else
8     echo User $client does not exist
9 fi
```

Στο παραπάνω script ενδιαφέρον παρουσιάζει η εντολή `id $client >/dev/null 2>&1`. Εδώ αυτό που γίνεται είναι ότι θέλουμε μόνο την κατάσταση εξόδου της εντολής χωρίς να θέλουμε να εμφανιστούν στην οθόνη τα αποτελέσματα που δίνει η εντολή ούτε τα αποτελέσματα λάθους. Γι' αυτό τον λόγο ανακατευθύνουμε τόσο την κανονική έξοδο όσο και την έξοδο λαθών σε ένα ειδικό αρχείο (`/dev/null`) που αποτελεί τον «κάδο αχρήστων» του λειτουργικού συστήματος Linux.



Περισσότερα για την ανακατεύθυνση εισόδου – εξόδου μπορούμε να δούμε στο (Raevskiy, 2020)

## Έλεγχος τιμής

Ο φλοιός bash διαθέτει την εντολή **test** η οποία δέχεται μια λογική έκφραση και ελέγχει αν αυτή η λογική έκφραση είναι αληθής ή ψευδής. Το αποτέλεσμα του ελέγχου επιστρέφεται με το αποτέλεσμα εκτέλεσης της εντολής **test**. Η γενική μορφή της εντολής **test** είναι η παρακάτω:

***test λογική\_έκφραση***

Όπου λογική έκφραση είναι η πράξη ενός τελεστή πάνω σε ένα ή δύο αντικείμενα που επιστρέφει μια λογική τιμή (αληθές, ψευδές). Επειδή η πιο συνηθισμένη χρήση της **test** ήταν η χρήση της σαν συνθήκη στην εντολή **if** αναπτύχθηκε μια νέα μορφή με σκοπό να κάνει πιο ευανάγνωστη την εντολή **if**. Η καινούργια μορφή της εντολής **test** χρησιμοποιεί τους χαρακτήρες αγκύλες και είναι η παρακάτω:

**[ αντικείμενο1 τελεστής αντικείμενο2 ]**

**[ τελεστής αντικείμενο ]**

Προσοχή, πρέπει να υπάρχουν κενά ανάμεσα α) στον τελεστή και τα αντικείμενα β) στα αντικείμενα και τις αγκύλες. Μερικοί από τους βασικούς ελέγχους που υποστηρίζει η εντολή **test** βρίσκονται στον Πίνακας 2- 1 που ακολουθεί:

| Έκφραση                           | Λειτουργικότητα  |
|-----------------------------------|--|
| <b>! expression</b>               | <i>expression</i> είναι ψευδής                         |
| <b>expression1 -a expression2</b> | <i>expression1</i> και <i>expression2</i> είναι αληθής |
| <b>expression1 -o expression2</b> | <i>expression1</i> η <i>expression2</i> είναι αληθής   |
| <b>-n string</b>                  | το μήκος του <i>string</i> είναι μη μηδενικό           |
| <b>String</b>                     | ισοδύναμο με το <b>-n string</b>                       |
| <b>-z string</b>                  | το μήκος του <i>string</i> είναι μηδέν                 |
| <b>string1 = string2</b>          | τα <i>strings</i> είναι ίδια                           |
| <b>string1 != string2</b>         | τα <i>strings</i> είναι διαφορετικά                    |
| <b>integer1 -eq integer2</b>      | <i>integer1</i> είναι ίσο με <i>integer2</i>           |
| <b>integer1 -ge integer2</b>      | <i>integer1</i> μεγαλύτερο ή ίσο από <i>integer2</i>   |
| <b>integer1 -gt integer2</b>      | <i>integer1</i> μεγαλύτερο από <i>integer2</i>         |
| <b>integer1 -le integer2</b>      | <i>integer1</i> μικρότερο ή ίσο από <i>integer2</i>    |
| <b>integer1 -lt integer2</b>      | <i>integer1</i> μικρότερο από <i>integer2</i>          |
| <b>integer1 -ne integer2</b>      | <i>integer1</i> είναι διαφορετικό από <i>integer2</i>  |

Πίνακας 2- 1: Βασικοί έλεγχοι της εντολής **test**

Ένας συνηθισμένος έλεγχος που κάνουμε είναι στο αν μία μεταβλητή έχει οριστεί ή όχι. Αυτό μπορεί να γίνει με τη χρήση της εντολής **test** είτε χρησιμοποιώντας την παράμετρο **z** είτε την παράμετρο **n**. Μία από τις δύο επιλογές παρουσιάζεται στο παρακάτω script:

```

1 #!/bin/bash
2 a=something
3 if [ -z $a ]; then
4     echo a is not set;
5 else
6     echo a is set;
7 fi

```

Παρατηρούμε ότι όταν θέλουμε να χρησιμοποιήσουμε μια μεταβλητή μέσα στην εντολή `test` θα πρέπει να χρησιμοποιήσουμε τον χαρακτήρα δολάριο (\$) πριν από το όνομα της.

Ένα χαρακτηριστικό παράδειγμα χρήσης αριθμητικού ελέγχου είναι ένα script το οποίο για να εκτελεστεί θέλει απαραίτητα μια παράμετρο. Στο παρακάτω πλαίσιο παρουσιάζεται ένα script το οποίο ελέγχει αν έχει δοθεί μία παράμετρος και στην περίπτωση που δεν έχει δοθεί τότε εμφανίζει τη χρήση του script με τη βοήθεια μιας συνάρτησης.

```

1 #!/bin/bash
2
3 function usage {
4     echo Usage: ./checkparam.sh option
5 }
6
7 if [ $# -ne 1 ]; then
8     usage
9 else
10    echo You passed $1 parameter
11 fi

```

Προσοχή πρέπει να υπάρχει κενό

Για να ελέγξουμε μεταβλητές που περιέχουν κείμενα μπορούμε να χρησιμοποιήσουμε την εντολή `test [ ]` αλλά θα πρέπει τα ονόματα των μεταβλητών να περιέχονται μέσα σε διπλά εισαγωγικά. Αυτό γίνεται γιατί σε περίπτωση που η κάποια μεταβλητή από αυτές που χρησιμοποιούνται στην `test` δεν έχει τιμή τότε θα δημιουργηθεί συντακτικό λάθος. Στο παρακάτω πλαίσιο παρουσιάζεται ένα παράδειγμα όπου ελέγχεται αν ο χρήστης είναι ο `root` ή όχι. Το script ελέγχει αν το αποτέλεσμα της εντολής `whoami` είναι η λέξη `root`.

```

1 #!/bin/bash
2
3 luser=$(whoami)
4
5 if [ "$luser" == "root" ]; then
6     echo Hello root
7 else
8     echo You are not root!
9 fi

```

Επίσης υπάρχει και η έκδοση της `test` που χρησιμοποιεί διπλές αγκύλες<sup>6</sup>. Σε αυτή την περίπτωση δεν χρειάζεται να βάζουμε διπλά εισαγωγικά στα ονόματα των μεταβλητών καθώς μπορούμε να χρησιμοποιήσουμε και **κανονικές εκφράσεις (regular expressions)**<sup>7</sup>. Ένα παράδειγμα ελέγχου με τη χρήση κανονικών εκφράσεων παρουσιάζεται στο παρακάτω πλαίσιο:

```
1 #!/bin/bash
2
3 value=$1
4
5 if [[ $value =~ ^[0-9]{9}$ ]]; then
6     echo Valid vat number
7 else
8     echo No valid vat number
9 fi
```

Στο παραπάνω παράδειγμα ελέγχουμε αν η παράμετρος θέσης που δίνει ο χρήστης είναι ένας αριθμός ΑΦΜ (δηλαδή αποτελείται από 9 ψηφία). Παρατηρούμε ότι ο τελεστής που ελέγχει αν ένα string ταιριάζει με μία κανονική έκφραση είναι ο (`=~`).

### Έλεγχος ιδιοτήτων αρχείου / καταλόγου

Μία δυνατότητα της `test` που χρησιμοποιείται πολύ συχνά στον προγραμματισμό στο `bash` είναι ο έλεγχος πάνω στις ιδιότητες ενός αρχείου ή καταλόγου. Οι ιδιότητες που μπορούν να ελεγχτούν παρουσιάζονται στον παρακάτω Πίνακας 2- 2:

| Έκφραση                      | Λειτουργικότητα  |
|------------------------------|--|
| <code>name1 -ef name2</code> | <code>name1</code> και <code>name2</code> υπάρχουν στο ίδιο device και έχουν τον ίδιο inode αριθμό |
| <code>name1 -nt name2</code> | Το <code>name1</code> έχει τροποποιηθεί πιο πρόσφατα από το <code>name2</code>                     |
| <code>name1 -ot name2</code> | Το <code>name1</code> έχει τροποποιηθεί παλιότερα από το <code>name2</code>                        |
| <code>-b name</code>         | <code>name</code> είναι αρχείο τύπου block   |
| <code>-c name</code>         | <code>name</code> είναι αρχείο τύπου character   |
| <code>-d name</code>         | <code>name</code> είναι κατάλογος  |
| <code>-e name</code>         | <code>name</code> υπάρχει  |
| <code>-f name</code>         | <code>name</code> είναι κανονικό αρχείο  |
| <code>-g name</code>         | <code>name</code> έχει ενεργοποιημένο στο set group id SGID  |
| <code>-G name</code>         | <code>name</code> ανήκει στο ίδιο effective group ID   |
| <code>-h name</code>         | <code>name</code> είναι symbolic link (ίδιο με <code>-L</code> )                                   |
| <code>-k name</code>         | <code>name</code> έχει ενεργοποιημένο το sticky bit  |
| <code>-L name</code>         | (ίδιο με <code>-h</code> )   |
| <code>-O name</code>         | <code>name</code> ανήκει στο ίδιο effective user ID  |
| <code>-p name</code>         | <code>name</code> είναι named pipe   |
| <code>-r name</code>         | <code>name</code> έχει ενεργοποιημένο το read permission   |
| <code>-s name</code>         | <code>name</code> έχει μέγεθος μεγαλύτερο από μηδέν  |

<sup>6</sup> Υπάρχει και ο έλεγχος με διπλές παρενθέσεις για αριθμητικούς ελέγχους

<sup>7</sup> Περισσότερα για τις κανονικές εκφράσεις στην συνέχεια του κειμένου.



|                  |  |
|------------------|--|
| <b>-S name</b>   | <i>name</i> υπάρχει και είναι socket                           |
| <b>-t number</b> | Ο <i>number</i> έχει ανοιχτεί σαν file descriptor              |
| <b>-u name</b>   | <i>name</i> υπάρχει και έχει ενεργοποιημένο το set-user-ID bit |
| <b>-w name</b>   | <i>name</i> έχει ενεργοποιημένο το write permission            |
| <b>-x name</b>   | <i>name</i> έχει ενεργοποιημένο το execute permission          |

Πίνακας 2- 2: Έλεγχοι της εντολής test σε αρχεία

Ένα παράδειγμα χρήσης των ελέγχων σε ιδιότητες αρχείων ή καταλόγων παρουσιάζεται στο παρακάτω πλαίσιο. Εδώ ο χρήστης περνάει σαν παράμετρο το όνομα ενός καταλόγου. Αν ο χρήστης δεν περάσει παράμετρο τότε παίρνει προκαθορισμένη τιμή το όνομα "backup". Στη συνέχεια το script ελέγχει αν υπάρχει αυτός ο κατάλογος στον κατάλογο ανωτέρου επιπέδου του user και αν έχει αυτός δικαιώματα εγγραφής σε αυτό. Σε περίπτωση που ισχύουν και τα δύο τότε δημιουργεί ένα αρχείο με το όνομα "data.log" σε αυτόν. Σε διαφορετική περίπτωση εμφανίζει μήνυμα λάθους.

```

1 #!/bin/bash
2
3 dir=$1
4 dir=${dir:="backup"}
5
6 if [ -d "/home/user/$dir" -a -w "/home/user/$dir" ]; then
7     touch /home/user/$dir/data.log
8     echo data.log file created
9     exit 0
10 else
11     echo error in creating /home/${dir}/data.log file
12     exit 1
13 fi

```



Μία από τις εντολές επιλογής που διαθέτει ο bash είναι και η case (Yesmin, 2018) που δεν παρουσιάζεται για λόγους οικονομίας χώρου.

### 2.3.2 Δομή επανάληψης

Δύο είναι οι βασικές εντολές επανάληψης στον φλοιό bash οι while και for. Στην while η επανάληψη των εντολών συνεχίζεται όσο ισχύει μια συνθήκη ενώ στην for ο αριθμός επαναλήψεων είναι προκαθορισμένος.

#### Εντολή while

Η εντολή **while** (Garrels, 2008) επιτρέπει την εκτέλεση μιας ή περισσότερων εντολών όσο ισχύει μία συνθήκη. Η σύνταξη της εντολής while είναι η παρακάτω

```

while συνθήκη; do
    εντολές_που_επαναλαμβάνονται

```

### *done*

Σαν λογική συνθήκη μπορεί να χρησιμοποιηθεί οποιαδήποτε συνθήκη που θα χρησιμοποιούταν και στην εντολή `if`. Ενώ οι εντολές που επαναλαμβάνονται είναι ανάμεσα στις λέξεις **`do`** και **`done`**.

Πολύ συνηθισμένη χρήση της `while` είναι η ανάγνωση δεδομένων από το πληκτρολόγιο. Σε αυτή την περίπτωση σαν συνθήκη θα χρησιμοποιηθεί η εντολή `read`. Η `read` έχει μη μηδενική κατάσταση εξόδου σε περίπτωση που ο χρήστης χρησιμοποιήσει τον συνδυασμό πλήκτρων `control` και `D`. Το script που παρουσιάζεται στο επόμενο πλαίσιο διαβάζει την είσοδο του χρήστη την οποία και τυπώνει σε ένα αρχείο με πρόθεμα τον αριθμό γραμμής.

```
1 #!/bin/bash
2
3 echo Give lines and press enter
4 echo Press control-D to stop
5 count=0
6 while read lineoftext; do
7     let count=$count+1
8     echo ${count}: $lineoftext >> mytext.txt
9 done
10 cat mytext.txt
```

Η `while` μπορεί να χρησιμοποιηθεί και στην περίπτωση που η ανάγνωση των δεδομένων δεν γίνεται από το πληκτρολόγιο αλλά από ένα αρχείο. Σε αυτή την περίπτωση μπορεί να χρησιμοποιηθεί η `read` σε συνδυασμό με την ανακατεύθυνση εισόδου όπως παρουσιάζεται στο παρακάτω πλαίσιο:

```
1 #!/bin/bash
2 count=0
3 while read lineoftext; do
4     let count=$count+1
5     echo ${count}: $lineoftext >> mytext.txt
6 done < myinput.txt
7 echo ----- mytext.txt -----
8 cat mytext.txt
```

### Η εντολή `for`

Η εντολή `for` επαναλαμβάνει κάποιες εντολές για συγκεκριμένο αριθμό επαναλήψεων. Συγκεκριμένα, στην `for` υπάρχει μια μεταβλητή που σε κάθε επανάληψη παίρνει μία τιμή από μία λίστα τιμών που περιέχονται σε ένα `string`. Η σύνταξη της εντολή `for` είναι η ακόλουθη:

*`for όνομα in λίστα_με_τιμές; do`*

### *εντολές\_που\_επαναλαμβάνονται*

#### *done*

Κατά την εκτέλεση της `for` η μεταβλητή “όνομα” παίρνει μία-μία τις τιμές που περιέχονται στη λίστα `_με_τιμές`. Για κάθε μία από τις τιμές αυτές εκτελούνται οι εντολές που υπάρχουν ανάμεσα στις λέξεις `do` και `done`.

Μία εντολή που χρησιμοποιείται συχνά σε συνδυασμό με την εντολή `for` είναι η εντολή `seq` που παράγει μια λίστα συνεχόμενων τιμών. Ένα παράδειγμα παρουσιάζεται στο επόμενο πλαίσιο όπου χρησιμοποιείται η εντολή `for` σε συνδυασμό με την `seq` για τη δημιουργία πολλών αρχείων.

```
1 #!/bin/bash
2
3 for name in {1..20}; do
4     touch data/file${name}.txt
5 done
```

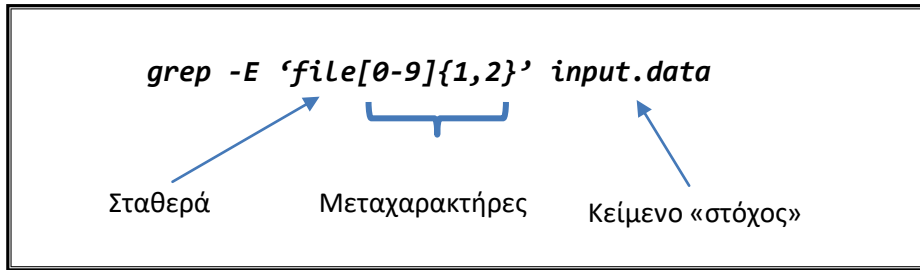
## 2.4 Μελέτη περιπτώσεων ανάλυσης αρχείων κειμένου με χρήση εργαλείων

Ένα από τα ισχυρά εργαλεία που διαθέτει ο προγραμματισμός στον φλοιό `bash` είναι οι κανονικές εκφράσεις (**regular expressions regexs**) (Friedl, 2006). Με τον όρο κανονική έκφραση αναφερόμαστε σε ομάδα χαρακτήρων που περιγράφουν μοτίβα κειμένου. Οι κανονικές εκφράσεις στο `bash` χρησιμοποιούνται από διάφορα εργαλεία όπως η εντολή `grep`, ο επεξεργαστής κειμένου `sed`, το `vi` αλλά και ο επεξεργαστής κειμένου `awk`. Στη συνέχεια θα αναφερθούμε στη δημιουργία κανονικών εκφράσεων μέσα από την εντολή `grep` καθώς θα περιγράψουμε τα βασικά χαρακτηριστικά της λειτουργίας τόσο του `sed` όσο και του `awk`.

### 2.4.1 Κανονικών εκφράσεων

Η λειτουργία των εργαλείων που χρησιμοποιούν κανονικές εκφράσεις είναι η ακόλουθη: Παίρνουν μία γραμμή από το κείμενο που θέλουμε να αναζητήσουμε και δημιουργούν ένα αντίγραφο από αυτή. Στη συνέχεια ελέγχουν αν αυτή η γραμμή περιέχει το «μοτίβο» που περιγράφει η κανονική έκφραση. Αν ναι τότε εκτελούν μια ενέργεια. Επαναλαμβάνουν τη διαδικασία, μία – μία γραμμή, για όλο το κείμενο που θέλουμε να επεξεργαστούμε.

Το εργαλείο του `bash` που θα χρησιμοποιήσουμε για να εξερευνήσουμε τις κανονικές εκφράσεις είναι η εντολή **grep**. Η εντολή `grep` δέχεται μία κανονική έκφραση μαζί με ένα ή περισσότερα αρχεία και εμφανίζει τις γραμμές των αρχείων αυτών που ταιριάζουν με το «μοτίβο» που περιγράφει η κανονική έκφραση. Μια κανονική έκφραση είναι μια ομάδα χαρακτήρων αποτελείται από χαρακτήρες «σταθερές» (**literal**) που θέλουμε να βρούμε αυτούσιους στο κείμενο και **μεταχαρακτήρες (metacharacters)** όπου δεν θέλουμε να τους βρούμε αυτούσιους αλλά περιγράφουν κάτι διαφορετικό για την αναζήτηση. Η μορφή μιας αναζήτησης με τη χρήση της `grep` παρουσιάζεται στην Εικόνα 2- 2:



Εικόνα 2- 2: Παράδειγμα χρήσης της grep

Εδώ βλέπουμε ότι οι χαρακτήρες “file” είναι σταθερές αφού αυτούς τους χαρακτήρες θέλουμε να βρούμε στο κείμενο ενώ οι χαρακτήρες “[0-9]{1,2}” είναι μεταχαρακτήρες επειδή δεν ψάχνουμε να βρούμε τους ίδιους τους χαρακτήρες αλλά δηλώνουν κάτι για την αναζήτηση μας που θα εξηγήσουμε στη συνέχεια. Το αρχείο “input.data” θα ελεγχθεί για το εάν περιέχει τις κανονικές εκφράσεις ενώ η παράμετρος -E πληροφορεί την grep ότι θα χρησιμοποιήσουμε το εκτεταμένο σεν κανονικών εκφράσεων.



Η grep μπορεί να πάρει πολλές παραμέτρους που να τροποποιήσουν τη λειτουργικότητα της (Linuxize-Grep, 2019). Επίσης υπάρχει το σύνολο των βασικών όσο και το σύνολο των εκτεταμένων (extended) μεταχαρακτήρων. Η χρήση της παραμέτρου E στην grep ορίζει ότι η αναζήτηση μας περιέχει μεταχαρακτήρες που ανήκουν στο σύνολο των εκτεταμένων.

Οι μεταχαρακτήρες χωρίζονται σε τρεις κατηγορίες ανάλογα με το είδος της τροποποίησης που προκαλούν. Οι κατηγορίες αυτές είναι:

- **Μπαλαντέρ (wildcards)** είναι οι χαρακτήρες που χρησιμοποιούνται στη θέση άλλων χαρακτήρων.
- **Τροποποίησης (modifiers)** είναι οι χαρακτήρες που καθορίζουν το πλήθος εμφανίσεων του προηγούμενο χαρακτήρα.
- **Θέσης (anchors)** είναι οι χαρακτήρες που καθορίζουν τη θέση ως αναφορά τη γραμμή ή τη λέξη της κανονικής έκφρασης.

Στη συνέχεια θα παρουσιάσουμε τους χαρακτήρες που ανήκουν στην καθεμία από τις παραπάνω κατηγορίες μαζί με παραδείγματα χρήσης τους με την grep.

### Μεταχαρακτήρες WildCards

Οι μεταχαρακτήρες που ανήκουν στην κατηγορία των wildcards στην πραγματικότητα περιγράφουν άλλους χαρακτήρες. Για παράδειγμα όταν θέλουμε να πούμε ότι ψάχνουμε να βρούμε οποιοσδήποτε χαρακτήρα τότε τοποθετούμε τον χαρακτήρα τελεία (.). Ο παρακάτω πίνακας παρουσιάζει όλους τους μεταχαρακτήρες wildcards μαζί με τη λειτουργικότητα τους.

| Μεταχαρακτήρας | Λειτουργικότητα                          |
|----------------|--|
| .              | Ένας οποιοσδήποτε χαρακτήρας             |
| [abh]          | Ένας από τους χαρακτήρες που περιέχονται |

|           |  |
|-----------|--|
| [a-g]     | Ένας από τους χαρακτήρες από a μέχρι g               |
| [^abc]    | Ένας χαρακτήρας εκτός από αυτούς που περιέχονται     |
| [:digit:] | Ένα ψηφίο  |
| [:alnum:] | Ένας χαρακτήρας από a-z ή A-Z ή ψηφίο                |
| [:alpha:] | Ένας χαρακτήρας από a-z ή A-Z                        |
| [:blank:] | Κενό ή tab   |
| [:punct:] | .,'"?!,:#\$%&()*+~/<>=@[]\^_{} ~                     |
| [:space:] | Ένας μη εκτυπώσιμος χαρακτήρας                       |
| [:upper:] | Ένας χαρακτήρας από A-Z                              |
| [:lower:] | Ένας χαρακτήρας από a-z                              |
| \         | Αναιρεί την ιδιότητα του μεταχαρακτήρα που ακολουθεί |

Πίνακας 2- 3: Οι μεταχαρακτήρες wildcards

Ακολουθούν μερικά παραδείγματα κανονικών εκφράσεων που περιέχουν wildcard μεταχαρακτήρες μαζί με την εξήγησή τους και την εκτέλεση τους:

- '[sd]' Ένας από τους χαρακτήρες s ή d που ακολουθείται από τον χαρακτήρα κενό
- '[k-p]w' Ένας από τους χαρακτήρες από το k μέχρι το p που ακολουθείται από τον χαρακτήρα w
- '[:upper:]' Ένας οποιοσδήποτε κεφαλαίος χαρακτήρας
- 'f..' ο χαρακτήρας f ακολουθούμενος από δύο οποιοσδήποτε χαρακτήρες και ακολουθούμενο από το κενό
- '[^e]' οποιοσδήποτε χαρακτήρας εκτός από τον e ακολουθούμενος από το κενό
- '\.txt' ο χαρακτήρας . ακολουθούμενος από τους χαρακτήρες txt

```
[user@ekdda regexs]$ cat input.data
this is a file and a new file22
power for the old and Test file3txt
old3 down the hill file.txt
file1 forever increase
[user@ekdda regexs]$ grep -E '[sd]' input.data
this is a file and a new file22
power for the old and Test file3txt
[user@ekdda regexs]$ grep -E '[k-p]w' input.data
power for the old and Test file3txt
old3 down the hill file.txt
[user@ekdda regexs]$ grep -E '[:upper:]' input.data
power for the old and Test file3txt
[user@ekdda regexs]$ grep -E 'f..' input.data
power for the old and Test file3txt
[user@ekdda regexs]$ grep -E '[^e]' input.data
this is a file and a new file22
power for the old and Test file3txt
old3 down the hill file.txt
file1 forever increase
[user@ekdda regexs]$ grep -E '\.txt' input.data
old3 down the hill file.txt
```

## Μεταχαρακτήρες Modifier

Οι μεταχαρακτήρες modifier έχουν σαν σκοπό να περιγράψουν πόσες επαναλήψεις του προηγούμενου χαρακτήρα αναζητούμε. Για παράδειγμα, αν θέλουμε να πούμε ότι ψάχνουμε να βρούμε τον χαρακτήρα *e* μία ή περισσότερες φορές συνεχόμενες τότε γράφουμε *e+*. Ο παρακάτω πίνακας παρουσιάζει όλους τους μεταχαρακτήρες modifier μαζί με τη λειτουργικότητά τους.

| Μεταχαρακτήρας | Λειτουργικότητα                                     |
|----------------|---|
| ?              | Ο προηγούμενος χαρακτήρας μηδέν ή μία φορά          |
| *              | Ο προηγούμενος χαρακτήρας μηδέν ή άπειρες φορές     |
| +              | Ο προηγούμενος χαρακτήρας μία ή άπειρες φορές       |
| {i}            | Ο προηγούμενος χαρακτήρας ακριβώς i φορές           |
| {i,j}          | Ο προηγούμενος χαρακτήρας από i μέχρι j φορές       |
| {i,}           | Ο προηγούμενος χαρακτήρας από i μέχρι άπειρες φορές |

Πίνακας 2- 4: Οι μεταχαρακτήρες modifier

Ακολουθούν μερικά παραδείγματα κανονικών εκφράσεων που περιέχουν modifier μεταχαρακτήρες μαζί με την εξήγησή τους και την εκτέλεσή τους:

- **'file[[:digit:]]?'** Η λέξη file ακολουθούμενη από ένα ή κανένα ψηφίο και μετά ο χαρακτήρας κενό
- **'file[[:digit:]]\*'** Η λέξη file ακολουθούμενη από κανένα μέχρι άπειρα ψηφία και μετά ο χαρακτήρας κενό
- **'{2,}'** Ο χαρακτήρας κενό από 2 και πάνω φορές συνεχόμενες
- **'file[[:digit:]]{2,4}'** Η λέξη file ακολουθούμενη από 2 μέχρι 4 ψηφία και μετά ο χαρακτήρας κενό

```
[user@ekdda regexs]$ cat input2.data
color and new test file and test
for the colour of the file1 older
another file22 of the file322 and the
new test file4322 end into
[user@ekdda regexs]$ grep -E 'file[[:digit:]]?' input2.data
color and new test file and test
for the colour of the file1 older
[user@ekdda regexs]$ grep -E 'file[[:digit:]]*' input2.data
color and new test file and test
for the colour of the file1 older
another file22 of the file322 and the
new test file4322 end into
[user@ekdda regexs]$ grep -E '{2,}' input2.data
for the colour of the file1 older
[user@ekdda regexs]$ grep -E 'file[[:digit:]]{2,4}' input2.data
another file22 of the file322 and the
new test file4322 end into
```

## Μεταχαρακτήρες Anchor

Η λειτουργικότητα των anchor μεταχαρακτήρων είναι η περιγραφή της θέσης σε μία γραμμή ή σε μία λέξη στην οποία θέλουμε να βρούμε το κείμενο που αναζητούμε. Για παράδειγμα αν ψάχνουμε να βρούμε τη λέξη *file* αλλά μόνο όταν βρίσκεται στην αρχή της γραμμής τότε γράφουμε `^file`. Ο παρακάτω πίνακας παρουσιάζει όλους τους μεταχαρακτήρες modifier μαζί με τη λειτουργικότητα τους.

| Μεταχαρακτήρας     | Λειτουργικότητα       |
|--------------------|-----------------------|
| <code>^</code>     | Στην αρχή της γραμμής |
| <code>\$</code>    | Στο τέλος της γραμμής |
| <code>\&lt;</code> | Στην αρχή μιας λέξης  |
| <code>\&gt;</code> | Στο τέλος μιας λέξης  |

Πίνακας 2- 5: Οι μεταχαρακτήρες anchor

Μερικά παραδείγματα κανονικών εκφράσεων που περιέχουν anchor μεταχαρακτήρες μαζί με την εξήγησή τους και την εκτέλεσή τους σε πλαίσιο ακολουθεί:

- `'the$'` η λέξη *the* στο τέλος μιας γραμμής
- `'/<e'` ο χαρακτήρας *e* στην αρχή μιας λέξης

```
user@ekdda regexs]$ cat input2.data
color and new test file and test
for the colour of the file1 older
another file22 of the file322 and the
new test file4322 end into
[user@ekdda regexs]$ grep -E 'the$' input2.data
another file22 of the file322 and the
[user@ekdda regexs]$ grep -E '\<e' input2.data
new test file4322 end into
```

### 2.4.2 Ο επεξεργαστής ροών κειμένου sed

Ένα πολύ χρήσιμο εργαλείο που βοηθά στην επεξεργασία κειμένου είναι ο επεξεργαστής ροών κειμένου **sed (stream editor)**. Η επεξεργασία ενός κειμένου με τη χρήση του sed μπορεί να γίνει με έναν από τους παρακάτω τρόπους:

```
sed [παράμετροι] sed_script [αρχείο_για_επεξεργασία]
```

```
cat [αρχείο_για_επεξεργασία] | sed [παράμετροι] sed_script
```

Βλέπουμε δηλαδή ότι μπορούμε να επεξεργαστούμε είτε ένα αρχείο που υπάρχει ήδη είτε να επεξεργαστούμε μία ροή κειμένου που δημιουργείται δυναμικά μέσω μιας διασωλήνωσης (**pipe**). Οι *παράμετροι* της εντολής sed εξειδικεύουν τη λειτουργία της. Το τμήμα *sed\_script* είναι το σημαντικότερο και καθορίζει το είδος της επεξεργασίας που θα πραγματοποιήσει το sed και περιέχεται μέσα σε μονά εισαγωγικά. Η βασική δομή του *sed\_script* είναι η παρακάτω:

```
'addrXcmd_options'
```

Όπως παρατηρούμε ένα `sed_script` αποτελείται από 3 τμήματα τα οποία είναι τα ακόλουθα:

- **X** Είναι ένας χαρακτήρας που καθορίζει το είδος της επεξεργασίας που θα εκτελέσει το `sed` (π.χ. `r` για εκτύπωση)
- **addr** Καθορίζεται το τμήμα της ροής του κειμένου στο οποίο θα εφαρμοστεί η επεξεργασία που ορίζει ο χαρακτήρας `X`
- **cmd\_options** Είναι κάποιες παράμετροι που χρειάζονται για την εφαρμογή της επεξεργασίας που καθορίζει ο χαρακτήρας `X`. Δεν χρειάζονται σε όλες τις επεξεργασίες.



Σκοπός του κειμένου είναι η παρουσίαση των βασικών δυνατοτήτων που παρέχει το `sed`. Για μια αναλυτική παρουσίαση όλων των επιλογών μπορείτε να επισκεφθείτε την ιστοσελίδα αναφοράς του (Pizzini, n.d.).

Για να μελετήσουμε τη λειτουργικότητα του `sed` δημιουργούμε το παρακάτω αρχείο κειμένου (**data.txt**) το οποίο θα χρησιμοποιήσουμε για επεξεργασία.

```
[user@ekdda sed]$ cat data.txt
Name      Count Value Perfecture
Florina   33    28000 Dyt Makedonia
Veroia    43    58000 Kent Makedonia
Larisa    76    83000 Thesalia
Kozani    45    64000 Dyt Makedonia
```

### Εκτύπωση γραμμών.

Η απλούστερη εντολή που διαθέτει ο `sed` είναι η εκτύπωση των γραμμών. Ο χαρακτήρας του `sed_script` που καθορίζει τη λειτουργία αυτή είναι το **p (print)**. Επομένως η εντολή η οποία επιβάλλει στο `sed` να μας εμφανίσει την τρίτη γραμμή του αρχείου μας είναι η παρακάτω:

```
[user@ekdda sed]$ sed '3p' data.txt
Name      Count Value Perfecture
Florina   33    28000 Dyt Makedonia
Veroia    43    58000 Kent Makedonia
Veroia    43    58000 Kent Makedonia
Larisa    76    83000 Thesalia
Kozani    45    64000 Dyt Makedonia
```

Αυτό που παρατηρούμε από την έξοδο της εντολής είναι ότι τυπώθηκαν όλες οι γραμμές του αρχείου, αλλά η τρίτη γραμμή τυπώθηκε δύο φορές. Αυτό οφείλεται στο γεγονός ότι, η προκαθορισμένη λειτουργία του `sed` είναι να τυπώνει όλες τις γραμμές ανεξάρτητα αν τις επεξεργάζεται ή όχι. Αν θέλουμε να αναιρέσουμε αυτή τη λειτουργία μπορούμε να



χρησιμοποιήσουμε την παράμετρο **-n** που καθορίζει στο `sed` να μην εμφανίζει όλες τις εντολές αλλά μόνο για όσες εφαρμόζει την εντολή.

```
[user@ekdda sed]$ sed -n '3p' data.txt
Veroia 43      58000 Kent Makedonia
```

### Επιλογή γραμμών.

Παρατηρούμε ότι στο προηγούμενο παράδειγμα το τμήμα (**addr**) του `sed` script αποτελείται από έναν αριθμό που καθορίζει τον αριθμό γραμμής. Γενικότερα, οι επιλογές που έχουμε για το τμήμα (**addr**) ενός `sed` script είναι οι παρακάτω:

- Ένας αριθμός που συμβολίζει μία γραμμή π.χ. **3p**
- Ένα ζευγάρι αριθμών που συμβολίζει μία ομάδα συνεχόμενων γραμμών π.χ. **2,4p**
- Μία κανονική έκφραση που συμβολίζει τις γραμμές για τις οποίες ισχύει αυτό το ισχύει π.χ. **/Dyt Makedonia\$/p**
- Δύο κανονικές εκφράσεις. Σε αυτή την περίπτωση θα εφαρμοστεί σε όλες τις γραμμές ξεκινώντας από αυτή που ταιριάζει με την πρώτη κανονική έκφραση μέχρι την πρώτη γραμμή που ταιριάζει η δεύτερη κανονική έκφραση (αν δεν υπάρχει μέχρι το τέλος του αρχείου) π.χ. **/43/,/76/**

Τα αποτελέσματα χρήσης των παραπάνω επιλογών στην εντολή `print` του `sed` εμφανίζονται στο παρακάτω πλαίσιο.

```
[user@ekdda sed]$ sed -n '3p' data.txt
Veroia 43      58000 Kent Makedonia
[user@ekdda sed]$ sed -n '2,4p' data.txt
Florina 33     28000 Dyt Makedonia
Veroia 43      58000 Kent Makedonia
Larisa 76      83000 Thesalia
[user@ekdda sed]$ sed -n '/Dyt Makedonia$/p' data.txt
Florina 33     28000 Dyt Makedonia
Kozani 45      64000 Dyt Makedonia
[user@ekdda sed]$ sed -n '/^Veroia/,/Larisa/p' data.txt
Veroia 43      58000 Kent Makedonia
Larisa 76      83000 Thesalia
```

### Άλλες εντολές του sed.

Εκτός από την εντολή εκτύπωσης (`p`) το `sed` διαθέτει και άλλες εντολές που μπορούν να επεξεργαστούν μια ροή κειμένου. Μερικές από αυτές είναι οι παρακάτω:

- **Διαγραφής (d)**. Διαγράφει από τη ροή εξόδου τις γραμμές που περιγράφονται. Για παράδειγμα για να διαγράψουμε τις γραμμές που περιέχουν τη λέξη `Kozani` εφαρμόζουμε την εντολή `sed '/Kozani/d' data.txt`

```
[user@ekdda sed]$ sed '/Kozani/d' data.txt
Name      Count Value Perfecture
Florina 33    28000 Dyt Makedonia
Veroia 43    58000 Kent Makedonia
Larisa 76    83000 Thesalia
```

- **Εισαγωγής κειμένου πριν (i).** Εισάγει το κείμενο πριν από τη γραμμή που περιγράφεται. Αν θέλουμε για παράδειγμα να εισάγουμε το κείμενο “ΑΙΤΙΣΕΙΣ ΑΝΑ ΠΕΡ ΕΝΟΤΙΤΑ” πριν από την πρώτη γραμμή εφαρμόζουμε την εντολή `sed '1iΑΙΤΙΣΕΙΣ ΑΝΑ ΠΕΡ ΕΝΟΤΙΤΑ' data.txt`

```
[user@ekdda sed]$ sed '1iΑΙΤΙΣΕΙΣ ΑΝΑ ΠΕΡ ΕΝΟΤΙΤΑ' data.txt
ΑΙΤΙΣΕΙΣ ΑΝΑ ΠΕΡ ΕΝΟΤΙΤΑ
Name      Count Value Perfecture
Florina 33    28000 Dyt Makedonia
Veroia 43    58000 Kent Makedonia
Larisa 76    83000 Thesalia
Kozani 45    64000 Dyt Makedonia
```

- **Προσθήκης κειμένου μετά (a).** Εισάγει το κείμενο μετά τη γραμμή που περιγράφεται. Αν θέλουμε για παράδειγμα να εισάγουμε το κείμενο “SE EKREMOHTHTA” μετά από κάθε γραμμή που περιέχει το κείμενο “Thesalia” εφαρμόζουμε την εντολή `sed '/Thesalia/aSE EKREMOHTHTA' data.txt`

```
user@ekdda sed]$ sed '/Thesalia/aSE EKREMOHTHTA' data.txt
Name      Count Value Perfecture
Florina 33    28000 Dyt Makedonia
Veroia 43    58000 Kent Makedonia
Larisa 76    83000 Thesalia
SE EKREMOHTHTA
Kozani 45    64000 Dyt Makedonia
```

- **Αντικατάστασης γραμμής (c).** Αντικαθιστά με το κείμενο που δίνουμε γραμμές που ταιριάζουν με το pattern. Για παράδειγμα αν θέλουμε να αντικαταστήσουμε τις γραμμές που περιέχουν κάτω από 40 αιτήσεις με το κείμενο “XAMILES ΑΙΤΙΣΕΙΣ” εφαρμόζουμε την εντολή `sed '/ [0-3][0-9] /c XAMILES ΑΙΤΙΣΕΙΣ' data.txt`

```
[user@ekdda sed]$ sed '/ [0-3][0-9] /c XAMILES AITISEIS' data.txt
Name      Count Value Perfecture
XAMILES AITISEIS
Veroia    43     58000 Kent Makedonia
Larisa    76     83000 Thesalia
Kozani    45     64000 Dyt Makedonia
```

Να σημειωθεί ότι οι εντολές αυτές δεν τροποποιούν το αρχείο που τους δίνεται αλλά απλώς τυπώνουν τα αποτελέσματα τους στην οθόνη. Βέβαια μπορεί να δημιουργηθεί ένα καινούργιο αρχείο με τη χρήση της ανακατεύθυνσης εξόδου όπως στο παρακάτω παράδειγμα.

```
[user@ekdda sed]$ sed '1iAITISEIS ANA PER ENOTITA' data.txt >
data2.txt
```

Στη περίπτωση που θέλουμε οι ενέργειες του sed να εκτελούνται στο ίδιο το αρχείο και όχι απλώς να παρουσιάζονται στην οθόνη μπορούμε να χρησιμοποιήσουμε την επιλογή -i. Με τη χρήση αυτής της επιλογής οι αλλαγές (διαγραφές, εισαγωγές κ.τ.λ.) γίνονται στο ίδιο το αρχείο. Προσοχή, με αυτό τον τρόπο είναι πολύ εύκολο να «καταστρέψουμε» το αρχικό αρχείο όπως στο παράδειγμα που παρουσιάζεται στο παρακάτω πλαίσιο

```
[user@ekdda sed]$ sed -n -i '1p' data2.txt
[user@ekdda sed]$ cat data2.txt
AITISEIS ANA PER ENOTITA
```

### Εντολή αντικατάστασης κειμένου

Μία από τις πιο συχνά χρησιμοποιούμενες ενέργειες του sed είναι η εντολή αντικατάστασης κειμένου που συμβολίζεται με τον χαρακτήρα s. Με την εντολή αντικατάστασης κειμένου περιγράφουμε με μία κανονική έκφραση το κείμενο που θέλουμε να αντικατασταθεί και παρέχουμε και το κείμενο που θα το αντικαταστήσει. Η σύνταξη της εντολής αντικατάστασης είναι κάπως διαφορετική από τις προηγούμενες και παρουσιάζεται παρακάτω:

```
sed 's/κανονική_έκφραση/νέο_κείμενο/g' [αρχείο]
```

Η «κανονική έκφραση» είναι το μοτίβο κειμένου που αναζητούμε στο αρχείο και το «νέο κείμενο» είναι αυτό που θα το αντικαταστήσει. Για παράδειγμα η εντολή που παρουσιάζεται στο παρακάτω πλαίσιο θα αντικαταστήσει όλα τα ποσά με αστερίσκους.

```
[user@ekdda sed]$ sed -E 's/[[:digit:]]{5}/#####/g' data.txt
Name      Count Value Perfecture
Florina 33      ##### Dyt Makedonia
Veroia 43      ##### Kent Makedonia
Larisa 76      ##### Thesalia
Kozani 45      ##### Dyt Makedonia
```

Στην παραπάνω εντολή παρατηρούμε ότι χρησιμοποιούμε την παράμετρο `-E` του `sed` για να τον ενημερώσουμε ότι θέλουμε να χρησιμοποιήσουμε το εκτεταμένο σεντ κανονικών εκφράσεων. Επιπρόσθετα στο τέλος έχουμε προσθέσει την παράμετρο `g` της εντολής αντικατάστασης για να εκτελέσει την αντικατάσταση για όσες φορές χρειαστεί σε μία γραμμή (αν και στο παράδειγμα μας θα γίνει μόνο μία αντικατάσταση). Στην αντικατάσταση μπορούμε να χρησιμοποιήσουμε και το κείμενο που «ταιριάζει» με το μοτίβο με τη χρήση του χαρακτήρα `&`. Στο παρακάτω παράδειγμα προσθέτουμε τη λέξη `euros` μετά το τέλος κάθε ποσού.

```
[user@ekdda sed]$ sed -E 's/[[:digit:]]{5}/& euros/g' data.txt
Name      Count Value Perfecture
Florina 33      28000 euros Dyt Makedonia
Veroia 43      58000 euros Kent Makedonia
Larisa 76      83000 euros Thesalia
Kozani 45      64000 euros Dyt Makedonia
```

### Εντολές σύνδεσης με τον φλοιό.

Υπάρχουν και μερικές εντολές του `sed` που επιτρέπουν τη σύνδεση του με τον φλοιό όπως οι ακόλουθες:

- Εκτέλεση εντολών του `bash` (**e**). Με τη χρήση της εντολής `e` μπορούμε να εκτελούμε μία εντολή του `bash` για κάθε διεύθυνση που περιγράφουμε. Για παράδειγμα μπορούμε να εκτελέσουμε την εντολή `date +%Y` πριν από την πρώτη γραμμή

```
[user@ekdda sed]$ sed '1e echo Year $(date +%Y)' data.txt
Year 2020
Name      Count Value Perfecture
Florina 33      28000 Dyt Makedonia
Veroia 43      58000 Kent Makedonia
Larisa 76      83000 Thesalia
Kozani 45      64000 Dyt Makedonia
```

- Τερματισμός εκτέλεσης (**q**). Με τη χρήση της `q` τερματίζουμε την εκτέλεση του `sed` και μπορούμε να επιστρέψουμε έναν κωδικό εξόδου. Για παράδειγμα μπορούμε να ελέγξουμε αν το αρχείο μας περιέχει ένα ποσό μικρότερο του 30000.

```
[user@ekdda sed]$ sed -n -E '/ [0-2][0-9]{4} /q1' data.txt
[user@ekdda sed]$ echo $?
1
```

- Εκτέλεση πολλών εντολών sed (**παράμετρος e**). Με τη χρήση της παραμέτρου e μπορούμε να εκτελέσουμε παραπάνω από μία εντολές του sed μαζί. Για παράδειγμα μπορούμε να εκτυπώσουμε την πρώτη γραμμή αλλά και να ελέγξουμε αν κάποιο ποσό είναι κάτω από τις 30000.

```
[user@ekdda sed]$ sed -n -E -e '/ [0-2][0-9]{4} /q1' -e '1p'
data.txt
Name      Count Value Perfecture
```

### 2.4.3 Ο επεξεργαστής κειμένου awk

Ένα από τα πολύ χρήσιμα εργαλεία που παρέχει το λειτουργικό σύστημα Linux είναι και το awk. Το awk είναι ένας επεξεργαστής κειμένου που βοηθά στη δημιουργία αναφορών. Ενσωματώνει μια γλώσσα προγραμματισμού με συντακτικό παρόμοιο με τη γλώσσα προγραμματισμού C και παρέχει μεγάλη λειτουργικότητα στην επεξεργασία των αρχείων κειμένου.

#### Εισαγωγή – δομή αρχείων script

Το awk λαμβάνει σαν είσοδο μία-μία τις γραμμές ενός αρχείου κειμένου και σε αυτές εκτελεί κάποιες ενέργειες ανάλογα με τις εντολές που του έχουμε δώσει. Η διαδικασία αυτή επαναλαμβάνεται μέχρι να τελειώσουν οι γραμμές του αρχείου. Η σύνταξη μιας εντολής awk έχει την παρακάτω μορφή:

*awk εντολές αρχείο\_κειμένου*

Για παράδειγμα μια εντολή που θα τύπωνε όλες τις γραμμές του αρχείου `/etc/passwd` είναι η παρακάτω:

```
[user@ekdda sed]$ awk '{print $0}' /etc/passwd
```



Μια αναλυτική παράθεση όλων των λειτουργιών που προσφέρει το awk μπορείτε να βρείτε στο εγχειρίδιο χρήσης του (Robbins, 2020).

Ο τρόπος που λειτουργεί το awk είναι ο ακόλουθος: Αρχικά χωρίζει το αρχείο κειμένου σε **εγγραφές (records)** οι οποίες στην προκαθορισμένη περίπτωση είναι οι γραμμές. Στη συνέχεια κάθε γραμμή τη χωρίζει σε **πεδία (fields)** χρησιμοποιώντας σαν διαχωριστικό τους χαρακτήρες κενό ή tab. Όταν θέλουμε να αναφερθούμε στα πεδία κάθε γραμμής χρησιμοποιούμε τον χαρακτήρα δολάριο (\$) μαζί με έναν αριθμό που καθορίζει τον αύξοντα αριθμό του πεδίου ξεκινώντας από το 1. Στο προηγούμενο παράδειγμα το \$0 είναι

ένα ειδικό πεδίο που περιγράφει όλη τη γραμμή. Αν θέλαμε να τυπώσουμε το τμήμα της γραμμής από την αρχή μέχρι την πρώτη εμφάνιση κενού ή tab, δηλαδή το πρώτο πεδίο, θα δίναμε την παρακάτω εντολή:

```
[user@ekdda sed]$ awk '{print $1}' /etc/passwd
```

Υπάρχουν πολύπλοκες επεξεργασίες όπου το awk χρειάζεται να εκτελέσει παραπάνω από μία εντολές. Σε αυτή την περίπτωση είναι δυνατή η δημιουργία ενός ξεχωριστού αρχείου με εντολές του awk που ονομάζεται **awk script**. Η σύνταξη με την οποία καθορίζουμε στο awk να δεχτεί τις εντολές από ένα αρχείο script είναι η παρακάτω:

```
awk -f awk_script αρχείο_κειμένου
```

Η βασική δομή ενός αρχείου awk script περιέχει τρία τμήματα (**block**). Το πρώτο τμήμα ονομάζεται **BEGIN** και περιέχει εντολές που εκτελούνται μία φορά πριν από την επεξεργασία της πρώτης γραμμής. Το **Βασικό** τμήμα που εκτελείται για κάθε γραμμή του κειμένου και τέλος το τμήμα **END** που εκτελείται μία φορά στο τέλος αφού έχει υποστεί επεξεργασία και η τελευταία γραμμή του αρχείου κειμένου. Τα τμήματα BEGIN και END είναι προαιρετικά και μπορεί να υπάρχουν περισσότερα από ένα βασικά block. Η δομή ενός awk script παρουσιάζεται στο επόμενο σχήμα

```
BEGIN{
    Εκτέλεση στην αρχή της επεξεργασίας του αρχείου
}

{
    Βασικό Block
    Εκτέλεση σε κάθε γραμμή του αρχείου κειμένου
}

{
    ... πιθανώς και άλλα Βασικά Block
}

END{
    Εκτέλεση στο τέλος της επεξεργασίας του αρχείου
}
```

Ένα παράδειγμα awk script παρουσιάζεται στο παρακάτω πλαίσιο.

```
BEGIN{
    print "Starting process...."
}

{
    print "processing: " $0
}

END{
    print "End of processing"
}
```

### Επιλεκτική εκτέλεση εντολών

Μεγάλο βαθμό της λειτουργικότητας του το awk την οφείλει στο γεγονός ότι τα βασικά blocks μπορούν να εκτελούνται επιλεκτικά σε κάποιες εγγραφές με βάση κάποιες συνθήκες. Στη συνέχεια θα παρουσιάσουμε τη λειτουργικότητα του awk χρησιμοποιώντας σαν παράδειγμα το αρχείο **data.txt** που δημιουργήσαμε για τον επεξεργαστή ροών sed (2.4.2). Μερικές από τις συνθήκες που μπορούν να χρησιμοποιηθούν για την επιλεκτική εκτέλεση εντολών στο awk είναι οι παρακάτω:

**Κανονικές εκφράσεις.** Ένα εσωτερικό block μπορεί να εκτελεστεί μόνο σε περίπτωση που η γραμμή του αρχείου κειμένου ικανοποιεί μια κανονική έκφραση. Για παράδειγμα μπορούμε να τυπώσουμε την πόλη και το ποσό μόνο για τις γραμμές που περιέχουν τη λέξη "Kozani".

```
/^Kozani/{
    print $1 " " $3
}
```

**Λογικές εκφράσεις.** Ένα block μπορεί να εκτελεστεί μόνο για τις γραμμές του αρχείου στις οποίες μια λογική έκφραση έχει αποτιμηθεί ως αληθής. Υπάρχουν οι τελεστές σύγκρισης "==" , "!=", "<", "<=", ">", ">=" για αριθμητικές τιμές καθώς και οι τελεστές "~" και "!~" που μπορούν να εφαρμοστούν σε κανονικές εκφράσεις. Για παράδειγμα μπορούμε να τυπώσουμε τα ονόματα των πόλεων και τον αριθμό των αιτήσεων για τις γραμμές στις οποίες ο αριθμός αιτήσεων είναι μικρότερος του 50.

```
$2<50{
    print $1 " " $2
}
```

**Εντολές επιλογής.** Ένας εναλλακτικός τρόπος για να εκτελέσουμε επιλεκτικά κάποιες εντολές είναι με τη βοήθεια εντολών επιλογής. Σε αυτή την περίπτωση έχουμε μια μικρή διαφοροποίηση στη σύνταξη. Το εσωτερικό block θα εκτελεστεί κανονικά αλλά μπορούμε

να επιλέξουμε αν θα εκτελεστεί κάποιο τμήμα του με τη χρήση της κατάλληλης συνθήκης στην εντολή επιλογής. Η εκτέλεση της ίδιας λειτουργικότητας με το προηγούμενο παράδειγμα γίνεται με τον παρακάτω κώδικα και τη χρήση της εντολής επιλογής if.

```
{
  if($2<50)
    print $1 " " $3
}
```

### Μεταβλητές

Η γλώσσα του awk υποστηρίζει μεταβλητές. Υπάρχουν κάποιες ειδικές μεταβλητές που εξειδικεύουν τη λειτουργικότητα του awk ή παρέχουν πληροφορίες σχετικές με το αρχείο κειμένου που επεξεργάζεται. Μερικές ενδεικτικές είναι οι παρακάτω:

**Διαχωριστής πεδίων (Field Separator FS).** Μία ειδική μεταβλητή που υποστηρίζει το awk είναι η μεταβλητή FS. Η τιμή αυτής της μεταβλητής καθορίζει τον χαρακτήρα που χρησιμοποιείται για τον διαχωρισμό της κάθε γραμμής σε πεδία. Για παράδειγμα αν θέλουμε να εκτυπώσουμε το όνομα των λογαριασμών που υπάρχουν στο σύστημά μας θα πρέπει να εμφανίσουμε το πρώτο πεδίο του αρχείου **/etc/passwd** χρησιμοποιώντας όμως σαν διαχωριστή πεδίου τον χαρακτήρα άνω και κάτω τελεία (:) όπως στο παρακάτω πλαίσιο.

```
[user@ekdda awk]$ awk 'BEGIN{FS=":"}{print $1}' /etc/passwd
^H
[user@ekdda awk]$ awk -F: '{print $1}' /etc/passwd
```

Όπως παρατηρούμε στο παραπάνω παράδειγμα μπορούμε να αναθέσουμε τιμή στη μεταβλητή FS και με τη χρήση της παραμέτρου -F. Η τιμή της μεταβλητής FS μπορεί να είναι ακόμα και κανονική έκφραση.

**Διαχωριστής εγγραφών (Record Separator RS).** Όπως έχουμε αναφέρει το awk επεξεργάζεται το αρχείο κειμένου χωρίζοντάς το σε γραμμές και την κάθε γραμμή τη χωρίζει σε πεδία με τη χρήση της μεταβλητής FS. Στην πραγματικότητα το awk χωρίζει το αρχείο κειμένου σε εγγραφές-records με τη βοήθεια της μεταβλητής RS και την κάθε εγγραφή σε πεδία-fields με τη χρήση της μεταβλητής FS. Απλώς η προκαθορισμένη τιμή της μεταβλητής RS είναι ο χαρακτήρας αλλαγής γραμμής οπότε οι προκαθορισμένες εγγραφές είναι οι γραμμές ενός αρχείου. Αλλάζοντας όμως την τιμή της μεταβλητής **RS="end"** κάθε εγγραφή πλέον είναι το κομμάτι του αρχείου κειμένου μέχρι να συναντήσει το κείμενο "end". Στο παρακάτω πλαίσιο παρουσιάζουμε παράδειγμα όπου έχουμε ένα αρχείο **data2.txt** με τα δεδομένα του αρχείου data.txt που χρησιμοποιούμε αλλά κάθε στοιχείο σε διαφορετική γραμμή και με τη λέξη end να διαχωρίζει την κάθε εγγραφή. Η επεξεργασία του αρχείου με το awk γίνεται με αλλαγή της τιμής της μεταβλητής RS.



```
[user@ekdda awk]$ tail -5 data2.txt
Kozani
45
64000
Dyt Makedonia
end
[user@ekdda awk]$ awk 'BEGIN{RS="end"}{print $3}' data2.txt
Value
28000
58000
83000
64000
```

**Αριθμός εγγραφών, πεδίων (Number of Records Fields NR,NF).** Οι μεταβλητές NR και NF δεν εξειδικεύουν τη λειτουργία του awk αλλά παρέχουν πληροφορίες σχετικές με το αρχείο κειμένου που επεξεργάζεται. Συγκεκριμένα, η NF περιέχει την πληροφορία του αριθμού των πεδίων που περιέχει η τρέχουσα εγγραφή. Ενώ η NR περιέχει τον αύξοντα αριθμό της τρέχουσας εγγραφής που επεξεργάζεται εκείνη τη στιγμή το awk. Στο παρακάτω παράδειγμα χρησιμοποιούμε την NR για να τυπώσει μόνο τη 2 και 3 γραμμή του αρχείου εισόδου.

```
[user@ekdda awk]$ awk '(NR>=2)&&(NR<=3){print $0}' data.txt
Florina 33      28000 Dyt Makedonia
Veroia  43      58000 Kent Makedonia
```

**Μεταβλητές που ορίζονται από τον χρήστη.** Εκτός από τις μεταβλητές που παρέχει το ίδιο το awk μεταβλητές μπορεί να δημιουργήσει και ο ίδιος ο χρήστης προκειμένου να αποθηκεύσει τα δεδομένα που θέλει. Το awk είναι εξαιρετικά ευέλικτο στον τύπο μιας μεταβλητής. Γενικά όλες οι μεταβλητές του awk είναι τύπου κειμένου. Παρόλα αυτά, αν το awk διαπιστώσει ότι η μεταβλητή αυτή χρησιμοποιείται σε αριθμητική πράξη και ότι η τιμή της μεταβλητής είναι αριθμός, κάνει αυτόματα τη μετατροπή και υπολογίζει το αποτέλεσμα. Στο παρακάτω παράδειγμα υπολογίζεται ο συνολικός αριθμός των αιτήσεων (πεδίο 2).

```
[user@ekdda awk]$ cat sum.awk
BEGIN{sum=0}
NR>1{sum=sum+$2}
END{print "the sum is:" sum}
[user@ekdda awk]$ awk -f sum.awk data.txt
the sum is:197
```

Ένα ακόμα χρήσιμο χαρακτηριστικό του awk είναι η συνεργασία του με τον φλοιό bash. Συγκεκριμένα, μπορούμε να αναθέσουμε σε μία μεταβλητή awk την τιμή μιας μεταβλητής του φλοιού bash με την παράμετρο -v. Για παράδειγμα μπορούμε να δημιουργήσουμε ένα

script όπου ο χρήστης θα δίνει έναν αριθμό από το πληκτρολόγιο και αυτό θα αναζητά στο αρχείο data.txt και θα επιστρέφει τις πόλεις που έχουν μεγαλύτερο αριθμό αιτήσεων από αυτόν τον αριθμό.

```
1 #!/bin/bash
2
3 read -p 'Ελάχιστος αρ αιτήσεων:' aitiseis
4
5 awk -v ait="$aitiseis" '$2>ait{print $0}' data.tx
```

## 2.5 Μελέτη περιπτώσεων scripts που επιλύουν καθημερινά προβλήματα ενός διαχειριστή

Στη συνέχεια θα αναφέρουμε μερικά παραδείγματα script που έχουν σαν σκοπό να βοηθήσουν την καθημερινότητα ενός διαχειριστή ενός συστήματος Linux Server.



Τα script που αναφέρονται σε καμία περίπτωση δεν μπορούν να θεωρηθούν σαν «ολοκληρωμένα» αλλά ο σκοπός τους είναι να αναδείξουν τη λειτουργικότητα και την πρακτική χρησιμότητα των λειτουργιών που αναφέρθηκαν ως τώρα. Η εξέλιξη τους είναι απαραίτητη και η λειτουργικότητα τους περιορίζεται μόνο από τη φαντασία του προγραμματιστή.

### Έλεγχος πληρότητας κατατμήσεων

Σαν πρώτο παράδειγμα θα δημιουργήσουμε ένα script που θα ελέγχει τις κατατμήσεις που υπάρχουν στο σύστημά μας και θα μας ενημερώνει για το εάν αυτά έχουν ελεύθερο χώρο ή όχι. Η εντολή που μας δείχνει τις κατατμήσεις και τον ελεύθερο χώρο που έχουν είναι η **df -h** (η έξοδος έχει επεξεργαστεί για οικονομία χώρου)

```
Filesystem                Size      Used Avail Use% Mounted on
devtmpfs                  1.9G         0  1.9G   0% /dev
tmpfs                     2.0G         0  2.0G   0% /dev/shm
tmpfs                     2.0G   1.8M  2.0G   1% /run
/dev/mapper/fedora_localhost--live-root 17G   15G  1.5G  91% /
...*
```

Θα δημιουργήσουμε ένα script με το όνομα **mydf.sh** που θα εμφανίζει τις πραγματικές κατατμήσεις που έχουν γεμίσει πάνω από κάποιο ποσοστό. Την τιμή του ποσοστού θα μπορεί να τη δώσει ο χρήστης σαν παράμετρο κατά την κλήση του script. Αν δεν δώσει παράμετρο η προκαθορισμένη τιμή του ποσοστού θα είναι το 70. Το ολοκληρωμένο script είναι το παρακάτω.

```

1 #!/bin/bash
2
3 # usage mydf [limit]
4 # prints partitions that percent usage is over limit
5 # default limit is 70%
6
7 tmpfile="/var/tmp/size$$awk"
8
9 trap "rm -f $tmpfile" EXIT
10
11 if [ $# -ne 1 ]; then
12     limit=70
13 else
14     limit=$1
15 fi
16
17 cat<<'EOF'>$tmpfile
18 /^\/dev/{ if ($5>var) print $0}
19 EOF
20
21 df -h | sed 's/%//' | awk -v var="$limit" -f $tmpfile

```

Η λειτουργικότητα των εντολών του script είναι η ακόλουθη. Στη γραμμή 7 το script δημιουργεί ένα προσωρινό αρχείο το οποίο θα έχει μέσα του ένα awk script. Στην εντολή της γραμμής 9 φροντίζουμε ότι όταν τερματιστεί το script μας, με όποιον τρόπο και αν γίνει αυτό, θα διαγραφεί το προσωρινό αρχείο που έχουμε δημιουργήσει. Οι εντολή if που υπάρχει στις γραμμές 11-15 θέτει τιμή στη μεταβλητή limit. Αν ο χρήστης δώσει παράμετρο η τιμή της limit είναι αυτή η παράμετρος αν δεν δώσει η limit παίρνει την τιμή 70.

Μια ιδιαιτερότητα του script παρουσιάζεται στις γραμμές 17-19. Εδώ χρησιμοποιείται η τεχνική προσωρινών αρχείων (**heredocument**) (Cooper, 2014) για να δημιουργήσουμε ένα προσωρινό awk script. Με αυτόν τον τρόπο ο χρήστης πρέπει να έχει μόνο το αρχείο bash script για να εκτελέσει τη λειτουργία και όχι ένα πρόσθετο αρχείο awk. Η λειτουργικότητα του awk παρουσιάζεται στη γραμμή 18.

```

18 /^\/dev/{ if ($5>var) print $0}

```

Αν παρατηρήσουμε τη γραμμή 18 βλέπουμε ότι η οδηγίες που δίνουμε στον awk είναι να τυπώνει ολόκληρες τις γραμμές (\$0) για τις γραμμές που:

1. Ξεκινάνε με τους χαρακτήρες /dev. Επομένως δεν τυπώνει τις προσωρινές κατατμήσεις που ξεκινάνε με tmp

2. Έχουν στο πέμπτο πεδίο τιμή μεγαλύτερη από την τιμή της μεταβλητής *var*. Το 5 πεδίο που εμφανίζει η *df* είναι το ποσοστό που είναι γεμάτη μια κατάτμηση

Η βασική εντολή του script περιέχεται στη γραμμή 21.

```
21 df -h | sed 's%/%' | awk -v var="$limit" -f $tmpfile
```

Εδώ εκτελείται αρχικά η εντολή *df* και τα αποτελέσματά της παραδίδονται στο *sed*. Το *sed* με τη σειρά του φροντίζει να διαγράψει το % από τα περιεχόμενα του κειμένου, δηλαδή το σύμβολο του ποσοστού από το πέμπτο πεδίο. Και τέλος καλείται το *awk* με οδηγίες που περιέχονται στο προσωρινό αρχείο που δημιουργήσαμε καθώς και με τον ορισμό μιας *awk* μεταβλητής *var* που έχει σαν τιμή την τιμή της μεταβλητής *limit* του script. Τα αποτελέσματα κλήσης του script που δημιουργήσαμε εμφανίζονται στο παρακάτω πλαίσιο.

```
[user@ekdda docsconfigs]$ ./mydf.sh 40
/dev/mapper/fedora_localhost--live-root 17G 15G 1.5G 91 /
/dev/sda1 2.7G 1.9G 678M 74
/home/data
```

### Εργαλείο υπενθύμισης δραστηριοτήτων

Δεύτερο παράδειγμα script είναι ένα εργαλείο υπενθύμισης δραστηριοτήτων. Ο χρήστης θα τρέχει το script και θα του εμφανίζονται οι εργασίες που πρέπει να εκτελέσει με βάση ένα αρχείο εργασιών. Το αρχείο αυτό ονομάζεται **.agenda** και υπάρχει στον home κατάλογο του χρήστη. Τα περιεχόμενα του αρχείου αυτού είναι η ημερομηνία που πρέπει να εκτελέσει την ενέργεια και η περιγραφή της ενέργειας όπως παρακάτω

```
Saturday Create System Backup
4/12/2020 Add new storage
5/5 Annual Report
```

Όπως βλέπουμε από τα περιεχόμενα του αρχείου **.agenda** ο χρήστης μπορεί να βάλει υπενθυμίσεις με μία από τις τρεις μορφές

1. **(Ημέρα)** για μία εργασία που γίνεται κάθε εβδομάδα μια συγκεκριμένη μέρα
2. **(Ημέρα/Μήνας)** για μία εργασία που γίνεται κάθε έτος μια συγκεκριμένη ημερομηνία
3. **(Ημέρα/Μήνας/Έτος)** για μία εργασία που θα γίνει μόνο μία φορά

Το script που παρέχει τη λειτουργικότητα που περιγράψαμε είναι το παρακάτω

```
1 #!/bin/bash
2
3 function missing {
4     echo File $AGENDAFILE is missing!
5     echo You should create agenda file
6     exit 1
7 }
8
9 AGENDAFILE="$HOME/.agenda"
10
11 if [ ! -e $AGENDAFILE ]; then
12     missing
13 fi
14
15 WEEKDAY=$(date +%a)
16 DAY=$(date +%d)
17 MONTH=$(date +%b)
18 YEAR=$(date +%Y)
19
20 echo Today is: $WEEKDAY $DAY/$MONTH/$YEAR
21 echo =====
22
23 format1=$WEEKDAY
24 format2="$DAY/$MONTH"
25 format3="$DAY/$MONTH/$YEAR"
26
27 while read date descr
28 do
29     if [[ $date == $format1 || $date == $format2 || $date ==
30 $format3 ]]; then
31         echo $descr
32     fi
33 done <$AGENDAFILE
```

Αρχικά το script ελέγχει στη γραμμή 11 αν υπάρχει το αρχείο `.agenda` στον home κατάλογο του χρήστη. Σε περίπτωση που δεν υπάρχει εμφανίζει το κατάλληλο μήνυμα και τερματίζει το script με τη βοήθεια της συνάρτησης `missing`.

Στις γραμμές 15-18 το script λαμβάνει με τη χρήση της εντολής `date` την τρέχουσα ημέρα εβδομάδας, ημέρα, μήνα και έτος και δημιουργεί τα αντίστοιχα `format` που περιμένει να συναντήσει στις γραμμές 23-26.

Το κύριο μέρος του script παρουσιάζεται στις γραμμές 27-32 όπου γίνεται ανάγνωση των γραμμών του αρχείου `.agenda` μία γραμμή κάθε φορά και στις συνέχειες γίνεται έλεγχος αν το πεδίο της ημερομηνίας της συγκεκριμένης γραμμής ταιριάζει με ένα από τα `format` που

περιμένει να συναντήσει. Αν αυτό ισχύει τότε τυπώνει το πεδίο της περιγραφής της εργασίας της συγκεκριμένης γραμμής του αρχείου.

## 3 Βασικές λειτουργίες διαχείρισης ενός Linux Server

### 3.1 Διαχείριση χρηστών / ομάδων (groups)

Μία από τις βασικές λειτουργίες ενός διαχειριστή συστήματος LINUX είναι η διαχείριση των χρηστών (**users**) και των ομάδων (**group**). Ένας χρήστης μπορεί να είτε να αντιπροσωπεύει έναν άνθρωπο είτε να δημιουργείται αποκλειστικά για τη λειτουργία μιας εφαρμογής (π.χ. ενός mail Server). Οι ομάδες αναπαριστούν σχήματα οργάνωσης ενός οργανισμού συγκεντρώνοντας χρήστες που έχουν κάποια κοινά χαρακτηριστικά. Οι χρήστες που ανήκουν στην ίδια ομάδα έχουν τα ίδια δικαιώματα στα αρχεία που ανήκουν σε αυτή την ομάδα.

#### 3.1.1 Εισαγωγή, βασικά αρχεία για χρήστες / ομάδες

Κάθε χρήστης αντιστοιχίζεται από το LINUX με έναν μοναδικό αριθμό που ονομάζεται user ID (**UID**) (Hess, 2019). Ομοίως κάθε ομάδα αντιστοιχίζεται με έναν μοναδικό αριθμό που ονομάζεται group ID (**GID**). Οι περισσότερες διανομές του LINUX «δεσμεύουν» μέχρι κάποιον αριθμό για τα UID και τα GID που ανήκουν στο σύστημα (π.χ. η Fedora μέχρι και το 999)<sup>8</sup>.

Ο κάθε χρήστης μπορεί να ανήκει σε παραπάνω από μία ομάδες. Αυτές οι ομάδες χωρίζονται σε:

**Πρωταρχική ομάδα (primary group):** Είναι η ομάδα η οποία ανατίθεται στον χρήστη από τη στιγμή που θα συνδεθεί στο σύστημα. Όλες οι διεργασίες και τα αρχεία που δημιουργεί ανήκουν σε αυτή την ομάδα. Στις περισσότερες διανομές ενεργοποιείται ο μηχανισμός των user private group (αναφέρεται παρακάτω).

**Δευτερεύουσες ομάδες (secondary groups):** Όλες οι υπόλοιπες ομάδες στις οποίες ανήκει ο χρήστης. Ο χρήστης μπορεί ανά πάσα στιγμή να θέσει σαν ενεργή μία από τις δευτερεύουσες ομάδες στις οποίες ανήκει αντί για την πρωταρχική.

Αρκετές διανομές LINUX (π.χ. Fedora, Ubuntu), ακολουθούν το σχήμα των **user private groups**. Σύμφωνα με αυτό το σχήμα για κάθε χρήστη που προστίθεται στο σύστημα δημιουργείται μία καινούργια ομάδα που έχει το όνομά του με μοναδικό μέλος της τον ίδιο τον χρήστη. Η χρησιμοποίηση των user private groups κάνει ασφαλέστερη την ανάθεση προκαθορισμένων δικαιωμάτων σε ένα αρχείο / κατάλογο με τη χρήση της εντολής umask (IAS, 2010).

Οι πληροφορίες για τα UID του χρήστη καθώς και τις ομάδες στις οποίες ανήκει εμφανίζονται με την εντολή id <όνομα χρήστη> όπως στο παρακάτω πλαίσιο

```
[user@ekdda ~]$ id user
uid=1000(user) gid=1000(user) groups=1000(user),10(wheel)
```

<sup>8</sup> Η ρύθμιση υπάρχει στο αρχείο /etc/login.defs και στις μεταβλητές UID\_MIN και GID\_MIN

Μπορούμε να δημιουργήσουμε μια καινούργια ομάδα, αν έχουμε δικαιώματα διαχειριστή, με την εντολή `groupadd` όπως παρακάτω

```
[user@ekdda ~]$ sudo groupadd managers
[sudo] password for user:
```

Εάν θέλουμε να χρησιμοποιήσουμε τα δικαιώματα μιας άλλης ομάδας, από τις δευτερεύουσες ομάδες στις οποίες ήδη ανήκουμε μπορούμε να το κάνουμε με την εντολή `newgrp` όπως στο παρακάτω πλαίσιο

```
[user@ekdda ~]$ id user
uid=1000(user) gid=1000(user)
groups=1000(user),10(wheel),1001(managers)
[user@ekdda ~]$ newgrp - managers
[user@ekdda ~]$ touch testfile
[user@ekdda ~]$ ls -l testfile
-rw-r--r--. 1 user managers 0 Dec  3 15:58 testfile
```

Στο παραπάνω πλαίσιο φαίνεται ότι `user` έχει σαν δευτερεύουσα ομάδα την `managers` την οποία και ενεργοποιεί με τη χρήση της εντολής `newgrp`. Η προαιρετική παύλα πριν από το όνομα της ομάδας προκαλεί την αρχικοποίηση του περιβάλλοντος του λογαριασμού σαν να έχει κάνει μόλις login. Την αλλαγή της ομάδας του `user` την παρατηρούμε από την επόμενη εντολή όπου το αρχείο που δημιουργεί (`testfile`) ανήκει στην ομάδα `managers`.

Μπορούμε να αλλάξουμε και την ομάδα «ιδιοκτήτη» ενός αρχείου / καταλόγου με τη χρήση της εντολής `chown` (Linuxize2, 2019). Η σύνταξη της εντολής `chown` είναι ακόλουθη

**`chown [χρήστης]:[ομάδα] όνομα_αρχείου`**

Μπορούμε να αλλάξουμε χρήστη ιδιοκτήτη μόνο αν έχουμε δικαιώματα `root` ενώ μπορούμε να αλλάξουμε ομάδα ιδιοκτήτη αν είμαστε ο ιδιοκτήτης του αρχείου και ανήκουμε στην καινούργια αυτή ομάδα στην οποία θα ανήκει. Στο παρακάτω πλαίσιο ο χρήστης `user` αλλάζει ομάδα ιδιοκτήτη στο αρχείο `testfile` που μόλις δημιούργησε.

```
[user@ekdda ~]$ ls -l testfile
-rw-r--r--. 1 user managers 0 Dec  3 16:02 testfile
[user@ekdda ~]$ chown :user testfile
[user@ekdda ~]$ ls -l testfile
-rw-r--r--. 1 user user 0 Dec  3 16:02 testfile
[user@ekdda ~]$
```

### Βασικά αρχεία σχετικά με τους χρήστες και τις ομάδες

Υπάρχουν μερικά αρχεία συστήματος όπου το `linux` αποθηκεύει την πληροφορία για τους χρήστες και τις ομάδες (Sheer, 2019). Το αρχείο που κρατείται η πληροφορία σχετικά με



τους χρήστες ενός Linux συστήματος είναι το **/etc/passwd**. Το αρχείο είναι αναγνώσιμο από όλους και περιέχει μία γραμμή για κάθε λογαριασμό που υπάρχει στο σύστημα. Η μορφή της γραμμής αυτής είναι μια σειρά από πεδία χωρισμένα με τον χαρακτήρα άνω και κάτω τελεία (:) παρουσιάζεται στο παρακάτω πλαίσιο.

```
[user@ekdda ~]$ tail -1 /etc/passwd
user:x:1000:1000:user:/home/user:/bin/bash
```

Η πληροφορία που περιέχουν τα πεδία είναι η ακόλουθη:

- Το όνομα του χρήστη (**user**)
- Αν υπάρχει ο χαρακτήρας x στο επόμενο πεδίο τότε υπάρχει συνθηματικό (password) σε κρυπτογραφημένη μορφή για τον συγκεκριμένο λογαριασμό στο αρχείο **/etc/shadow**
- Το **UID** του χρήστη (**1000**)
- Το **GID** της πρωταρχικής ομάδας του χρήστη (**1000**)
- Ένα προαιρετικό πεδίο που περιέχει περισσότερες πληροφορίες για τον χρήστη όπως το πλήρες όνομα του (**user**)
- Ο κατάλογος στον οποίο αποθηκεύονται τα αρχεία του χρήστη (**/home/user**)
- Ο προκαθορισμένος φλοιός που θα χρησιμοποιεί ο χρήστης (**/bin/bash**)

Οι αρχικές διανομές του Linux περιείχαν την πληροφορία για το συνθηματικό ενός χρήστη (την κρυπτογραφημένη μορφή του) στο αρχείο **/etc/passwd**. Νεότερες διανομές έχουν μεταφέρει αυτή την πληροφορία στο αρχείο **/etc/shadow**. Η τεχνική αυτή ονομάζεται (**shadowed passwords**) (FedoraDocs-ShadowPasswords, n.d.) εφαρμόζεται από την Fedora εδώ και καιρό και παρουσιάζει τα παρακάτω πλεονεκτήματα:

- Αυξημένη ασφάλεια επειδή το αρχείο **/etc/shadow** αυτό είναι αναγνώσιμο μόνο από τον root σε αντίθεση με το **/etc/passwd** που είναι αναγνώσιμο από όλους.
- Στο αρχείο **/etc/shadow** μπορεί να προστεθεί η πληροφορία σχετικά με τη γήρανση των συνθηματικών

Το αρχείο **/etc/shadow** είναι αναγνώσιμο μόνο από χρήστες που διαθέτουν δικαιώματα root και η κάθε του γραμμή αντιστοιχεί σε έναν χρήστη. Για κάθε χρήστη περιέχονται πολλές πληροφορίες χωρισμένες σε πεδία με τη χρήση του χαρακτήρα άνω και κάτω τελεία (:) όπως παρακάτω

```
[user@ekdda ~]$ sudo tail -1 /etc/shadow
[sudo] password for user:
user:$6$fcmet5JgVhoB815V$vSMJ9x9YYUitGxvFauvNsVUUrioz7FETMAewwUpV
4kqQkKE4sp36dZ6QaSNbztMvgWo.Rco74ar2zID704bg80:18589:0:99999:7:::
```

Οι πληροφορίες που περιέχουν τα πεδία είναι:

- Το όνομα του χρήστη (**user**)

- Το συνθηματικό του χρήστη σε κρυπτογραφημένη μορφή. Αν ξεκινάει με τον χαρακτήρα ! τότε σημαίνει ότι το συνθηματικό αυτού του λογαριασμού είναι κλειδωμένο
- Τα επόμενα πεδία περιλαμβάνουν πληροφορίες που έχουν να κάνουν με την πολιτική γήρανσης του συγκεκριμένου συνθηματικού και θα παρουσιαστούν στη συνέχεια

Οι πληροφορίες σχετικά με τις ομάδες που υπάρχουν και τους χρήστες που ανήκουν σε αυτές υπάρχουν στο αρχείο **/etc/group**. Όπως και στα προηγούμενα αρχεία και αυτό περιέχει μια γραμμή για κάθε ομάδα με τις πληροφορίες κάθε ομάδας να χωρίζονται με τον χαρακτήρα άνω και κάτω τελεία (:).

Οι πληροφορίες που περιέχει το **/etc/group** είναι οι παρακάτω:

```
[user@ekdda ~]$ tail -2 /etc/group
user:x:1000:
managers:x:1001:user
```

- Το όνομα της ομάδας (π.χ. **manager**)
- Αν υπάρχει ο χαρακτήρας x στο επόμενο πεδίο τότε υπάρχει συνθηματικό για την ομάδα στο αρχείο **/etc/gshadow**<sup>9</sup>
- Το GID της ομάδας (π.χ. **1001**)
- Το UID των χρηστών που ανήκουν στη συγκεκριμένη ομάδα (π.χ. **user**). Να σημειωθεί ότι η πρωταρχική ομάδα στην οποία ανήκει ένας χρήστης βρίσκεται στο αρχείο **/etc/passwd** και όχι στο **/etc/group**

### 3.1.2 Δημιουργία, διαμόρφωση διαγραφή λογαριασμού

Μπορούμε να δημιουργήσουμε έναν καινούργιο λογαριασμό χρήστη με τη χρήση της εντολής **useradd** μόνο αν έχουμε δικαιώματα διαχειριστή. Η σύνταξη της εντολής **useradd** είναι η παρακάτω

**useradd [παράμετροι] όνομα\_χρήστη**

Στο παρακάτω πλαίσιο δημιουργούμε ένα χρήστη με το όνομα **student** με τις προκαθορισμένες επιλογές. Διαπιστώνουμε ότι δημιουργήθηκε ελέγχοντας τα περιεχόμενα του αρχείου **/etc/passwd** και εκτελώντας την εντολή **id**

```
[user@ekdda ~]$ sudo useradd student
[sudo] password for user:
[user@ekdda ~]$ tail -1 /etc/passwd
student:x:1001:1002::/home/student:/bin/bash
[user@ekdda ~]$ id student
uid=1001(student) gid=1002(student) groups=1002(student)
```

<sup>9</sup> Ο μηχανισμός των συνθηματικών για τις ομάδες δεν χρησιμοποιείται στις περισσότερες διανομές

Η δημιουργία ενός χρήστη με τις προκαθορισμένες επιλογές έχει σαν αποτέλεσμα τη δημιουργία ενός λογαριασμού με «κλειδωμένο» συνθηματικό. Για να μπορεί να ενεργοποιηθεί ο λογαριασμός αυτός χρειάζεται η χρήση της εντολής **passwd** από τον **root** για την αρχική ανάθεση ενός συνθηματικού όπως φαίνεται παρακάτω (στη συνέχεια ο χρήστης μπορεί να αλλάξει αυτό το συνθηματικό σε ότι επιθυμεί).

```
[user@ekdda ~]$ sudo passwd student
Changing password for user student.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
```

Όπως είδαμε και προηγουμένως η εντολή **useradd** που χρησιμοποιείται για τη δημιουργία ενός καινούργιου χρήστη μπορεί να πάρει παραμέτρους για τη διαφοροποίηση της λειτουργίας της. Οι βασικές παράμετροι της εντολής **useradd** παρουσιάζονται στον Πίνακα 3-1.

| Παράμετρος             | Λειτουργικότητα  |
|------------------------|--|
| <b>-c</b> 'σχόλιο'     | Είναι ένα κείμενο που παίρνει τη θέση του πραγματικού ονόματος του χρήστη.   |
| <b>-d</b> κατάλογος    | Καθορίζει το home κατάλογο του χρήστη αντί για το προκαθορισμένο /home/username/.  |
| <b>-e</b> ημερομηνία   | Ημερομηνία στην οποία θα απενεργοποιηθεί ο λογαριασμός. Είναι της μορφής YYYY-MM-DD.   |
| <b>-f</b> ημέρες       | Αριθμός των ημερών στις οποίες ο λογαριασμός απενεργοποιείται από την στιγμή που θα "λήξει» το συνθηματικό. Ειδικές τιμές (0 άμεσα -1 ποτέ). |
| <b>-g</b> όνομα_ομάδας | Η πρωταρχική ομάδα στην οποία θα ανήκει ο χρήστης. Αυτή η ομάδα πρέπει να προϋπάρχει.  |
| <b>-G</b> λίστα_ομάδων | Λίστα με τα υπόλοιπες ομάδες, όχι πρωταρχική, χωρισμένες με κόμμα στις οποίες θα συμμετέχει ο χρήστης. Οι ομάδες αυτές πρέπει να προϋπάρχουν |
| <b>-m</b>              | Δημιουργεί το home κατάλογο εάν αυτό δεν υπάρχει   |
| <b>-M</b>              | ΔΕΝ δημιουργεί το home κατάλογο.   |
| <b>-N</b>              | Δεν δημιουργεί ένα user private group για τον χρήστη   |
| <b>-p</b> συνθηματικό  | Το <i>συνθηματικό</i> κρυπτογραφημένο με την crypt.  |
| <b>-r</b>              | Δημιουργεί ένα λογαριασμό συστήματος (UID μικρότερο από 1000 και χωρίς home κατάλογο).   |
| <b>-s</b>              | Ο φλοιός του χρήστη προκαθορισμένος το /bin/bash.  |
| <b>-u</b> UID          | Το UID του χρήστη το οποίο πρέπει να είναι μοναδικό και μεγαλύτερο από 999.  |

Πίνακας 3-1: Παράμετροι της εντολής **useradd**

Στο παρακάτω παράδειγμα δημιουργούμε έναν καινούργιο χρήστη με το όνομα "ngeorgiou" με περιγραφή "Nikolaos Georgiou" και έχει σαν home κατάλογο τον "/home/contract"

```
[root@ekdda ~]# useradd -c "Nikolaos Georgiou" -d /home/contract
ngeorgiou
```

Μπορούμε επίσης να δημιουργήσουμε έναν χρήστη με το όνομα “*etasiou*” που να έχει σαν πρωταρχική ομάδα την ομάδα *managers* και να ανήκει στην ομάδα “*web*” και ο λογαριασμός του να λήγει την ημερομηνία “*31 12 2022*”.

```
[root@ekdda ~]# groupadd managers
[root@ekdda ~]# groupadd web
[root@ekdda ~]# useradd -g managers -G web -e 2022-12-31 etasiou
[root@ekdda ~]# id etasiou
uid=1002(etasiou) gid=1002(managers) groups=1002(managers),
1003(web)
```

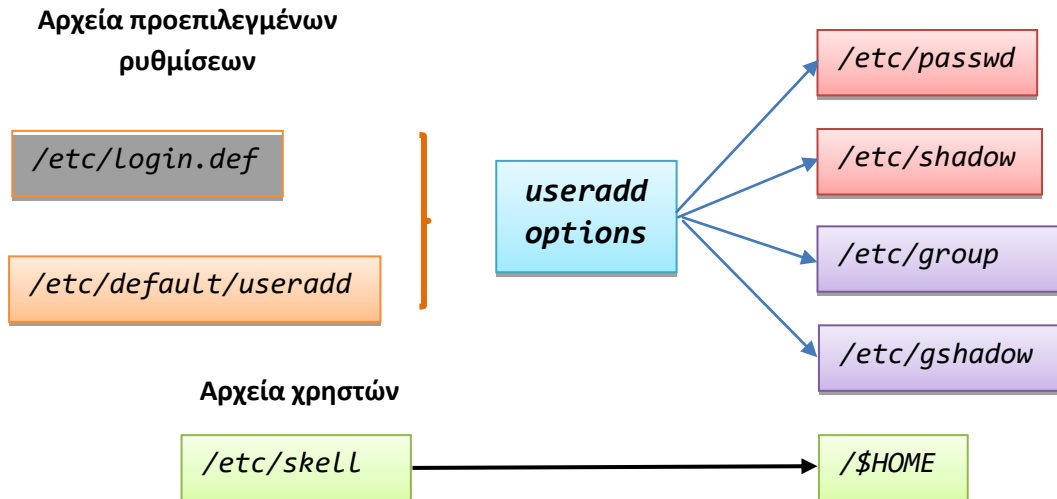
Όπως παρατηρούμε στο παραπάνω πλαίσιο η ομάδα (*managers*) στην οποία θα ανήκει ο χρήστης θα πρέπει να προϋπάρχει.

### Ενέργειες κατά τη δημιουργία χρήστη

Όταν δημιουργούμε τον χρήστη *student* με τη χρήση της εντολής *useradd* χωρίς να χρησιμοποιήσουμε κάποιες από τις επιλογές της, οι ενέργειες που θα πραγματοποιήσει το Linux είναι οι παρακάτω (RedHat2, 2017):

- Θα δημιουργήσει μια γραμμή για τον *student* στο αρχείο */etc/passwd*
- Θα δημιουργήσει μια γραμμή για το συνθηματικό του *student* στο αρχείο */etc/shadow*
- Θα δημιουργήσει μια γραμμή για την ομάδα *student* (user private group) στο αρχείο */etc/group*
- Θα δημιουργήσει μια γραμμή για το συνθηματικό της ομάδας *student* στο */etc/gshadow*
- Θα δημιουργήσει ένα καινούργιο κατάλογο */home/student*
- Θα μεταφέρει τα αρχεία από το */etc/skel* στο */home/student*

Όπως παρατηρούμε κατά τη δημιουργία ενός χρήστη πραγματοποιούνται κάποιες προκαθορισμένες επιλογές από το σύστημα, π.χ. η δημιουργία user private group ή ο προκαθορισμένος φλοιός. Οι προκαθορισμένες επιλογές αυτές μπορούν να ρυθμιστούν και βρίσκονται στα αρχεία */etc/login.defs* και */etc/default/useradd*. Αν θέλουμε να τροποποιήσουμε τις προκαθορισμένες επιλογές μόνο για έναν χρήστη χρησιμοποιούμε τις επιλογές της εντολής *useradd*. Η λειτουργία της *useradd* παρουσιάζεται στην Εικόνα 3- 1.



Εικόνα 3- 1: Λειτουργία useradd

### Τροποποίηση χαρακτηριστικών χρήστη

Από τη στιγμή που θα έχει δημιουργηθεί ένας χρήστης μπορούν να τροποποιηθούν οι ιδιότητες του με τη χρήση της εντολής `usermod`. Οι βασικές παράμετροι της εντολής `usermod` παρουσιάζονται στον Πίνακας 3- 2:

| Παράμετρος                   | Λειτουργικότητα   |
|------------------------------|---|
| <code>-c</code> 'σχόλιο'     | Ίδια με την <code>useradd</code>  |
| <code>-d</code> κατάλογος    | Ίδια με την <code>useradd</code>  |
| <code>-e</code> ημερομηνία   | Ίδια με την <code>useradd</code>  |
| <code>-g</code> όνομα_ομάδας | Ίδια με την <code>useradd</code>  |
| <code>-G</code> λίστα_ομάδων | Ίδια με την <code>useradd</code>  |
| <code>-a</code>              | Προσθέτει λογαριασμό σε δευτερεύουσα ομάδα χωρίς να τον διαγράψει από προηγούμενες (μαζί με την <code>-G</code> ) |
| <code>-l</code>              | Αλλάζει το όνομα χρήστη   |
| <code>-L</code>              | Κλειδώνει το συνηματικό έτσι ώστε ο χρήστης δεν μπορεί να εισέλθει με <code>login</code>                          |
| <code>-m</code>              | Μετακινεί τα περιεχόμενα από το <code>home</code> σε ένα καινούργιο κατάλογο                                      |
| <code>-p</code>              | Καινούργιο συνηματικό   |
| <code>-s</code>              | Ίδια με την <code>useradd</code>  |
| <code>-u</code> UID          | Ίδια με την <code>useradd</code>  |
| <code>-U</code>              | Ξεκλειδώνει έναν λογαριασμό που είχε προηγούμενο κλειδωθεί με την παράμετρο <code>-L</code>                       |

Πίνακας 3- 2: Παράμετροι της εντολής usermod

Μία εργασία στην οποία θα πρέπει να είμαστε προσεκτικοί είναι όταν προσθέτουμε έναν χρήστη σε μία δευτερεύουσα ομάδα ενώ είναι ήδη μέλος σε μία άλλη (δευτερεύουσα) ομάδα. Αν χρησιμοποιήσουμε την παράμετρο `-G` και όνομα της καινούργιας τότε θα τον θέσουμε στην καινούργια και θα τον αφαιρέσουμε από την παλιά. Οπότε ή θα χρησιμοποιήσουμε την παράμετρο `-G` αναφέρουμε όλες τις δευτερεύουσες ομάδες ή θα χρησιμοποιήσουμε την παράμετρο `-a`.

```
[root@ekdda ~]# id ngeorgiou
uid=1002(ngeorgiou) gid=1003(ngeorgiou) groups=1003(ngeorgiou)
[root@ekdda ~]# usermod -G managers ngeorgiou
[root@ekdda ~]# id ngeorgiou
uid=1002(ngeorgiou) gid=1003(ngeorgiou) groups=1003(ngeorgiou),
1004(managers)
[root@ekdda ~]# groupadd web
[root@ekdda ~]# usermod -aG web ngeorgiou
[root@ekdda ~]# id ngeorgiou
uid=1002(ngeorgiou) gid=1003(ngeorgiou)
groups=1003(ngeorgiou),1004(managers),1005(web)
```

Μία από τις ενέργειες διαχείρισης χρηστών που θα χρειαστεί να εκτελέσουμε συχνά σαν διαχειριστές ενός συστήματος είναι εμποδίσουμε έναν χρήστη από το να συνδεθεί στο σύστημα. Μία λύση που έχουμε είναι να κλειδώσουμε το συνθηματικό του. Αυτό μπορεί να γίνει με τη χρήση της παραμέτρου `-L` της `usermod`. Για παράδειγμα μπορούμε να κλειδώσουμε το συνθηματικό του χρήστη `ngeorgiou` και να το επιβεβαιώσουμε από τον χαρακτήρα `!` που υπάρχει στην αρχή του κρυπτογραφημένου συνθηματικού του στο αρχείο `/etc/shadow`.

```
[root@ekdda ~]# usermod -L ngeorgiou
[root@ekdda ~]# cat /etc/shadow | grep ngeorgiou
ngeorgiou:!$6$BwnQXvC7yUoCwohq$ZFPuH8mu1lQ8zkzYfzX8vgWIp5dhsHgoEO1
lRUXvOT05PVeeNQj2WhwnOL/XUQphbZpuKsz2W0qD8PmShh86MK1:18601:0:9999
9:7:::
```

Με το κλείδωμα του συνθηματικού ο χρήστης δεν μπορεί να κάνει login στο σύστημα μας με συνθηματικό μπορεί όμως να κάνει login αν έχει ενεργοποιήσει ένα κλειδί ssh hash. Μία καλύτερη λύση είναι να απενεργοποιήσουμε όχι το συνθηματικό αλλά τον ίδιο τον λογαριασμό του. Αυτό μπορούμε να το κάνουμε τοποθετώντας μια παλαιότερη ημερομηνία σαν ημερομηνία λήξης του συναγερμού.

```
[root@ekdda ~]# usermod -U ngeorgiou
[root@ekdda ~]# usermod -e 1970-1-1 ngeorgiou
[root@ekdda ~]# cat /etc/shadow | grep ngeorgiou
ngeorgiou:$6$BwnQXvC7yUoCwohq$ZFPuH8mu1lQ8zkzYfzX8vgWIp5dhsHgoEO1
RUXvOT05PVeeNQj2WhwnOL/XUQphbZpuKsz2W0qD8PmShh86MK1:18601:0:99999
:7::0:
```

Μία τελευταία περίπτωση αλλαγής χαρακτηριστικών χρήστη που θα παρουσιάσουμε είναι η περίπτωση που ο χρήστης θελήσει να αλλάξει username. Σε αυτή την περίπτωση πρέπει να κάνουμε μια σειρά από ενέργειες: α) να αλλάξουμε το username β) να αλλάξουμε το user private group γ) να αλλάξουμε το όνομα του home καταλόγου. Πριν από αυτές τις ενέργειες θα πρέπει να βεβαιωθούμε ότι ο χρήστης δεν είναι συνδεδεμένος στο σύστημα

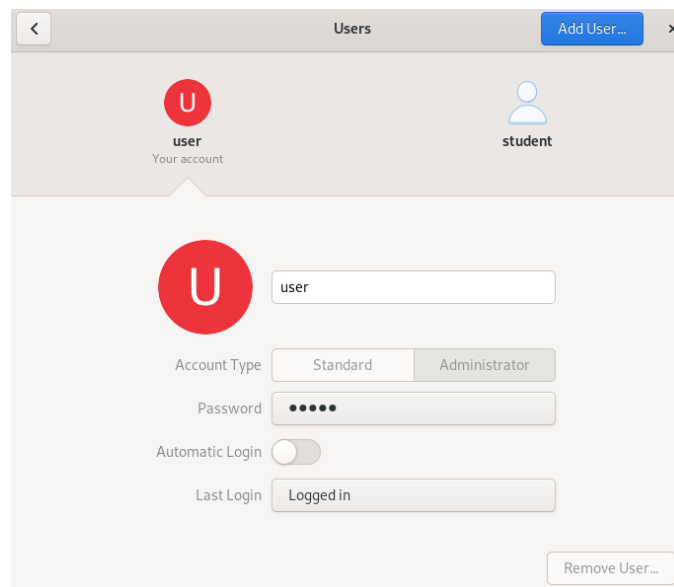
μας. Οι ενέργειες αυτές παρουσιάζονται στο παρακάτω πλαίσιο όπου μεταβάλουμε το όνομα ενός χρήστη από “stergiou” σε “nstergiou”

```
[root@ekdda ~]# id stergiou
uid=1003(stergiou) gid=1006(stergiou) groups=1006(stergiou)
[root@ekdda ~]# usermod -l nstergiou stergiou
[root@ekdda ~]# groupmod -n nstergiou stergiou
[root@ekdda ~]# usermod -d /home/nstergiou -m nstergiou
[root@ekdda home]# id nstergiou uid=1003(nstergiou)
gid=1006(nstergiou) groups=1006(nstergiou)
```

### 3.1.3 Χρήση γραφικών εργαλείων για τη διαχείριση χρηστών

Η δημιουργία και η διαχείριση λογαριασμών χρηστών μπορεί να γίνει και με τη χρήση του γραφικού εργαλείου **Users** χωρίς όμως να παρέχονται όλες οι δυνατότητες των εντολών `useradd` και `usermod`. Μπορούμε να ξεκινήσουμε το εργαλείο **Users** μέσα από τα **Settings** του συστήματος ή πιέζοντας το πλήκτρο `super` στη συνέχεια γράφοντας `users` στο πλαίσιο αναζήτησης και πατώντας `Enter`.

Η μορφή του εργαλείου **Users** παρουσιάζεται στην Εικόνα 3- 2. Αρχικά θα πρέπει να πατήσουμε το εικονίδιο με το λουκέτο και να δώσουμε το συνθηματικό για να αποδείξουμε ότι έχουμε δικαιώματα `root` στο σύστημα. Το εργαλείο `users` παρουσιάζει στο πάνω μέρος τους λογαριασμούς του συστήματος και στο κάτω μέρος λεπτομέρειες για έναν από τους λογαριασμούς που έχουμε επιλέξει όπως το όνομα του, τον τύπο του και το συνθηματικό του.



Εικόνα 3- 2: Διαχείριση χρηστών με γραφικό τρόπο

Μπορούμε να προσθέσουμε έναν καινούργιο λογαριασμό αν πατήσουμε το εικονίδιο “Add User...” που υπάρχει και το οποίο θα έχει σαν αποτέλεσμα να μας εμφανίσει το παράθυρο προσθήκης νέου χρήστη όπως παρουσιάζεται στην Εικόνα 3- 3. Εδώ μπορούμε να εισάγουμε τα στοιχεία του καινούργιου χρήστη όπως και το αν θα έχει δικαιώματα `root`

καθώς και να καθορίσουμε εμείς συνθηματικό ή να επιτρέψουμε στον χρήστη να το καθορίσει αυτός την επόμενη φορά που θα συνδεθεί στο σύστημα.

Εικόνα 3- 3: Προσθήκη χρήστη με γραφικό τρόπο

### 3.1.4 Διαχείριση συνθηματικών

Οι σύγχρονες διανομές Linux που χρησιμοποιούν τον μηχανισμό των shadowed passwords δίνουν τη δυνατότητα να θέσουμε πολιτικές στη χρήση των συνθηματικών από τους χρήστες. Όπως είδαμε οι πληροφορίες των συνθηματικών υπάρχουν μέσα στο αρχείο **/etc/shadow**. Η δυνατότητα ορισμού πολιτικής συνθηματικών επιτρέπεται μόνο αν διαθέτουμε δικαιώματα διαχειριστή και δίνεται με την εντολή **chage** όπως παρακάτω:

***chage [παράμετροι] όνομα\_χρήστη***

Οι προαιρετικές παράμετροι που μπορούμε να δώσουμε στην εντολή **chage** αναφέρονται στον Πίνακα 3- 3.

| Παράμετρος           | Λειτουργικότητα  |
|----------------------|--|
| <b>-d</b> ημέρες     | Καθορίζει τον αριθμό των ημερών από τις 1/1/1970 μέχρι που άλλαξε το password  |
| <b>-E</b> ημερομηνία | Καθορίζει την ημέρα στην οποία ο λογαριασμός θα κλειδωθεί  |
| <b>-I</b> ημέρες     | Καθορίζει τις ημέρες που επιτρέπει να χρησιμοποιείται ο λογαριασμός μετά τη λήξη του συνθηματικού μέχρι να κλειδωθεί (0 δεν κλειδώνεται) |
| <b>-l</b>            | Εμφανίζει τις ρυθμίσεις τις πολιτικής των συνθηματικό  |
| <b>-m</b> ημέρες     | Καθορίζει τον ελάχιστο αριθμό ημερών μετά από τις οποίες μπορεί να αλλάξει το συνθηματικό  |
| <b>-M</b> ημέρες     | Καθορίζει τον μέγιστο αριθμό ημερών στις οποίες το συνθηματικό είναι ενεργό. Μετά πρέπει να το αλλάξει ο χρήστης                         |
| <b>-W</b> ημέρες     | Καθορίζει τις ημέρες πριν τη λήξη του συνθηματικό όπου   |



|  |   |
|--|---|
|  | το σύστημα θα εμφανίζει προειδοποίηση αλλαγής στον χρήστη |
|--|---|

Πίνακας 3-3: Παράμετροι της εντολής chage

Η chage μπορεί να χρησιμοποιηθεί και χωρίς καμία παράμετρο οπότε λειτουργεί διαδραστικά εμφανίζοντας και επιτρέποντας τη ρύθμιση (με Enter παραμένει η ίδια τιμή) των ρυθμίσεων της πολιτικής των συνθηματικών. Η χρήση της chage διαδραστικά παρουσιάζεται στο παραπάνω πλαίσιο

```
[user@ekdda ~]$ sudo chage user
Changing the aging information for user
Enter the new value, or press ENTER for the default

Minimum Password Age [0]:
Maximum Password Age [99999]:
Last Password Change (YYYY-MM-DD) [2020-11-23]:
Password Expiration Warning [7]:
Password Inactive [-1]:
Account Expiration Date (YYYY-MM-DD) [-1]:
```

### 3.1.5 Δικαιώματα πρόσβασης μέσω group

Οι ομάδες είναι ένα διαχειριστικό εργαλείο με τη χρήση του οποίου μπορούμε να αποδώσουμε δικαιώματα σε ένα υποσύνολο χρηστών με μία κίνηση. Για παράδειγμα ας δούμε πώς μπορεί ο διαχειριστής να δώσει δικαίωμα πρόσβασης και εγγραφής σε έναν κατάλογο σε μία συγκεκριμένη ομάδα χρηστών.

```
[root@ekdda ~]# mkdir /financial
[root@ekdda ~]# chown :managers /financial
[root@ekdda ~]# chmod 770 /financial
[root@ekdda ~]# ls -ld /financial
drwxrwx---. 2 root managers 4096 Jan  2 10:49 /financial
```

Παρατηρούμε ότι ο διαχειριστής (root) δημιούργησε έναν κατάλογο (/financial) στον οποίο του έδωσε σαν ομάδα ιδιοκτήτη την ομάδα **managers**. Επίσης μετέβαλε τα δικαιώματα σε αυτόν τον κατάλογο έτσι ώστε να έχουν πρόσβαση και εγγραφή μόνο αυτοί που ανήκουν στην ομάδα αυτή.

Πράγματι μπορούμε να παρατηρήσουμε ότι ο χρήστης ngeorgiou που ανήκει στην ομάδα managers μπορεί να έχει πρόσβαση στο συγκεκριμένο κατάλογο καθώς και να δημιουργήσει αρχεία μέσα σε αυτόν. (Προσοχή θα πρέπει να αλλάξει την πρωταρχική του ομάδα αν θέλει οι άλλοι χρήστες της ομάδας να έχουν πρόσβαση στα αρχεία που δημιουργεί).

```
[ngeorgiou@ekdda ~]$ id
uid=1002(ngeorgiou) gid=1003(ngeorgiou)
groups=1003(ngeorgiou),1004(managers),1005(web)
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[ngeorgiou@ekdda ~]$ cd /financial
[ngeorgiou@ekdda financial]$ newgrp managers
[ngeorgiou@ekdda financial]$ touch mybill
[ngeorgiou@ekdda financial]$ ls -l mybill
-rw-r--r--. 1 ngeorgiou managers 0 Jan  2 10:57 mybill
```

Ενώ παρατηρούμε ότι ο χρήστης user που δεν ανήκει στην ομάδα managers δεν μπορεί να έχει πρόσβαση στον κατάλογο /financial.

```
[user@ekdda ~]$ id
uid=1000(user) gid=1000(user) groups=1000(user),10(wheel)
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[user@ekdda ~]$ cd /financial
bash: cd: /financial: Permission denied
```

Μία ομάδα που δημιουργεί αυτόματα η διανομή Fedora κατά την εγκατάσταση της είναι η ομάδα (**wheel**). Αυτή η ομάδα έχει μια ξεχωριστή λειτουργικότητα. Συγκεκριμένα, τα μέλη αυτής της ομάδα έχουν δικαιώματα διαχειριστή του συστήματος με τη χρήση της εντολής **sudo** (FedoraDocs-Sudo, n.d.). Η sudo είναι μια εντολή με την οποία μπορεί ένας κανονικός χρήστης να εκτελέσει μία ή περισσότερες εντολές με δικαιώματα διαχειριστή. Για να το κάνει αυτό πρέπει να γνωρίζει μόνο το δικό του συνθηματικό και όχι το συνθηματικό του διαχειριστή. Είναι μια καλή πρακτική η χρήση της sudo αντί του λογαριασμού διαχειριστή για τις καθημερινές εργασίες ενός διαχειριστή συστήματος. Γι' αυτό τον λόγο πολλές διανομές δεν ζητούν τον ορισμό του συνθηματικού του root κατά τη διάρκεια εγκατάστασης τους.

Για παράδειγμα ο χρήστης user που είναι μέλος της ομάδας wheel μπορεί να εκτελέσει την εντολή sudo ώστε να δει τα περιεχόμενα του αρχείου /etc/shadow.

```
[user@ekdda ~]$ id
uid=1000(user) gid=1000(user) groups=1000(user),10(wheel)
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[user@ekdda ~]$ sudo tail -1 /etc/shadow
[sudo] password for user:
npetrou:$6$9s9vxNeGJpmo.LB1$52IgQYkNAYOiwFB6rRIIJf4xmjnJ8I78UiONa
Tg/XeUWa.cmI8z5jwUj7p5KLWoR6.A4qteZHKyWNj0vxrw/X/:18629:0:99999:7
:::
```

Αντίθετα ο χρήστης ngeorgiou δεν μπορεί να εκτελέσει την εντολή sudo λόγο του ότι δεν ανήκει στην ομάδα wheel.

```
[ngeorgiou@ekdda ~]$ id
uid=1002(ngeorgiou) gid=1003(ngeorgiou)
groups=1003(ngeorgiou),1004(managers),1005(web)
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[ngeorgiou@ekdda ~]$ sudo tail -1 /etc/shadow
[sudo] password for ngeorgiou:
ngeorgiou is not in the sudoers file. This incident will be
reported.
```

Μπορούμε πολύ εύκολα να δώσουμε δικαιώματα διαχειριστή και στον χρήστη ngeorgiou με το να τον προσθέσουμε στην ομάδα wheel.

```
[root@ekdda ~]# usermod -aG wheel ngeorgiou
```

### 3.2 Διαχείριση πακέτων λογισμικών (packages)

Ένα από τα καθήκοντα του διαχειριστή ενός Linux Server είναι η εγκατάσταση, διαγραφή, ενημέρωση και παραμετροποίηση των λογισμικών του συστήματος. Στις διάφορες διανομές του Linux τα λογισμικά διανέμονται με τη μορφή των **πακέτων λογισμικού**<sup>10</sup> (**software packages**). Ένα πακέτο λογισμικού είναι ένα συμπιεσμένο αρχείο που μέσα του περιέχει όλα τα αρχεία που χρειάζονται για να εγκατασταθεί ένα λογισμικό στο Linux. Συγκεκριμένα, ένα πακέτο λογισμικού μπορεί να περιέχει τον πηγαίο κώδικα, τα εκτελέσιμα αρχεία, τα αρχεία σεναρίων scripts και τα μεταδοδεμένα που απαιτούνται για την εγκατάσταση του λογισμικού. Μεταδεδομένα ενός λογισμικού είναι πληροφορίες όπως: αν το λογισμικό πρέπει να εγκατασταθεί σε κάποιον συγκεκριμένο κατάλογο, ή αν χρειάζεται βιβλιοθήκες για την εκτέλεση του, η έκδοση του κτλ. Ένα σημαντικό μεταδεδομένο είναι η πληροφορία ότι το συγκεκριμένο λογισμικό για την εγκατάσταση του χρειάζεται την ύπαρξη ενός άλλου λογισμικού ήδη στο σύστημα. Το μεταδεδομένο αυτό περιγράφεται με τον όρο **εξάρτηση λογισμικών (software dependency)**. Στο παρακάτω πλαίσιο παρουσιάζεται η εντολή που παρουσιάζει τα στοιχεία που περιέχει το πακέτο λογισμικού *htop*.

```
[user@ekdda ~]$ dnf repoquery -l htop
```

Τα πακέτα λογισμικού είναι οργανωμένα με τη μορφή των **αποθητηρίων λογισμικού (software repositories, repos)**. Ειδικότερα, τα αποθετήρια λογισμικού είναι τεράστιες λίστες από πακέτα λογισμικών που είναι διαθέσιμα για κατέβαση και εγκατάσταση. Επίσης, οι περισσότερες διανομές του Linux, ανάμεσα τους και η Fedora, διαθέτουν

<sup>10</sup> Στην συνέχεια της ενότητας οι όροι λογισμικό και πακέτο λογισμικού θα χρησιμοποιούνται με την ίδια έννοια

επίσημα αποθετήρια λογισμικού που περιέχουν πακέτα λογισμικού που έχουν δοκιμαστεί και εκτελούνται χωρίς πρόβλημα στη συγκεκριμένη διανομή (FedoraPackages, n.d.).



Η δομή των πακέτων λογισμικού είναι ένα από τα σημεία στα οποία διαφοροποιούνται οι διάφορες διανομές του Linux. Υπάρχουν δύο «οικογένειες» διανομών. Η πρώτη ακολουθεί τη διανομή Red Hat και τα πακέτα λογισμικού έχουν την κατάληξη rpm. Η δεύτερη ακολουθεί τη διανομή Debian και τα πακέτα λογισμικού έχουν την κατάληξη apt (Mohan, 2019).

Η ονομασία ενός πακέτου λογισμικού έχει κάποια συγκεκριμένη μορφή που περιέχει πληροφορίες για το λογισμικό αυτό. Ένα παράδειγμα είναι το παρακάτω όνομα ενός πακέτου λογισμικού.

***nano-4.3-3.fc31.x86\_64.rpm***

Αρχικό τμήμα του ονόματος του πακέτου είναι το όνομα του λογισμικού που στο παραπάνω παράδειγμα είναι ο επεξεργαστής κειμένου **nano**. Στη συνέχεια το όνομα περιέχει την version και το release του συγκεκριμένου λογισμικού (**4.3.3**) καθώς και το γεγονός ότι βρίσκεται στο αποθετήριο για την έκδοση 31 της fedora (**fc31**). Η τελευταία πληροφορία που περιέχει το όνομα είναι ότι το συγκεκριμένο λογισμικό εκτελείται στις εκδόσεις 32 και 64 bit των επεξεργαστών της εταιρίας Intel (**x86\_64**).

Η διαχείριση των λογισμικών στη διανομή Fedora μπορεί να γίνει είτε με γραφικό τρόπο είτε με τη χρήση του DNF. Το DNF (FedoraWiki, 2020) είναι ένα εξειδικευμένο πρόγραμμα που εκτελεί αναζήτηση, εγκατάσταση, απεγκατάσταση και ενημέρωση πακέτων λογισμικού. Επιπρόσθετα, μπορεί και πραγματοποιεί αναζήτηση των εξαρτήσεων (dependency resolution) έτσι ώστε να εγκαθιστά ή να ενημερώνει και τα εξαρτώμενα λογισμικά που είναι απαραίτητα για τη λειτουργία ενός πακέτου λογισμικού. Το DNF αρχικά αναζητά λογισμικά μόνο στα αυθεντικά αποθετήρια λογισμικών της Fedora αλλά μπορούμε να αλλάξουμε τις ρυθμίσεις έτσι ώστε να αναζητά στα αποθετήρια που θέλουμε. Η αναλυτική παρουσίαση όλων των λειτουργιών του DNF παρουσιάζεται στην ιστοσελίδα αναφοράς του (DNF, n.d.).

### 3.2.1 Έλεγχος, ενημέρωση, αναζήτηση και εγκατάσταση πακέτων λογισμικών

Η λειτουργία του DNF γίνεται με τη χρήση της εντολής dnf η οποία για τις περισσότερες ενέργειες απαιτεί δικαιώματα διαχειριστή. Οι ενέργειες που θα παρουσιάσουμε είναι η εγκατάσταση, η αναζήτηση, ο έλεγχος και η ενημέρωση των λογισμικών με τη χρήση του dnf.

#### Εγκατάσταση / Απεγκατάσταση / Ενημέρωση

Η εγκατάσταση ενός λογισμικού μέσω του DNF γίνεται με τη χρήση της εντολής **dnf install**. Η σύνταξη της dnf install για την εγκατάσταση του λογισμικού *htop* στο σύστημα μας παρουσιάζεται στο παρακάτω πλαίσιο (κάποια τμήματα έχουν παραληφθεί για οικονομία χώρου)

```
[root@ekdda ~]# dnf install htop
Dependencies resolved.
...
Total download size: 112 k
Installed size: 250 k
Is this ok [y/N]:y
...
Installed:
  htop-2.2.0-6.fc31.x86_64

Complete!
```

Όπως παρατηρούμε στο παραπάνω πλαίσιο, η `dnf install` αρχικά κάνει έλεγχο των εξαρτήσεων του για να κατεβάσει και να εγκαταστήσει τα λογισμικά που είναι απαραίτητα για την εγκατάσταση του λογισμικού `htop`. Στη συνέχεια εμφανίζει ένα μήνυμα με το συνολικό μέγεθος α) των δεδομένων που θα «κατέβουν» από το αποθετήριο και β) τον χώρο που θα καταλάβει το λογισμικό στο σκληρό δίσκο όταν εγκατασταθεί. Συνεχίζει με ένα μήνυμα για το εάν επιθυμούμε να συνεχίσουμε με την εγκατάσταση ή όχι. Αν απαντήσουμε θετικά στο μήνυμα ξεκινά η εγκατάσταση και εμφανίζονται πληροφορίες προόδου καθώς και το μήνυμα της τελικής ολοκλήρωσης.



Με την χρήση της παραμέτρου `-y` η εντολή `dnf install` πραγματοποιεί απευθείας την εγκατάσταση χωρίς να ζητήσει την έγκριση μας (yes/no).

Η εντολή `dnf install` μας παρέχει περισσότερες δυνατότητες στην εγκατάσταση λογισμικών όπως το ότι το όνομα του λογισμικού μπορεί να περιέχει μεταχαρακτήρες αν θέλουμε να εγκαταστήσουμε περισσότερα από ένα λογισμικά μαζί. Το όνομα που περιέχει μεταχαρακτήρες θα πρέπει να περιέχεται σε μονά εισαγωγικά. Επίσης υπάρχει η χρήσιμη δυνατότητα να δώσουμε το όνομα της εντολής που θέλουμε να εγκαταστήσουμε στο σύστημα μας και το `dnf` να εγκαταστήσει το λογισμικό που την περιέχει. Παραδείγματα αυτών των δυνατοτήτων παρουσιάζονται στο παρακάτω πλαίσιο.

```
[root@ekdda ~]# dnf install 'geany-plugins-*'
[root@ekdda ~]# dnf install /usr/bin/gparted
```

Μία ακόμα δυνατότητα που παρέχει η εντολή `dnf install` είναι η εγκατάσταση μιας συγκεκριμένης έκδοσης ενός λογισμικού αν της δώσουμε το όνομα της. Επιπρόσθετα, μπορούμε να κατεβάσουμε ένα λογισμικό με την εντολή `dnf download` και να το εγκαταστήσουμε στη συνέχεια σε αυτόν ή σε άλλο υπολογιστή με τη χρήση της `dnf install`. Το παρακάτω πλαίσιο παρουσιάζει παραδείγματα εγκατάστασης συγκεκριμένης έκδοσης λογισμικού καθώς και «κατέβασμα» και στη συνέχεια εγκατάσταση ενός λογισμικού.

```
[root@ekdda ~]# dnf install parted-3.2.153-1.fc31
[root@ekdda ~]# dnf download neofetch
Last metadata expiration check: 0:24:58 ago on Wed 09 Dec 2020
03:36:03 PM EET.
neofetch-7.1.0-3.fc31.noarch.rpm                26 kB/s | 90 kB
00:03
[root@ekdda ~]# dnf install ./neofetch-7.1.0-3.fc31.noarch.rpm
```

Η απεγκατάσταση ενός λογισμικού γίνεται με τη χρήση της εντολής **dnf remove**. Κατά την απεγκατάσταση γίνεται έλεγχος ώστε να απομακρυνθούν και οι εξαρτήσεις του λογισμικού που διαγράφεται. Υποστηρίζονται οι ίδιες δυνατότητες όπως και στην εγκατάσταση (μεταχαρακτήρες, όνομα μιας εντολής που υπάρχει στο σύστημα μας). Για παράδειγμα μπορούμε να απεγκαταστήσουμε το λογισμικό *gparted* που είναι εγκατεστημένο στο σύστημα μας με την παρακάτω εντολή.

```
[root@ekdda ~]# dnf remove gparted
```

Η εντολή με την οποία μπορούμε να ελέγξουμε αν υπάρχουν ενημερώσεις για ένα λογισμικό που είναι εγκατεστημένο στο σύστημα μας είναι η **dnf check-update**. Σε περίπτωση που αποφασίσουμε ότι θέλουμε να κάνουμε την ενημέρωση τότε θα χρησιμοποιήσουμε την εντολή **dnf upgrade**. Και οι δύο παραπάνω εντολές αν δεν έχουν το όνομα ενός λογισμικού εκτελούνται για όλα τα λογισμικά που είναι εγκατεστημένα στο σύστημα μας. Η εντολή με την οποία ενημερώνουμε το λογισμικό *graphite2* στην τελευταία του έκδοση είναι η παρακάτω.

```
[root@ekdda ~]# dnf check-update graphite2
Last metadata expiration check: 0:51:17 ago on Wed 09 Dec 2020
06:38:03 PM EET.

graphite2.x86_64                1.3.14-1.fc31
updates
[root@ekdda ~]# dnf upgrade graphite2
```

### Αναζήτηση / παράθεση λογισμικών

Για να αναζητήσουμε κάποιο λογισμικό στα αποθετήρια λογισμικών εκτελούμε την εντολή **dnf search** με τον όρο που θέλουμε να βρούμε. Για παράδειγμα αν θέλουμε να κάνουμε αναζήτηση για τον όρο *caffeine* στα αποθετήρια εκτελούμε την παρακάτω εντολή

```
[root@ekdda ~]# dnf search caffeine
```

Η εντολή αυτή θα ψάξει στα ονόματα και τις περιλήψεις των λογισμικών για τον όρο *caffeine*. Αν θέλουμε να γίνει αναζήτηση και στις περιγραφές αλλά και τα URL's των λογισμικών τότε θα χρησιμοποιήσουμε και τον όρο **all** όπως στο επόμενο πλαίσιο

```
[root@ekdda ~]# dnf search all caffeine
```

Όπως στην εγκατάσταση έτσι και εδώ ο όρος μπορεί να περιέχει μεταχαρακτήρες οι οποίοι όμως πρέπει να περιέχονται μέσα σε απλά εισαγωγικά.

Από τη στιγμή που επιλέξουμε ένα συγκεκριμένο λογισμικό μπορούμε να εμφανίσουμε αναλυτικές πληροφορίες για αυτό με τη χρήση της εντολής **dnf info**. Ενώ αν θέλουμε να εμφανίσουμε πληροφορίες και προηγούμενες εκδόσεις του λογισμικού χρησιμοποιούμε την εντολή **dnf repoquery**. Η χρήση των παραπάνω εντολών για το λογισμικό *transmission* παρουσιάζεται στο παρακάτω πλαίσιο.

```
[root@ekdda ~]# dnf info transmission
[root@ekdda ~]# dnf repoquery transmission --info
```

Εκτός όμως από την εμφάνιση πληροφοριών για ένα λογισμικό πολλές φορές χρειαζόμαστε πληροφορίες για ένα σύνολο λογισμικών όπως για παράδειγμα όλα τα αυτά που είναι εγκατεστημένα στο σύστημα μας. Η εντολή **dnf list** μπορεί να μας δώσει πληροφορίες για τα λογισμικά που υπάρχουν στα αποθετήρια ή στην τοπική βάση των λογισμικών που διαθέτει ο υπολογιστής μας. Οι πληροφορίες που εμφανίζει η **dnf list** μαζί με την αντίστοιχη παράμετρο της είναι οι παρακάτω:

- **all** όλα τα λογισμικά είτε αυτά υπάρχουν στην τοπική βάση του συστήματος είτε στα αποθετήρια
- **installed** τα λογισμικά της τοπικής βάσης του συστήματος
- **available** τα λογισμικά που υπάρχουν στα αποθετήρια
- **extras** τα λογισμικά που υπάρχουν στην τοπική βάση αλλά όχι στα αποθετήρια
- **upgrades** τα λογισμικά που διαθέτουν ενημερώσεις στα αποθετήρια
- **recent** οι πιο πρόσφατες προσθήκες στα αποθετήρια

Η παρακάτω εντολή βεβαιώνει ότι όλα τα λογισμικά που είναι εγκατεστημένα στο σύστημά μας προέρχονται από τα αποθετήρια από τα οποία είναι συνδεδεμένο το **dnf**.

```
[root@ekdda ~]# dnf list extras
Last metadata expiration check: 0:13:38 ago on Wed 09 Dec 2020
05:51:15 PM EET.
```

### Ομάδες λογισμικών (groups)

Εκτός από τα μεμονωμένα λογισμικά υπάρχουν και οι ομάδες λογισμικών που αποτελούνται από πολλά λογισμικά που εξυπηρετούν ένα συγκεκριμένο σκοπό. Χαρακτηριστικό παράδειγμα ομάδων λογισμικού είναι τα διάφορα γραφικά περιβάλλοντα

που μπορούμε να εγκαταστήσουμε (**Xfce**, **KDE**, **Mate**, κ.α.). Για τη διαχείριση των ομάδων λογισμικού υπάρχει η εντολή **dnf group**. Η εντολή αυτή μπορεί στη συνέχεια να πάρει τις εντολές `installed`, `check-update`, `upgrade`, `remove`, `list` όπως και στα κανονικά λογισμικά μόνο που πλέον οι ενέργειες θα γίνονται σε ομάδες λογισμικού.

Για παράδειγμα για να δούμε ποιες ομάδες λογισμικού υπάρχουν είτε στα αποθετήρια είτε στον σύστημά μας εκτελούμε την παρακάτω εντολή:

```
[root@ekdda ~]# dnf group list
```

Μπορούμε να εγκαταστήσουμε μία ομάδα λογισμικού (π.χ. "Security Lab") όπως εγκαθιστούμε ένα κανονικό λογισμικό αν τοποθετήσουμε τον χαρακτήρα `@` μπροστά από το όνομα της ομάδας ή να χρησιμοποιήσουμε την εντολή **dnf group install**.

```
[root@ekdda ~]# dnf install @"Security Lab"
[root@ekdda ~]# dnf group install "Security Lab"
```

Ενδιαφέρον παρουσιάζει και η εντολή **dnf group info** που παρουσιάζει τα λογισμικά από τα οποία αποτελείται μια ομάδα λογισμικού.

Οι βασικές κατηγορίες εντολών που παρέχει το DNF ομαδοποιημένες με βάση τη λειτουργικότητα τους παρουσιάζονται στην Εικόνα 3- 4.



Εικόνα 3- 4: Κατηγορίες εντολών του DNF

## Ιστορικό



Μια χρήσιμη λειτουργία που παρέχει το DNF είναι η παράθεση πλήρους ιστορικού σχετικά με τις λειτουργίες διαχείρισης των λογισμικών που έχουμε εκτελέσει στο σύστημά μας και παρέχεται με την εντολή **dnf history**.

```
[root@ekdda ~]# dnf history
```

Το αποτέλεσμα εκτέλεσης της παραπάνω εντολής είναι η εμφάνιση των ενεργειών που έχουν πραγματοποιηθεί στο σύστημά μας μία σε κάθε γραμμή. Συγκεκριμένα η κάθε γραμμή περιέχει την ακόλουθη πληροφορία

- **ID** Ένας αριθμός που χαρακτηρίζει μοναδικά κάθε ενέργεια
- **Command line** Η εντολή που εκτελέστηκε και προκάλεσε αυτή την ενέργεια
- **Date and time** Η ημερομηνία και ώρα που εκτελέστηκε η συγκεκριμένη ενέργεια
- **Actions** Τι ακριβώς αποτέλεσμα είχε η εντολή (εγκατάσταση, απεγκατάσταση)
- **Altered** Πόσα πακέτα λογισμικών επηρεάστηκαν από τη συγκεκριμένη ενέργεια

Εκτός από την απλή παράθεση των ενεργειών που έχουν γίνει, το ιστορικό του DNF παρέχει τη δυνατότητα ανάρτησης ή επανάληψης μιας συγκεκριμένης ενέργειας, π.χ. της ενέργειας με id 4, που έγινε στο παρελθόν με τη χρήση των παρακάτω εντολών

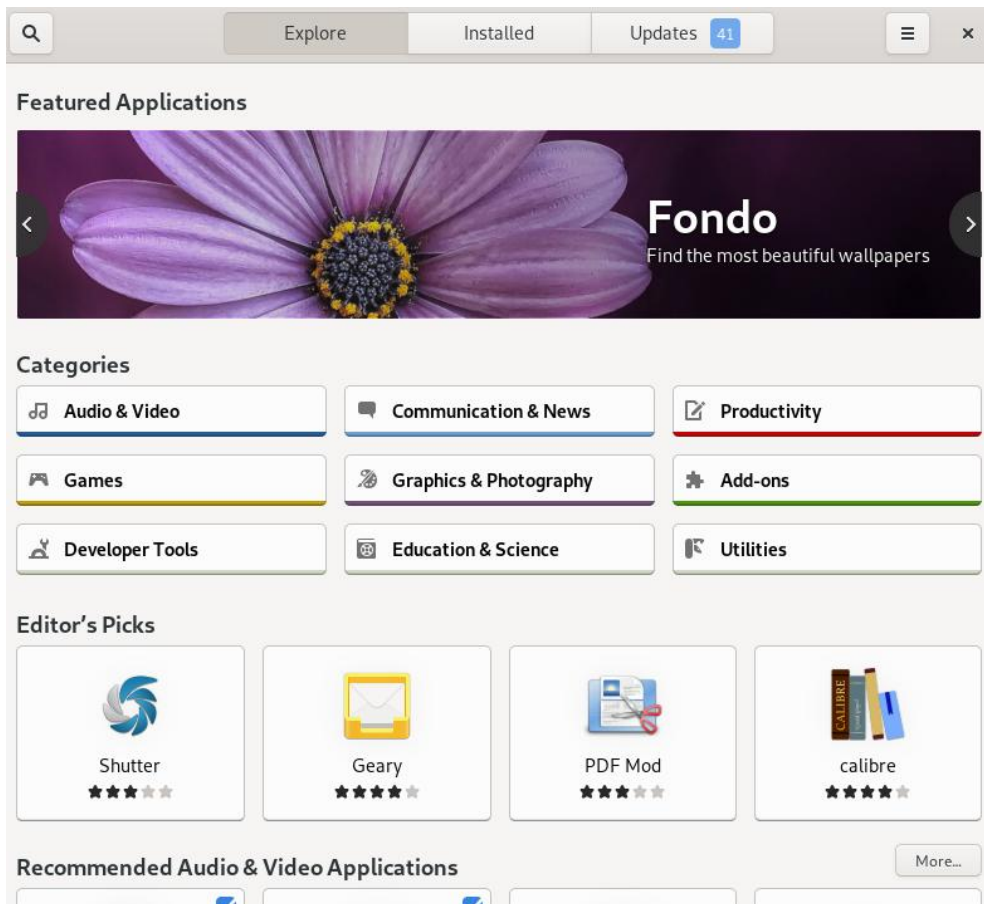
```
[root@ekdda ~]# dnf undo 4  
[root@ekdda ~]# dnf redo 4
```

### 3.2.2 Χρήση γραφικών εργαλείων για τη διαχείριση πακέτων λογισμικών

Η διανομή Fedora παρέχει και ένα γραφικό εργαλείο για τη διαχείριση των λογισμικών του συστήματος μας (Kili, 2018). Για την εκτέλεση του εργαλείου πρέπει να κάνουμε κλικ στο “Activities” και στο πλαίσιο αναζήτησης να γράψουμε τη λέξη “**Software**” και να επιλέξουμε το εικονίδιο που εμφανίζεται και θα παρουσιαστεί το εργαλείο της Εικόνα 3- 5.

Στην καρτέλα “**Explore**” παρουσιάζονται τα λογισμικά που είναι διαθέσιμα για το σύστημα μας. Για τη διευκόλυνση της αναζήτησης μας υπάρχουν κάποιες κατηγορίες που περιέχουν λογισμικά με την αντίστοιχη λειτουργικότητα (π.χ. “Audio & Video” ή “Developer Tools”). Επίσης υπάρχουν κάποια προτεινόμενα λογισμικά από τους δημιουργούς του εργαλείου “Software”. Αν θέλουμε να αναζητήσουμε κάποιο συγκεκριμένο λογισμικό επιλέγουμε το εικονίδιο αναζήτησης (μεγεθυντικός φακός) που βρίσκεται πάνω αριστερά και γράφουμε τον όρο που θέλουμε να βρούμε.

Δίπλα από την καρτέλα “Explore” υπάρχει η καρτέλα “**Installed**”. Αυτή περιέχει όλα τα λογισμικά που είναι εγκατεστημένα στο σύστημα μας. Για το καθένα από τα λογισμικά αυτά μας δείχνει το μέγεθος που καταλαμβάνει στον χώρο αποθήκευσης καθώς και μας παρέχει τη δυνατότητα απεγκατάστασης του. Τελευταία καρτέλα είναι η “**Updates**” που περιέχει έναν κατάλογο με τα λογισμικά που είναι εγκατεστημένα στο σύστημα μας αλλά παράλληλα υπάρχουν γι’ αυτά διαθέσιμες ενημερώσεις στα αποθετήρια.



Εικόνα 3- 5: Το εργαλείο "Software" για τη διαχείριση λογισμικών

Η ρύθμιση των αποθετηρίων στα οποία αναζητά λογισμικά το εργαλείο Software μπορεί να γίνει από το πλήκτρο ρυθμίσεων, πάνω δεξιά, και την επιλογή **"Software Repositories"**. Εκεί εμφανίζεται μια λίστα με τα αποθετήρια το αν είναι ενεργοποιημένη ή όχι η αναζήτηση σε αυτά. Την επιλογή αυτή μπορούμε να την αλλάξουμε με ένα κλικ στο κατάλληλο πλήκτρο. Τέλος μας δίνει τη δυνατότητα να εγκαταστήσουμε ένα καινούργιο αποθετήριο με τη χρήση του πλήκτρου "Install".

### 3.2.3 Διαχείριση repositories

Όπως έχουμε ήδη αναφέρει, στην προκαθορισμένη ρύθμιση του το DNF έχει εγκατεστημένη μία λίστα από εξουσιοδοτημένα αποθετήρια της διανομής Fedora. Κάποια από τα αποθετήρια είναι ενεργά ενώ κάποια άλλα ανενεργά. Για να δούμε τα αποθετήρια τα οποία είναι ενεργά εκτελούμε την εντολή **dnf repolist**.

```
[root@ekdda ~]# dnf repolist
repo id           repo name          status
*fedora           Fedora 31 - x86_64 56,255
*fedora-modular   Fedora Modular 31 - x86_64 142
*updates          Fedora 31 - x86_64 - Updates 22,403
*updates-modular  Fedora Modular 31 - x86_64 - Updates 139
```

Όπως βλέπουμε το κάθε αποθετήριο έχει ένα αναγνωριστικό “*repo id*” το οποίο το ξεχωρίζει από τα υπόλοιπα. Για να δούμε και τα ανενεργά αποθετήρια που είναι εγκατεστημένα στο DNF προσθέτουμε την παράμετρο **all** στην τελευταία εντολή.

Μπορούμε να μεταβάλουμε από ενεργό / ανενεργό ένα αποθετήριο, π.χ. το “*updates*”, με τη χρήση των παρακάτω εντολών

```
[root@ekdda ~]# dnf config-manager --set-enabled "updates"
[root@ekdda ~]# dnf config-manager --set-disabled "updates"
```

Οι ρυθμίσεις του DNF βρίσκονται στο αρχείο **/etc/dnf/dnf.conf**. Αυτό το αρχείο περιέχει ένα τμήμα [**main**] που περιλαμβάνει τις γενικές ρυθμίσεις του DNF που είναι κοινές για όλα τα αποθετήρια. Μπορεί να περιέχει όμως και ένα ή περισσότερα [**repository**] τμήματα που περιέχουν ρυθμίσεις για το αποθετήριο “*repository*”. Συνίσταται όμως η παράθεση των ρυθμίσεων για το κάθε αποθετήριο σε ξεχωριστό αρχείο που βρίσκεται στο **/etc/yum.repos.d/** κατάλογο.

Τα αρχεία των ρυθμίσεων τόσο για το τμήμα “*main*” όσο και για τα τμήματα “*repositories*” είναι μια λίστα από όνομα-τιμή ζευγάρια που καθορίζουν τη λειτουργικότητα του DNF. Μερικές από τις τιμές που μπορούν να οριστούν για το *main* τμήμα είναι οι παρακάτω:

- ***debugLevel=τιμή***  
Όπου η τιμή μπορεί να είναι από 0 μέχρι το 10. Η τιμή 0 καθορίζει στο DNF να μην εμφανίζει μηνύματα debugging κατά την διάρκεια εγκατάστασης ενός λογισμικού, ενώ η τιμή 10 εμφανίζει τα μηνύματα όσο πιο αναλυτικά μπορεί (προκαθορισμένη τιμή είναι το 2).
- ***exclude=όνομα\_πακέτου [όνομα\_πακέτου2]***  
Καθορίζουμε ότι τα συγκεκριμένα λογισμικά δεν θέλουμε να εγκατασταθούν στο σύστημα μας.
- ***gpgcheck=τιμή***  
Η τιμή μπορεί να είναι 0 οπότε δεν γίνεται έλεγχος της ακεραιότητας μέσω της υπογραφής ενώ με την τιμή 1 γίνεται έλεγχος για όλα τα λογισμικά που εγκαθίστανται. Αν υπάρχει η ίδια ρύθμιση στο *main* και σε ένα τοπικό αποθετήριο υπερισχύει η ρύθμιση του τοπικού αποθετηρίου.
- ***installonlypkgs=λίστα\_λογισμικού\_με\_κενά***  
Καθορίζονται τα λογισμικά που μπορούν να γίνουν εγκατάσταση αλλά όχι ενημέρωση. Σε αυτή την κατηγορία ανήκουν τα *kernel* του συστήματος.

Αντίστοιχα οι ρυθμίσεις που αφορούν κάποιο συγκεκριμένο αποθετήριο καταγράφονται μετά από μια γραμμή της μορφής [*repository*] και περιλαμβάνουν μερικές από τις παρακάτω βασικές ρυθμίσεις:

- ***name = όνομα\_αποθετηρίου***  
Είναι ένα κείμενο που περιγράφει το συγκεκριμένο αποθετήριο.
- ***parameter=repository\_url***

Έχει σαν τιμή μια url μια διεύθυνση στην οποία βρίσκονται τα λογισμικά από το συγκεκριμένο αποθετήριο.

- ***enabled=τιμή***

Αν έχει την τιμή 1 το συγκεκριμένο repository είναι ενεργοποιημένο και το dnf το χρησιμοποιεί στην αναζήτηση.

Για να προσθέσουμε ένα καινούργιο αποθετήριο σε αυτά του DNF πρέπει να προσθέσουμε ένα καινούργιο αρχείο repo για το συγκεκριμένο αποθετήριο στον κατάλογο **/etc/yum.repos.d/**. Συνήθως τα αποθετήρια έχουν έτοιμο δικό τους αρχείο repo το οποίο προστίθεται με την παρακάτω εντολή.

```
[root@ekdda ~]# dnf config-manager - --add-repo "repo_url"
```

### 3.3 Διαχείριση αρχείων / καταλόγων

Ένα από τα πολυτιμότερα αγαθά που υπάρχουν σε ένα σύστημα Linux Server είναι τα δεδομένα των χρηστών του. Σε αυτή την υποενότητα θα μελετήσουμε τη δημιουργία αντιγράφων ασφαλείας καθώς και τη διαχείριση των αποθηκευτικών μέσων του συστήματος.

#### 3.3.1 Αντίγραφα ασφαλείας

Η δημιουργία αντιγράφων ασφαλείας του συστήματος είναι μία από τις κύριες ασχολίες ενός διαχειριστή συστήματος. Υπάρχουν διαφορετικοί τύποι αντιγράφων ασφαλείας που παρουσιάζουν διαφορετικά χαρακτηριστικά σε ό,τι αφορά τη δημιουργία των αντιγράφων ασφαλείας όσο και την επαναφορά των δεδομένων (Fellows & Crocetti, 2020). Μερικοί από τους τύπους αντιγράφων ασφαλείας είναι οι παρακάτω:

- **Εικόνα συστήματος (System Image)**. Είναι ένα πλήρες αντίγραφο ενός δίσκου η μίας κατάτμησης που περιλαμβάνει τα πάντα όπως τις βιβλιοθήκες του συστήματος, τα αρχεία ρυθμίσεων καθώς και τα δεδομένα των χρηστών. Σκοπός αυτό του είδους αντιγράφου ασφαλείας είναι η γρήγορη αποκατάσταση του συστήματος σε μία κατάσταση που μπορεί να λειτουργήσει. Το αρχείο που δημιουργείται ονομάζεται αρχείο **iso**.
- **Πλήρες αντίγραφο ασφαλείας (Full backup)**. Σε ένα πλήρες αντίγραφο ασφαλείας αντιγράφονται όλα τα δεδομένα ανεξάρτητα από το πότε αυτά τροποποιήθηκαν. Απαιτεί αρκετό χρόνο για τη δημιουργία του αντιγράφου ασφαλείας καθώς και περισσότερο αποθηκευτικό χώρο αλλά η επαναφορά των δεδομένων γίνεται άμεσα.
- **Αυξητικό αντίγραφο ασφαλείας (Incremental backup)**. Σε ένα αυξητικό αντίγραφο ασφαλείας δεν αντιγράφονται όλα τα αρχεία αλλά μόνο αυτά που τροποποιήθηκαν από το τελευταίο αντίγραφο ασφαλείας (οποιοδήποτε τύπου). Αυτός ο τρόπος δημιουργίας αντιγράφων ασφαλείας απαιτεί λιγότερο χρόνο και αποθηκευτικό χώρο για τη δημιουργία του αλλά η ανάκτηση δεδομένων από αυτό είναι μια επίπονη διαδικασία.
- **Αντίγραφο ασφαλείας διαφοροποίησης (Differential backup)**. Ένα αντίγραφο ασφαλείας διαφοροποίησης είναι κάτι ενδιάμεσο μεταξύ ενός πλήρες και ενός

αυξητικού. Σε αυτή την περίπτωση κρατάμε αντίγραφα των αρχείων που τροποποιήθηκαν πριν από το τελευταίο πλήρες αντίγραφο ασφαλείας. Με αυτόν το τρόπο το αντίγραφο ασφαλείας διαφοροποίησης έχει λιγότερο κόπο από το πλήρες στη δημιουργία του και λιγότερο από το αυξητικό στην ανάκτηση των δεδομένων.

Η δημιουργία αντιγράφων ασφαλείας πολλές φορές καταλήγει στη δημιουργία αρχείων με μεγάλο μέγεθος. Επομένως, πολύ συχνά συνδυάζεται, με τη συμπίεση των αρχείων που δημιουργούνται. Μερικές από τις μεθόδους συμπίεσης που υποστηρίζονται στο λειτουργικό σύστημα Linux είναι οι παρακάτω (Muntaha, 2020):

- **gzip**. Για τη συμπίεση ενός αρχείου χρησιμοποιείται η εντολή gzip. Η επέκταση των αρχείων που δημιουργεί είναι gz.
- **bzip2**. Χρειάζεται λίγο περισσότερο χρόνο για τη συμπίεση από το gzip αλλά πραγματοποιεί καλύτερη συμπίεση. Η εντολή που χρησιμοποιείται είναι η bzip2 και δημιουργεί αρχεία με κατάληξη bz2.
- **xz**. Πραγματοποιεί καλύτερη συμπίεση τόσο από το gzip όσο και από το bzip2. Χρησιμοποιείται η εντολή xz και δημιουργεί αρχεία με την ίδια κατάληξη.
- **zip**. Είναι το παλαιότερο πρόγραμμα συμπίεσης. Χρησιμοποιεί την εντολή zip και δημιουργεί αρχεία με την ίδια κατάληξη.

Στο παρακάτω πλαίσιο εκτελούμε μια σύγκριση των μεθόδων συμπίεσης. Αρχικά αντιγράφουμε το αρχείο `/var/log/wtmp` στον κατάλογο `~/backup`. Στη συνέχεια δημιουργούμε τέσσερα αντίγραφα του αρχείου αυτού τα οποία και συμπιέζουμε το καθένα με διαφορετική μέθοδο. Παρατηρούμε ότι υπάρχει μια διαφορά στη σύνταξη της εντολής για τη συμπίεση με τη μέθοδο zip. Στο τέλος συγκρίνουμε τα μεγέθη των αρχείων που δημιουργήθηκαν από την κάθε μέθοδο σε σχέση με το αρχικό. Όπως διαπιστώνουμε το μεγαλύτερο ποσοστό συμπίεσης το πετυχαίνουμε με τη μέθοδο xz ενώ η μέθοδος zip έχει το μικρότερο ποσοστό συμπίεσης.

```
[user@ekdda backup]$ cp /var/log/wtmp .
[user@ekdda backup]$ cp wtmp wtmp1
[user@ekdda backup]$ cp wtmp wtmp2
[user@ekdda backup]$ cp wtmp wtmp3
[user@ekdda backup]$ cp wtmp wtmp4
[user@ekdda backup]$ gzip wtmp1
[user@ekdda backup]$ bzip2 wtmp2
[user@ekdda backup]$ xz wtmp3
[user@ekdda backup]$ zip wtmp4.zip wtmp4
  adding: wtmp4 (deflated 97%)
[user@ekdda backup]$ ls -l wtmp?.*
-rw-rw-r--. 1 user user 1384 Dec 12 08:47 wtmp1.gz
-rw-rw-r--. 1 user user 1299 Dec 12 08:47 wtmp2.bz2
-rw-rw-r--. 1 user user 1068 Dec 12 08:47 wtmp3.xz
-rw-rw-r--. 1 user user 1520 Dec 12 08:48 wtmp4.zip
[user@ekdda backup]$ ls -l wtmp
-rw-rw-r--. 1 user user 40320 Dec 12 08:47 wtmp
```

### 3.3.2 Διαχείριση αντιγράφων ασφαλείας

Η δημιουργία αντιγράφων ασφαλείας στο λειτουργικό σύστημα Linux μπορεί να γίνει με τη χρησιμοποίηση διαφόρων εντολών κάθε μία από τις οποίες είναι αποδοτικότερη ανάλογα τις απαιτήσεις που έχουμε. Στη συνέχεια θα παρουσιάσουμε την εντολή **tar** και τις εντολές **rsync** και **dd**.

#### Η εντολή tar

Το πρόγραμμα **tar (tape archiver)** (FedoraDocs-Tar, n.d.) είναι μια εντολή - εργαλείο που «ενώνει» πολλά αρχεία και καταλόγους σε ένα καινούργιο αρχείο (**archive**) που έχει την επέκταση **tar**. Επιπρόσθετα, η εντολή **tar** μπορεί να συμπιέζει το αρχείο αυτό που δημιουργεί με μία μέθοδο συμπίεσης και τότε παράγει ένα **tarball** αρχείο.

Η γενική μορφή χρήσης της εντολής **tar** είναι η παρακάτω:

```
tar -[παράμετροι] όνομα.tar αρχεία_και_κατάλογοι
```

Όπου *όνομα.tar* είναι το αρχείο **archive** και περιέχει τα περιεχόμενα των *αρχεία\_και\_κατάλογοι*. Μερικές από τις παραμέτρους που παρέχει η εντολή **tar** παρουσιάζονται στον Πίνακα 3- 4.

| Παράμετρος              | Λειτουργικότητα   |
|-------------------------|---|
| <b>-c</b>               | Δημιουργεί (create) ένα καινούργιο archive              |
| <b>-x</b>               | Επαναφέρει τα αρχεία που περιέχει ένα tar               |
| <b>-f</b>               | Το όνομα του archive που θα δημιουργήσει – αποσυμπιέσει |
| <b>-v</b>               | Εμφανίζει την πρόοδο στη δημιουργία του archive         |
| <b>-g όνομα_αρχείου</b> | Δημιουργεί incremental backup                           |
| <b>-J</b>               | Συμπιέζει με τη μέθοδο xz                               |
| <b>-z</b>               | Συμπιέζει με τη μέθοδο gzip                             |
| <b>-j</b>               | Συμπιέζει με τη μέθοδο bzip2                            |

Πίνακας 3- 4: Παράμετροι της εντολής **tar**

Στο παρακάτω πλαίσιο παρουσιάζεται η δημιουργία ενός αρχείου **tar** τόσο χωρίς συμπίεση όσο και με συμπίεση με τη χρήση της μεθόδου **gzip**.

```
[user@ekdda backup]$ touch file{1..3}.txt
[user@ekdda backup]$ tar -cvf data.tar file*
file1.txt
file2.txt
file3.txt
[user@ekdda backup]$ tar -zcvf data.tgz file*
file1.txt
file2.txt
file3.txt
[user@ekdda backup]$ ls -l data*
-rw-rw-r--. 1 user user 10240 Dec 12 09:21 data.tar
-rw-rw-r--. 1 user user 132 Dec 12 09:21 data.tgz
```

Η δημιουργία ενός αυξητικού αντίγραφου ασφαλείας με τη χρήση της παραμέτρου **g** παρουσιάζεται στο παρακάτω πλαίσιο. Η παράμετρος **g** παίρνει σαν όρισμα το όνομα ένα αρχείου μεταδεδομένων που περιέχει τις **χρονοσημάνσεις (timestamps)** των αρχείων που περιέχονται στο αρχείο **tar**. Την πρώτη φορά, μια και δεν υπάρχει το αρχείο με τις χρονοσημάνσεις, θα εκτελέσει πλήρες αντίγραφο ασφαλείας. Την επόμενη φορά που θα εκτελεστεί η εντολή θα χρησιμοποιήσει αυτό το αρχείο χρονοσημάνσεων για να προσθέσει στο αρχείο **tar** μόνο τα αρχεία αυτά που έχουν μεταβληθεί από το τελευταίο αντίγραφο ασφαλείας. Στο παράδειγμα μας μόνο το *file2* έχει τροποποιηθεί και είναι το μόνο που προστίθεται στο αυξητικό αντίγραφο ασφαλείας.

```
[user@ekdda backup]$ tar -g data.time -cvf data.tar file*
file1.txt
file2.txt
file3.txt
[user@ekdda backup]$ echo "Update" >> file2.txt
[user@ekdda backup]$ tar -g data.time -cvf data_incr.tar file*
file2.txt
```

Η εξαγωγή των αρχείων που περιέχει ένα αρχείο **tar** γίνεται με τη χρήση της παραμέτρου **x** και παρουσιάζεται στο παρακάτω πλαίσιο

```
[user@ekdda backup]$ ls
data.tar
[user@ekdda backup]$ tar -xvf data.tar
file1.txt
file2.txt
file3.txt
[user@ekdda backup]$ ls
data.tar file1.txt file2.txt file3.txt
```

Υπάρχουν και παράμετροι της **tar** που επιτρέπουν τον έλεγχο των περιεχομένων ενός αντιγράφου ασφαλείας οι οποίες παρουσιάζονται στον Πίνακα 3- 5.

| Παράμετρος | Λειτουργικότητα  |
|------------|--|
| <b>-d</b>  | Ελέγχει τα αρχεία που υπάρχουν σε ένα <b>tar</b> με εξωτερικά αρχεία |
| <b>-t</b>  | Εμφανίζει τα περιεχόμενα ενός <b>tar</b> αρχείου                     |

Πίνακας 3- 5: Παράμετροι εμφάνισης της εντολής **tar**

Το παράδειγμα στο παρακάτω πλαίσιο εμφανίζει τη λειτουργικότητα των δύο παραπάνω επιλογών.



```
[user@ekdda backup]$ tar -tf data.tar
file1.txt
file2.txt
file3.txt
[user@ekdda backup]$ echo "Update">>file3.txt
[user@ekdda backup]$ tar -df data.tar file*
file3.txt: Mod time differs
file3.txt: Size differs
```

Όπως παρατηρούμε η παράμετρος **t** εμφανίζει τα περιεχόμενα του αρχείου tar ενώ η παράμετρος **d** εμφανίζει μόνο τα αρχεία που περιέχονται στο αρχείο tar και διαφέρουν από τα εξωτερικά αρχεία με τα οποία συγκρίνονται. Η παράμετρος **d** είναι πολύ χρήσιμη στο να ελέγξουμε αν κάποιο αρχείο που περιέχεται στο tar έχει μεταβληθεί έτσι ώστε να δημιουργήσουμε καινούργιο αντίγραφο ασφαλείας.

### Η εντολή rsync

Ένα ακόμα χρήσιμο εργαλείο για τη δημιουργία αντιγράφων ασφαλείας είναι και η εντολή **rsync** (Jentis, 2020). Σκοπός της εντολής rsync είναι η αντιγραφή πολλών και μεγάλων σε μέγεθος αρχείων από ένα αποθηκευτικό μέσο σε ένα άλλο και παρουσιάζει τα παρακάτω βασικά πλεονεκτήματα:

- Μπορεί να αντιγράψει τα αρχεία είτε τοπικά είτε να χρησιμοποιήσει έναν απομακρυσμένο Server
- Μπορεί να αντιγράψει ειδικά αρχεία (π.χ. device)
- Μπορεί να διατηρεί τις χρονοσημάνσεις τα δικαιώματα και τους ιδιοκτήτες των αρχείων
- Δημιουργεί αυξητικά αντίγραφα ασφαλείας. Δηλαδή την πρώτη φορά μεταφέρει όλα τα δεδομένα την επόμενη φορά όμως μεταφέρει μόνο αυτά που έχουν μεταβληθεί

Η γενική μορφή χρήσης της rsync είναι η παρακάτω:

***rsync [παράμετροι] αφετηρία προορισμός***

Όπου αφετηρία είναι τα αρχεία / κατάλογοι που θέλουμε να δημιουργήσουμε αντίγραφο ασφαλείας και προορισμός η θέση στην οποία θέλουμε να αντιγραφούν. Οι παράμετροι εξειδικεύουν τη λειτουργία της rsync και μερικές από αυτές παρουσιάζονται στον Πίνακα 3- 6.

| Παράμετρος | Λειτουργικότητα  |
|------------|--|
| <b>-r</b>  | Μεταφέρει και τα αρχεία των υποκαταλόγων που περιέχονται |
| <b>-D</b>  | Διατηρεί τα ειδικά αρχεία (π.χ. αρχεία device)           |
| <b>-g</b>  | Διατηρεί την ίδια ομάδα ιδιοκτήτη                        |
| <b>-l</b>  | Αντιγράφει τα συμβολικά link                             |
| <b>-o</b>  | Διατηρεί τον ιδιοκτήτη                                   |
| <b>-p</b>  | Διατηρεί τις άδειες του αρχείου                          |



|           |  |
|-----------|--|
| <b>-t</b> | Διατηρεί τις χρονοσημάνσεις  |
| <b>-v</b> | Παρέχει πληροφορίες κατά τη διάρκεια εκτέλεσης της εντολής   |
| <b>-a</b> | Ισοδύναμο με τον συνδυασμό παραμέτρων <b>rlptgoD</b> . Χρησιμοποιείται συχνά                             |
| <b>-n</b> | Δείχνει τι θα εκτελέσει χωρίς να το κάνει (χρησιμοποιείται μαζί με <b>verbose</b> για να κάνουμε έλεγχο) |

Πίνακας 3- 6: Παράμετροι της εντολής `rsync`

Για παράδειγμα αν στον τρέχων κατάλογο υπάρχει ένας κατάλογος με το όνομα *data* για τον οποίο πρέπει να δημιουργηθεί αντίγραφο ασφαλείας μέσα στον κατάλογο */tmp/backups* η εντολή `rsync` που θα το δημιουργήσει παρουσιάζεται στο παρακάτω πλαίσιο

```
[user@ekdda backup]$ rsync -av data/ /tmp/backups
```

Η χρήση του `/` μετά το *data* καθορίζει ότι μέσα στον κατάλογο *backups* θα δημιουργηθούν τα αρχεία του *data* αλλά όχι ο ίδιος ο κατάλογος *data*.

Μία ιδιαίτερα χρήσιμη δυνατότητα της εντολής `rsync` είναι ότι τα αντίγραφα ασφαλείας μπορούν να δημιουργούνται σε έναν απομακρυσμένο υπολογιστή. Για παράδειγμα η δημιουργία ενός αντιγράφου ασφαλείας του καταλόγου *dir1* σε ένα απομακρυσμένο υπολογιστή γίνεται με τη χρήση της εντολής που παρουσιάζεται στο παρακάτω πλαίσιο:

```
rsync -a ~/dir1 username@remote_host:destination_directory
```

όπου *username* είναι το αναγνωριστικό του χρήστη στον απομακρυσμένο υπολογιστή, *remote\_host* είναι το όνομα του απομακρυσμένου υπολογιστή και *destination\_directory* ο κατάλογος στον απομακρυσμένο υπολογιστή όπου θα δημιουργηθεί το αντίγραφο ασφαλείας.

Αντίθετα η επαναφορά ενός αντιγράφου ασφαλείας που βρίσκεται σε έναν απομακρυσμένο υπολογιστή στον τοπικό υπολογιστή γίνεται με τη χρήση της εντολής

```
rsync -a username@remote_host:destination_directory  
place_to_sync_on_local_machine
```

όπου οι παράμετροι της εντολής `rsync` είναι ανάλογοι με την προηγούμενη περίπτωση. Επιπρόσθετα, στην περίπτωση χρήσης απομακρυσμένου υπολογιστή μπορούμε να καθορίσουμε το πρωτόκολλο μεταφοράς που θα χρησιμοποιήσουμε με την παράμετρο `-e`. Για παράδειγμα η δημιουργία αντιγράφου ασφαλείας με τη χρήση του `ssh` γίνεται με την παρακάτω εντολή

```
rsync -ae ssh ~/dir1 username@remote_host:destination_directory
```

Μερικές ακόμα παράμετροι που μπορεί να λάβει η εντολή `rsync` παρουσιάζονται στον Πίνακα 3- 7.

| Παράμετρος              | Λειτουργικότητα  |
|-------------------------|--|
| <code>--progress</code> | Εμφανίζει την πρόοδο της <code>rsync</code>  |
| <code>--exclude</code>  | Αφαιρεί συγκεκριμένα αρχεία – καταλόγους από την αντιγραφή                               |
| <code>--delete</code>   | Διαγράφει αρχεία στον κατάλογο προορισμού αν αυτά έχουν διαγραφεί από το αρχικό κατάλογο |

Πίνακας 3- 7: Ειδικές παράμετροι της εντολής `rsync`

Στο παρακάτω παράδειγμα έχουμε δύο καταλόγους, τον *data* που περιέχει τα αρχεία για τα οποία θέλουμε να δημιουργήσουμε αντίγραφο ασφαλείας και τον *backup* στον οποίο θέλουμε να μεταφέρουμε τα αντίγραφα αυτά. Αρχικά εκτελούμε την εντολή `rsync` και στη συνέχεια μεταβάλλουμε ένα αρχείο. Βλέπουμε ότι την επόμενη φορά που εκτελούμε την `rsync` αντιγράφεται μόνο το αρχείο αυτό που μεταβλήθηκε (αυξητικό αντίγραφο ασφαλείας).

```
[user@ekdda backup]$ rsync -av data/ backup
sending incremental file list
./
file1.txt
file2.txt
file3.txt

sent 234 bytes  received 76 bytes  620.00 bytes/sec
total size is 0  speedup is 0.00
[user@ekdda backup]$ echo "update file">> data/file2.txt
[user@ekdda backup]$ rsync -av data/ backup
sending incremental file list
file2.txt

sent 177 bytes  received 35 bytes  424.00 bytes/sec
total size is 12  speedup is 0.06
```

Στο παρακάτω παράδειγμα παρατηρούμε τη χρήση των παραμέτρων **delete** που διαγράφει τα αρχεία που μετακίνησε καθώς και την **exclude** που αγνοεί κατά τη μετακίνηση αρχεία που το όνομα τους ανήκει σε κάποιο καθορισμένο πρότυπο

```
[user@ekdda backup]$ touch data/file.tmp
[user@ekdda backup]$ rsync -av --delete --exclude *.tmp data/
backup
sending incremental file list
./
file1.txt
file2.txt
file3.txt

sent 262 bytes  received 76 bytes  676.00 bytes/sec
total size is 12  speedup is 0.04
```

### Η εντολή dd

Υπάρχει και η εντολή **dd** (Docile, 2020) που επιτρέπει την πιστή αντιγραφή των περιεχομένων ενός δίσκου ή μίας κατάτμησης δίσκου. Χρησιμοποιείται σε περιπτώσεις όπου θέλουμε να επεξεργαστούμε συνολικά τα δεδομένα ενός δίσκου όπως π.χ. να δημιουργήσουμε πιστό αντίγραφο ενός δίσκου σε περίπτωση που έχει καταστραφεί ή για να διαγράψουμε τα δεδομένα σε έναν δίσκο που θέλουμε να στείλουμε στην ανακύκλωση. Η σύνταξη της εντολής dd είναι η ακόλουθη:

***dd if=συσκευή-εισόδου of=συσκευή-εξόδου [παράμετροι]***

Η dd διαβάζει ένα *block* δεδομένων από την συσκευή/αρχείο *if* και το αντιγράφει στην συσκευή/αρχείο *of*. Επαναλαμβάνει τη διαδικασία μέχρι να τελειώσουν τα δεδομένα από την αρχική συσκευή. Η εντολή dd μπορεί να πάρει κάποιες παραμέτρους που εξειδικεύουν τη λειτουργία της και οι βασικότερες από αυτές παρουσιάζονται στον Πίνακα 3- 8.

| Παράμετρος           | Λειτουργία   |
|----------------------|--|
| <b>-bs=BYTES</b>     | Καθορίζει το μέγιστο μέγεθος του block που διαγράφει και διαβάζει σε κάθε φορά |
| <b>-count=N</b>      | Ο αριθμός των block που θα αντιγράψει  |
| <b>-status=LEVEL</b> | Το μέγεθος των πληροφοριών που θα εμφανίσει στο stderr                         |
| <b>conv=PAR</b>      | Πραγματοποιεί κάποια μετατροπή κατά την αντιγραφή των δεδομένων                |

Πίνακας 3- 8: Παράμετροι της εντολής dd

Η τιμή της παραμέτρου **status** καθορίζει τις πληροφορίες που θα εμφανίσει και είναι *none*, εμφανίζει μόνο λάθη, *noxfer* δεν εμφανίζει τελικά στατιστικά, *progress* εμφανίζει την πρόοδο κατά τη λειτουργία της.

Μία χρήσιμη εφαρμογή της εντολής dd είναι η δημιουργία αντίγραφου ασφαλείας του MBR table ενός δίσκου.

```
[root@ekdda ~]# dd if=/dev/sda of=mbr.img bs=512 count=1
1+0 records in
1+0 records out
512 bytes copied, 9.1094e-05 s, 5.6 MB/s
```

Στο παραπάνω παράδειγμα δημιουργούμε αντίγραφο του δίσκου **/dev/sda** καθορίζουμε το μέγεθος του *block* που θα αντιγράψει σε 512bytes καθώς και ορίζουμε ότι θα κάνει την αντιγραφή μόνο για 1 block (*count=1*). Επομένως το τελικό αποτέλεσμα είναι να αντιγράψει τα 512 πρώτα bytes του **/dev/sda** που περιέχουν και τον **partition table** σε έναν MBR δίσκο.

Μία άλλη περίπτωση χρήσιμη εφαρμογή της *dd* είναι η οριστική διαγραφή των περιεχομένων ενός δίσκου πριν τον στείλουμε στην ανακύκλωση.

```
[root@ekdda ~]# dd if=/dev/zero bs=1M of=/dev/sda1
dd: error writing '/dev/sda1': No space left on device
1954+0 records in
1953+0 records out
2048000000 bytes (2.0 GB, 1.9 GiB) copied, 1.04153 s, 2.0 GB/s
```

Εδώ χρησιμοποιούμε ένα «ειδικό» αρχείο το **/dev/zero** το οποίο είναι ένα «εικονικό» αρχείο που περιέχει όλο μηδενικά. Αυτό το αρχείο το αντιγράφουμε στο **/dev/sda1**. Στην πραγματικότητα αυτό που γίνεται είναι να γεμίσει το η πρώτη κατάτμηση του πρώτου δίσκου (**/dev/sda1**) με μηδενικά, δηλαδή να τον διαγράψουμε. Το μήνυμα λάθους που εμφανίζεται οφείλεται στο γεγονός ότι το εικονικό αρχείο έχει άπειρο μέγεθος οπότε εμφανίζει ότι γέμισε ο δίσκος προορισμού χωρίς να έχει τελειώσει το αρχείο αφετηρίας.

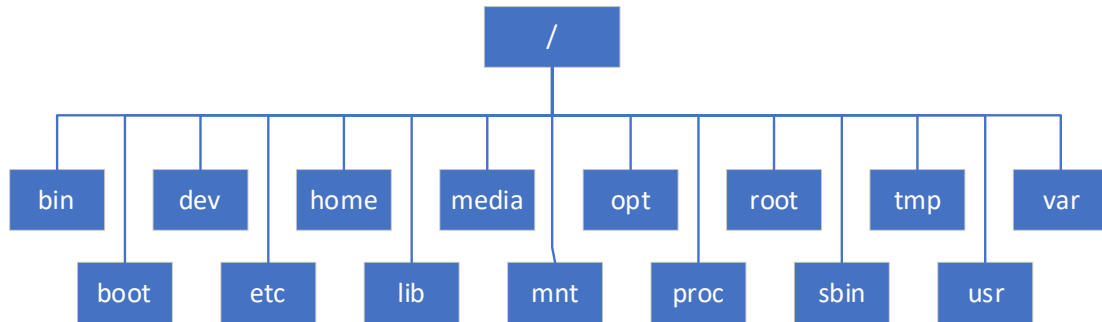
Ένα τελευταίο παράδειγμα χρήσης της *dd* είναι στο παρακάτω πλαίσιο όπου παρουσιάζεται η δημιουργία ενός αντίγραφου ασφαλείας ενός ολόκληρου δίσκου

```
[root@ekdda ~]# dd if=/dev/sda of=/dev/sdb bs=4096
conv=noerror, sync
97280+0 records in
97280+0 records out
99614720 bytes (100MB) copied, 1.04153 s, 2.0 GB/s
```

Εδώ πραγματοποιούμε αντίγραφο ασφαλείας του δίσκου **/dev/sda** στον **/dev/sdb** δίσκου. Θέτουμε το μέγεθος *block* σε 4096bytes (προσοχή το μέγεθος θα πρέπει να είναι πολλαπλάσιο του 512). Η παράμετρος **noerror** καθορίζει να συνεχίσει η διαδικασία και σε περίπτωση που βρει λάθη στην ανάγνωση των δεδομένων και η **sync** να πραγματοποιήσει *synchronized io*.

### 3.3.3 Η δομή του ριζικού δέντρου στο Linux

Ένας σύγχρονος σκληρός δίσκος μπορεί να αποθηκεύσει μερικές εκατοντάδες χιλιάδες αρχεία μέσα του. Όπως είναι φυσικό, χωρίς κάποιας μορφής οργάνωση, η διαχείριση όλης αυτής της πληροφορίας θα ήταν αδύνατη. Η οργάνωση των αποθηκευμένων πληροφοριών μέσα σε ένα μέσο αποθήκευσης ονομάζεται σύστημα αρχείων (**filesystem**). Το Linux, όπως και τα άλλα λειτουργικά συστήματα, χρησιμοποιεί τη δενδρική δομή όπου υπάρχουν κατάλογοι που περιέχουν μέσα τους αρχεία ή και άλλους καταλόγους. Το αποτέλεσμα είναι η δημιουργία ενός συστήματος αρχείων που παρομοιάζει με την Εικόνα 3- 6. Εικόνα 3- 6: Δομή ριζικού δέντρου στο Linux



Εικόνα 3- 6: Δομή ριζικού δέντρου στο Linux

Βέβαια η Εικόνα 3- 6 παρουσιάζει μερικές αποκλίσεις ανάλογα με τη διανομή του Linux που εξετάζεται. Σαν αρχικός κατάλογος του συστήματος αρχείων καθορίζεται ο κατάλογος root (/) κάτω από τον οποίο υπάρχουν οι παρακάτω κατάλογοι με τα αντίστοιχα περιεχόμενα (FedoraDocs-FileSystem, n.d.):

- **/bin** οι βασικές εντολές του Linux
- **/boot** Τα αρχεία που χρησιμοποιούνται στην εκκίνηση π.χ kernel.
- **/dev** Ειδικά αρχεία που περιγράφουν περιφερειακές συσκευές πχ δίσκοι.
- **/etc** Αρχεία ρυθμίσεων διαφόρων λειτουργιών του Linux.
- **/home** Κατάλογοι με τα αρχεία των χρηστών.
- **/lib** Βιβλιοθήκες, διαμοιραζόμενες βιβλιοθήκες και μέρη του C compiler.
- **/media** Ειδικά αρχεία που περιγράφουν τις αφαιρούμενες συσκευές π.χ. usb disks.
- **/mnt** Τα συστήματα αρχείων των αφαιρούμενων συσκευών
- **/opt** Προγράμματα τρίτων κατασκευαστών
- **/proc** Εικονικό σύστημα αρχείων που περιέχει πληροφορίες σχετικές με όλες τις διεργασίες που εκτελούνται κάθε στιγμή στο σύστημα
- **/root** Κατάλογος που περιέχει τα αρχεία του λογαριασμού root
- **/sbin** Εντολές για τη διαχείριση του συστήματος
- **/tmp** Προσωρινά αρχεία που διαγράφονται κάθε φορά που γίνεται επανεκκίνηση του συστήματος
- **/usr** Προγράμματα, βιβλιοθήκες, σελίδες βοήθειας για όλα τα προγράμματα που αναφέρονται στους χρήστες
- **/var** Εδώ αποθηκεύονται όλα τα αρχεία που δημιουργούνται από τους χρήστες όπως αρχεία καταγραφής, τα αρχεία του email, η «ουρά» του εκτυπωτή

Το λειτουργικό σύστημα Linux μπορεί να υποστηρίξει μια μεγάλη γκάμα αρχείων. Συγκεκριμένα, με την εκτέλεση της εντολής **ls -ld** ο πρώτος χαρακτήρας που εμφανίζεται περιγράφει και το είδος του αρχείου. Ο Πίνακας 3- 9 περιγράφει τα είδη των αρχείων μαζί με τον χαρακτήρα που τα περιγράφει καθώς και την εντολή που τα δημιουργεί.

| Τύπος αρχείου   | Χαρακτήρας | Δημιουργείται από |
|-----------------|------------|-------------------|
| Κανονικό αρχείο | -          | Επεξεργαστές κ.α. |
| Κατάλογος       | d          | mkdir             |
| Αρχείο συσκευής | c          | mknod             |

|                       |          |        |
|-----------------------|----------|--------|
| χαρακτήρα             |          |        |
| Αρχείο συσκευής block | <b>b</b> | mknod  |
| Socket                | <b>s</b> | socket |
| Διασωλήνωση           | <b>p</b> | mknod  |
| Συμβολικό link        | <b>l</b> | ln -s  |

Πίνακας 3- 9: Τύποι αρχείων στο Linux

### 3.3.4 Προσάρτηση συστήματος αρχείων

Ο βασικός τρόπος με τον οποίο γίνεται αποθήκευση των δεδομένων σε ένα υπολογιστικό σύστημα είναι με τη χρήση ενός σκληρού δίσκου. Οι τεχνολογίες δίσκων που υπάρχουν αυτή τη στιγμή είναι οι **Hard Disk Drives (HDD)** και οι **Solid State Drives (SSD)**. Οι SSD παρέχουν μεγαλύτερες ταχύτητες στην ανάγνωση και στην εγγραφή των δεδομένων από την άλλη όμως είναι πιο ακριβοί μέχρι σήμερα.

Το λειτουργικό σύστημα Linux χειρίζεται τους HDD και SSD με τον ίδιο τρόπο. Ένα χαρακτηριστικό των σκληρών δίσκων που ενδιαφέρει το λειτουργικό σύστημα είναι ο τρόπος με τον οποίο ο σκληρός δίσκος συνδέεται στη μητρική πλακέτα. Μερικοί από τους τρόπους σύνδεσης των σκληρών δίσκων είναι οι παρακάτω (Wikipedia-HardDisk, n.d.):

- **Parallel Advanced Technology Attachment (PATA)**. Χρησιμοποιεί πλατύ καλώδιο και επιτρέπει τη σύνδεση 2 συσκευών σε ένα προσαρμογέα. Χρησιμοποιείται σε παλιούς υπολογιστές.
- **Serial Advanced Technology Attachment (SATA)**. Χρησιμοποιεί στενό καλώδιο και επιτρέπει τη σύνδεση 4 συσκευών σε έναν προσαρμογέα. Είναι πιο γρήγορο από το PATA και είναι η σύνδεση που χρησιμοποιείται στους περισσότερους υπολογιστές σήμερα.
- **Small Computer System Interface (SCSI)**. Χρησιμοποιεί πλατύ καλώδιο αλλά είναι αρκετά γρήγορο. Επιτρέπει τη σύνδεση μέχρι και 8 συσκευών. Χρησιμοποιείται πιο σπάνια σήμερα.
- **Universal Serial Bus (USB)**. Επιτρέπει τη σύνδεση εξωτερικών δίσκων<sup>11</sup>.

Με τη σύνδεση ενός σκληρού δίσκου στον υπολογιστή ο Linux kernel δημιουργεί ένα αρχείο που είναι συνδεδεμένο με αυτή τη συσκευή στον κατάλογο **/dev**. Το αρχείο αυτό είναι τύπου block συσκευής και παρέχει μια σύνδεση του Linux με τη συσκευή αυτή. Η ονομασία αυτού του αρχείου εξαρτάται από τον τρόπο σύνδεσης του σκληρού δίσκου. Για τους δίσκους που συνδέονται με PATA το αρχείο αυτό έχει το όνομα **/dev/hdx** όπου **x** είναι ένα γράμμα που περιγράφει το συγκεκριμένο δίσκο ξεκινώντας από το **a** για τον πρώτο δίσκο. Για τους δίσκους που συνδέονται με SATA και SCSI το αρχείο αυτό έχει όνομα **/dev/sdx**. Αν το σύστημα μας διαθέτει και συσκευή ανάγνωσης CD/DVD αυτή έχει το όνομα **/dev/srx**. Για παράδειγμα ο πρώτος σκληρός δίσκος SATA στο σύστημα μας έχει το όνομα **/dev/sda** ο δεύτερος **/dev/sdb** κτλ. Στην παρακάτω εικόνα εμφανίζονται οι δίσκοι που διαθέτει το σύστημα μας με τη χρήση της εντολής **lsblk**.

<sup>11</sup> Η χρήση τους δεν θα εξεταστεί.

```
[user@ekdda ~]$ lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                                  8:0    0   20G  0 disk
├─sda1                               8:1    0    1G  0 part /boot
└─sda2                               8:2    0   19G  0 part
   ├─fedora_localhost--live-root
   │                               253:0    0   17G  0 lvm  /
   └─fedora_localhost--live-swap
       253:1    0    2G  0 lvm  [SWAP]
sr0                                  11:0    1   1.8G  0 rom
```

### Διαχείριση κατατμήσεων (partitions)

Τα περισσότερα λειτουργικά συστήματα, όπως και το Linux, επιτρέπει τον διαμοιρασμό ενός φυσικού σκληρού δίσκου σε πολλές κατατμήσεις. Μία κατάτμηση (**partition**) είναι ένα τμήμα ενός σκληρού δίσκου το οποίο το λειτουργικό σύστημα τον αντιμετωπίζει σαν ξεχωριστό χώρο αποθήκευσης. Με τη χρήση των κατατμήσεων είναι δυνατή η καλύτερη οργάνωση των δεδομένων καθώς και η μεγαλύτερη ασφάλεια. Για παράδειγμα σε περίπτωση που ένας κακόβουλος χρήστης γεμίσει με άχρηστα δεδομένα μία κατάτμηση το λειτουργικό σύστημα θα έχει τις υπόλοιπες για να λειτουργήσει.

Όπως έχει ήδη αναφερθεί υπάρχουν δύο τρόποι καταγραφής των κατατμήσεων στον σκληρό δίσκο. Ο πρώτος ονομάζεται **MBR** (Master Boot Record) και υποστηρίζει δίσκους χωρητικότητας μέχρι 2 Terrabytes και 4 primary partitions. Κάθε ένα από τα primary partitions μπορεί να χωριστεί σε περισσότερα extended. Ο δεύτερος ονομάζεται **GUID Partition Table** (GPT) και μπορεί να υποστηρίξει μέχρι 128 partitions.

Η διαχείριση των κατατμήσεων στο Linux μπορεί να γίνει με μια ομάδα προγραμμάτων όπως τα **fdisk**, **gdisk** και **GNU parted** και απαιτεί δικαιώματα διαχειριστή. Στη συνέχεια θα αναφερθούμε στη λειτουργία του προγράμματος GNU parted (Clausen, Kreuter, & Polzer, 2019). Η εκτέλεση της εντολής **parted** έχει σαν αποτέλεσμα την εμφάνιση της παρακάτω οθόνης

```
[root@ekdda ~]# parted
GNU Parted 3.2.153
Using /dev/sda
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted)
```

Μερικές από τις εντολές που διαθέτει η parted μαζί με τη λειτουργικότητα τους εμφανίζονται στον Πίνακα 3- 10.

| Εντολή   | Λειτουργικότητα                           |
|--|---|
| <b>mklabel</b> τύπος                             | Δημιουργεί ένα καινούργιο partition table |
| <b>mkpart</b> τύπος [τύπος FS] <b>ΑΡΧΗ ΤΕΛΟΣ</b> | Δημιουργεί ένα καινούργιο partition       |

|  |  |
|--|--|
| <b><i>print</i> [devices   free   all   all partition]</b> | Εμφανίζει πληροφορίες σχετικές με συσκευές, partitions |
| <b><i>resizepart</i> ΑΡΙΘΜΟΣ ΤΕΛΟΣ</b>                     | Μεταβάλλει το μέγεθος ενός partition                   |
| <b><i>select</i> ΣΥΣΚΕΥΗ</b>                               | Επιλέγει ένα συγκεκριμένο δίσκο / partition            |
| <b><i>unit</i> ΜΟΝΑΔΑ</b>                                  | Αλλάζει τις μονάδες μεγέθους KB, MB, GB                |
| <b><i>quit</i></b>   | Τερματίζει τη λειτουργία του parted                    |
| <b><i>rm</i> ΑΡΙΘΜΟΣ</b>                                   | Διαγράφει ένα partition                                |

Πίνακας 3- 10: Εντολές του εργαλείου parted

Το λειτουργικό σύστημα Linux δημιουργεί στον κατάλογο **/dev** ένα αρχείο για κάθε κατάτμηση. Η ονομασία αυτού του αρχείου είναι η ονομασία του σκληρού δίσκου που ανήκει και στη συνέχεια ένας αριθμός που καθορίζει τον αύξοντα αριθμό της κατάτμησης. Για παράδειγμα στο παρακάτω πλαίσιο χρησιμοποιούμε την εντολή **print all** της parted για να εμφανίσουμε την πληροφορία όλων των δίσκων. Με αυτόν τον τρόπο ανακαλύπτουμε ότι ο σκληρός δίσκος **/dev/sdb** χωρίζεται σε δύο κατατμήσεις τις **/dev/sdb1** και **/dev/sdb2**.

```
(parted) print all
Model: VMware, VMware Virtual S (scsi)
Disk /dev/sda: 21.5GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type    File system  Flags
  1      1049kB  1075MB  1074MB  primary ext4         boot
  2      1075MB  21.5GB  20.4GB  primary                lvm
```

Στη συνέχεια θα παρουσιάσουμε τις ενέργειες που χρειάζεται να κάνουμε για να προσθέσουμε και να χρησιμοποιήσουμε έναν καινούργιο σκληρό δίσκο. Η πρώτη ενέργεια που πρέπει να εκτελέσουμε είναι να διαπιστώσουμε αν το σύστημα μας έχει αναγνωρίσει τον σκληρό δίσκο και ποιο γράμμα του έχει αναθέσει. Αυτό μπορούμε να το καταφέρουμε με τη χρήση της εντολής **lsblk** όπως στο παρακάτω πλαίσιο.

```
[root@ekdda ~]# lsblk
NAME                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                  8:0    0   20G  0 disk
## παρακάπτουμε αποτελέσματα για οικονομία χώρου...
##
sdb                  8:16    0    5G  0 disk
...*
```

Όπως βλέπουμε πράγματι το σύστημα μας έχει αναγνωρίσει τον δίσκο και του έχει αποδώσει το όνομα **/dev/sdb**. Για τον συγκεκριμένο δίσκο επομένως γνωρίζουμε ότι είναι ή SATA ή SCSI και έχει μέγεθος **5GB**.



Θα προσπαθήσουμε να διαμορφώσουμε τον δίσκο αυτό σε δύο κατατμήσεις την πρώτη με μέγεθος 3GB και τη δεύτερη 2GB. Στο παρακάτω πλαίσιο εκτελούμε την εντολή `parted` με δικαιώματα διαχειριστή και στην συνέχεια εκτελούμε την `print all` για να δούμε τις πληροφορίες για το καινούργιο δίσκο.

```
(parted) print all
## παρακάπτουμε αποτελέσματα για οικονομία χώρου...
##
Error: /dev/sdb: unrecognised disk label
Model: ATA VMware Virtual S (scsi)
Disk /dev/sdb: 5369MB
Sector size (logical/physical): 512B/512B
Partition Table: unknown
Disk Flags:
```

Βλέπουμε ότι η εντολή `print all` εμφανίζει μήνυμα λάθους για τον συγκεκριμένο δίσκο καθώς δεν έχει δημιουργηθεί πίνακας κατατμήσεων σε αυτόν. Επομένως πρέπει να δημιουργήσουμε πίνακα κατατμήσεων στον συγκεκριμένο δίσκο. Αυτό θα γίνει με τη χρήση των εντολών `select /dev/sdb` για να πούμε ότι θέλουμε να επεξεργαστούμε τον δεύτερο δίσκο και `mklabel msdos` γιατί θέλουμε να τον διαμορφώσουμε με MBR όπως στο παρακάτω πλαίσιο.

```
(parted) select /dev/sdb
Using /dev/sdb
(parted) mklabel msdos
(parted) print all
## παρακάπτουμε αποτελέσματα για οικονομία χώρου...
##
Model: ATA VMware Virtual S (scsi)
Disk /dev/sdb: 5369MB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start  End  Size  Type  File system  Flags
```

Επόμενο βήμα είναι η δημιουργία των δύο κατατμήσεων. Αυτό θα γίνει με τη χρήση της εντολής `mkpart`.

```
(parted) mkpart
Partition type?  primary/extended? primary
File system type?  [ext2]? ext4
Start? 0
End? 3GB
Warning: The resulting partition is not properly aligned for best
performance:
1s % 2048s != 0s
Ignore/Cancel? i
(parted) mkpart
Partition type?  primary/extended? primary
File system type?  [ext2]? ext4
Start? 3001MB
End? 5369MB
(parted)
```

Αποτέλεσμα των παραπάνω εντολών είναι ο επιτυχής χωρισμός του σκληρού δίσκου /dev/sdb σε δύο κατατμήσεις όπως φαίνεται και παρακάτω.

```
Model: ATA VMware Virtual S (scsi)
Disk /dev/sdb: 5369MB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type    File system  Flags
  1      512B   3000MB  3000MB  primary
  2      3001MB 5369MB  2368MB  primary
```

### Μορφοποίηση partitions

Μετά τον διαχωρισμό ενός δίσκου σε κατατμήσεις ακολουθεί η διαμόρφωσή τους με ένα σύστημα αρχείων (filesystem). Με τον όρο σύστημα αρχείων περιγράφουμε τον τρόπο με τον οποίο είναι αποθηκευμένα τα δεδομένα σε μία κατάτμηση. Υπάρχουν πολλά συστήματα αρχείων με το κάθε ένα από αυτά να παρουσιάζει συγκεκριμένα πλεονεκτήματα και τα μειονεκτήματα (Salter, 2018). Παρακάτω αναφέρονται μερικά από τα βασικά συστήματα αρχείων που θα αντιμετωπίσουμε σαν διαχειριστές ενός συστήματος:

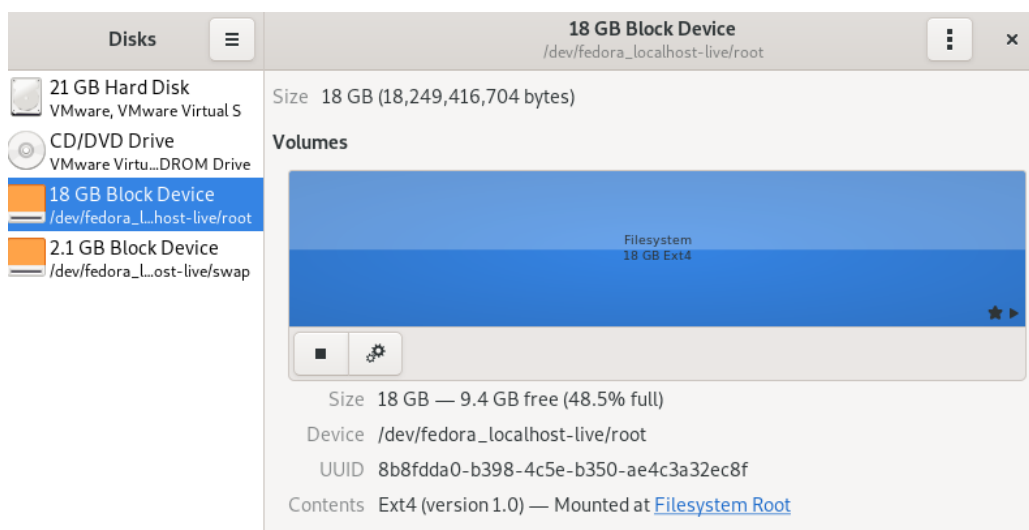
- **B-Tree File System (btrfs)**. Είναι ένα σύγχρονο σύστημα αρχείων το οποίο μπορεί να υποστηρίξει μεγάλο μέγεθος αρχείων καθώς και να υποστηρίξει τις τεχνολογίες Redundant Array of Inexpensive Disks (**RAID**) και Logical Volume Manager (**LVM**) καθώς και την τεχνολογία των **Subvolumes**.
- **Extended File System (ext3, ext4)**. Είναι απόγονοι του αρχικού συστήματος αρχείων του Linux (ext) υποστηρίζουν το journaling καθώς και άλλες προηγμένες τεχνολογίες απόδοσης.

- **swap**. Είναι ένα ειδικό σύστημα αρχείων το οποίο στην πραγματικότητα δημιουργεί εικονική μνήμη στον υπολογιστή χρησιμοποιώντας χώρο από έναν σκληρό δίσκο. Το σύστημα μπορεί στη συνέχεια να αντιμετωπίσει δεδομένα από τη μνήμη στο swap με αποτέλεσμα την πρόσθεση μνήμης στο σύστημα. Εδώ δεν μπορούμε να αποθηκεύσουμε δεδομένα μόνιμα.
- **New Technology File System (NTFS)**. Το NTFS είναι ένα σύστημα αρχείων που δημιούργησε η εταιρία Microsoft και χρησιμοποιείται στα λειτουργικά συστήματα Windows. Το λειτουργικό σύστημα Linux μπορεί να διαβάσει και να γράψει στο NTFS από την έκδοση kernel 2.6 και μετά.
- **Virtual File Allocation Table (VFAT)**. Το VFAT είναι η επέκταση του FAT που έχουν δημιουργηθεί από την εταιρία Microsoft. Δεν χρησιμοποιείται τόσο συχνά στους σκληρούς δίσκους που υπάρχουν στον υπολογιστή αλλά χρησιμοποιείται αρκετά στις φορητές μονάδες αποθήκευσης όπως τα USB flash disks.



Η διανομή Fedora από την έκδοση 33 χρησιμοποιεί σαν προκαθορισμένο σύστημα αρχείων το btrfs (Murphy, 2020).

Για την παρουσίαση των κατατμήσεων και των συστημάτων αρχείων τους μπορούμε να χρησιμοποιήσουμε το γραφικό εργαλείο **Disks**. Η εκκίνηση του γίνεται αν επιλέξουμε τα Activities και στη συνέχεια γράψουμε Disks στη γραμμή αναζήτησης και από τα αποτελέσματα επιλέγουμε το αντίστοιχο εικονίδιο. Αποτέλεσμα θα είναι η εκκίνηση του εργαλείου Disks όπως παρουσιάζεται στην Εικόνα 3- 7. Στη συνέχεια για να εμφανίσουμε πληροφορίες για κάποια συγκεκριμένη κατάτμηση πρέπει να το επιλέξουμε από τα αριστερά και στο κέντρο του παραθύρου εμφανίζονται οι πληροφορίες του συστήματος αρχείων για τη συγκεκριμένη κατάτμηση. Στην Εικόνα 3- 7 εμφανίζεται το η κατάτμηση **/fedora-localhost-line/root** η οποία είναι ένα λογικό partition<sup>12</sup> έχει μέγεθος 18GB και είναι γεμάτο κατά 48,5%. Ακόμα το σύστημα αρχείων σε αυτή την κατάτμηση είναι το ext4.



Εικόνα 3- 7: Το γραφικό εργαλείο Disks

<sup>12</sup> Περισσότερα για τα λογικά partition στην συνέχεια

Για να εμφανίσουμε πληροφορίες σχετικές με τα συστήματα αρχείων μέσα από τη γραμμή εντολής μπορούμε να χρησιμοποιήσουμε είτε την εντολή **mount** χωρίς ορίσματα είτε την εντολή **df -T**. Επειδή και οι δύο θα μας εμφανίσουν πληροφορίες για όλα τα partition μπορούμε να τις χρησιμοποιήσουμε σε συνδυασμό με την **grep** για να εμφανίσουμε πληροφορίες μόνο για αυτό που μας ενδιαφέρει.

```
[root@ekdda ~]# mount | grep /dev/sda1
/dev/sda1 on /boot type ext4 (rw,relatime,seclabel)
[root@ekdda ~]# df -T |grep /dev/sda1
/dev/sda1          ext4          999320 173656    756852 19% /boot
```

Η δημιουργία ενός συστήματος αρχείων σε μία κατάτμηση γίνεται με τη χρήση της οικογένειας εντολών **mkfs** (Lonston, 2018). Για κάθε σύστημα αρχείων που θέλουμε να δημιουργήσουμε υπάρχει και η αντίστοιχη εντολή **mkfs.x**. Η σύνταξη της εντολής είναι

***mkfs."filesystem" partition\_name***

Το παρακάτω πλαίσιο παρουσιάζει τα συστήματα αρχείων που μπορεί να υποστηρίξει το σύστημα μας.

```
[root@ekdda ~]# ls /usr/sbin/mkfs.*
/usr/sbin/mkfs.btrfs    /usr/sbin/mkfs.ext4    /usr/sbin/mkfs.msdos
/usr/sbin/mkfs.cramfs   /usr/sbin/mkfs.fat     /usr/sbin/mkfs.ntfs
/usr/sbin/mkfs.ext2     /usr/sbin/mkfs.hfsplus /usr/sbin/mkfs.vfat
/usr/sbin/mkfs.ext3     /usr/sbin/mkfs.minix   /usr/sbin/mkfs.xfs
```

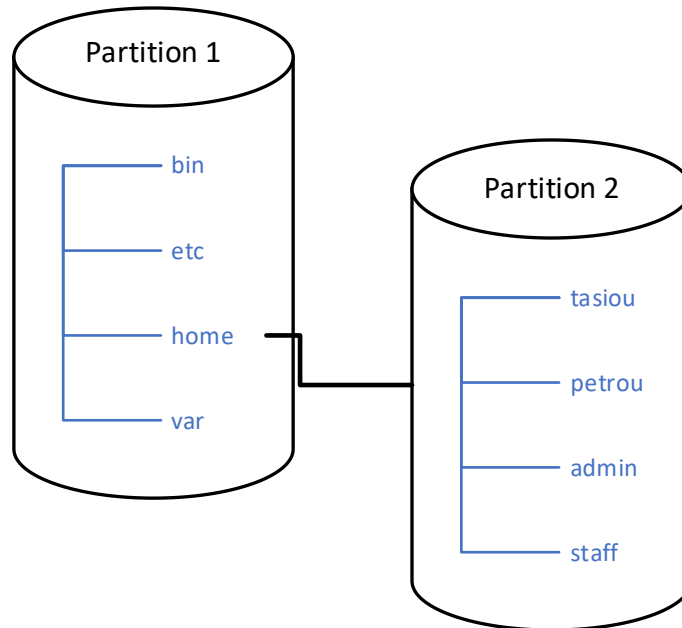
Στον παράδειγμα μας ας πούμε ότι θέλουμε να διαμορφώσουμε την πρώτη κατάτμηση του δίσκου που προσθέσαμε με το σύστημα αρχείων **ext4**. Η εντολή που θα το κάνει αυτό παρουσιάζεται στο παρακάτω πλαίσιο.

```
[root@ekdda ~]# mkfs.ext4 /dev/sdb1
mke2fs 1.45.3 (14-Jul-2019)
Creating filesystem with 732421 4k blocks and 183264 inodes
Filesystem UUID: 7d823d9a-f44c-44bf-a377-3eafa2ebc276
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done
```

### Προσάρτηση συστήματος αρχείων

Τελευταίο στάδιο πριν τη χρήση μίας κατάτμησης είναι η εισαγωγή της στο σύστημα αρχείων του Linux. Αυτό γίνεται με την τεχνική της προσάρτησης (**mount**) (Lonston, 2018). Συγκεκριμένα, σε κάθε σύστημα Linux υπάρχει μία κατάτμηση που περιέχει το root σύστημα αρχείων (/) και στο οποίο οι υπόλοιπες κατατμήσεις προσαρτούν τα δικά τους συστήματα αρχείων σε έναν κατάλογο που ονομάζεται **mounting point**. Για παράδειγμα στην Εικόνα 3- 8 παρατηρούμε ότι η δεύτερη κατάτμηση έχει σαν mounting point τον κατάλογο **/home**.



Εικόνα 3- 8: Προσάρτηση δίσκων

Η εντολή που δείχνει ποια είναι τα αποθηκευτικά μέσα και σε ποιο είναι το mounting point στο καθένα από αυτά είναι η `df -h` και η εκτέλεση της στο σύστημα μας παρουσιάζεται στο παρακάτω πλαίσιο.

```
[root@ekdda ~]# df -h
Filesystem                Size      Used Avail Use% Mounted on
devtmpfs                  1.9G         0  1.9G   0% /dev
tmpfs                     2.0G         0  2.0G   0% /dev/shm
tmpfs                     2.0G   1.8M  2.0G   1% /run
/dev/mapper/fedora_localhost--live-root 17G   7.9G  7.9G  51% /
tmpfs                     2.0G    80K  2.0G   1% /tmp
/dev/sda1                 976M   170M  740M  19% /boot
tmpfs                     391M    11M  381M   3% /run/user
/dev/sr0                  1.8G   1.8G     0 100%
/run/media/user/Fedora-WS-Live-31-1-9
```

Στο παραπάνω πλαίσιο τα **tmpfs** είναι προσωρινά συστήματα αρχείων που δημιουργεί το Linux και το `/dev/mapper` είναι ένα λογικό partition που θα παρουσιάσουμε στη συνέχεια.

Θα πρέπει λοιπόν να προσαρτήσουμε το πρώτο από τα partitions που δημιουργήσαμε. Η προσάρτηση γίνεται με τη χρήση της εντολής `mount` με την παρακάτω σύνταξη:

```
mount -t τύπος_συσ_αρχείων συσκευή mountpoint
```

Επομένως μπορούμε να προσαρτήσουμε το πρώτο partition `/dev/sdb1` στον κατάλογο `/home/data` με την παρακάτω εντολή. Προσοχή, ο κατάλογος `data` θα πρέπει να υπάρχει πριν την προσάρτηση.

```
[root@ekdda ~]# mkdir /home/data
[root@ekdda ~]# mount -t ext4 /dev/sdb1 /home/data
[root@ekdda ~]# df -h | grep /dev/sdb1
/dev/sdb1                2.7G  8.4M  2.6G   1% /home/data
```

Για να διαπιστώσουμε ότι πράγματι ο κατάλογος `/home/data` ανήκει σε διαφορετικό partition μπορούμε να δημιουργήσουμε ένα «ψεύτικο» αρχείο μεγάλου μεγέθους στον κατάλογο αυτό, με την εντολή `fallocate`, και να δούμε αν έχει αυξηθεί το ποσοστό χρήσης του αντίστοιχου partition.

```
[root@ekdda ~]# fallocate -l 2000MB /home/data/test.file
[root@ekdda ~]# df -h | grep /dev/sdb1
/dev/sdb1                2.7G  1.9G  678M  74% /home/data
```

Η προσάρτηση ενός partition σε ένα κατάλογο που γίνεται με την εντολή `mount` είναι διαθέσιμη όσο λειτουργεί το σύστημα μας. Αν το σύστημα μας επανεκκινήσει, η προσάρτηση αυτή έχει χαθεί. Λύση σε αυτό το πρόβλημα δίνει το αρχείο `/etc/fstab` που περιέχει όλες τις προσαρτήσεις που κάνει το σύστημα μας αυτόματα, όταν ξεκινά. Επομένως θα πρέπει να προσθέσουμε, με τη χρήση ενός επεξεργαστή κειμένου, την παρακάτω γραμμή στο `/etc/fstab` αρχείο (Carrigan, 2020).

```
/dev/sda1 /home/data                ext4 defaults 1 2
```

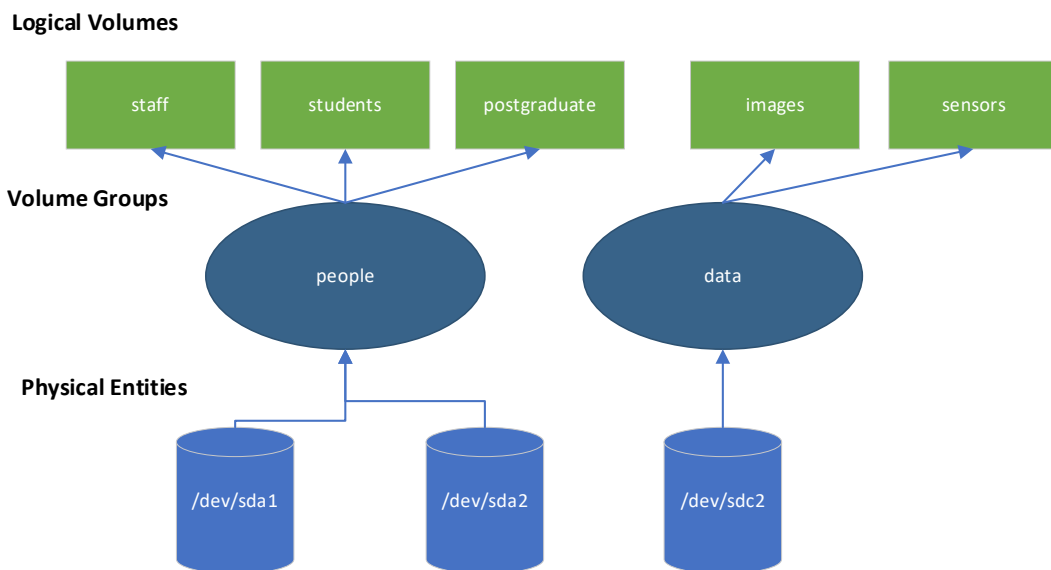
Και στη συνέχεια να εκτελέσουμε την εντολή `systemctl daemon-reload` για να ενημερωθεί η διεργασία `systemd` που κάνει την προσάρτηση των partitions κατά την εκκίνηση του συστήματος.

### Logical Volume Manager (LVM)

Ένα από τα προβλήματα που δημιουργούνται από τη χρήση των κατατμήσεων είναι ότι το μέγεθός τους είναι στατικό και σε περίπτωση που θέλουμε να μεγαλώσουμε το μέγεθος επειδή έχει γεμίσει ο φυσικός δίσκος, υπάρχει μεγάλο διαχειριστικό κόστος. Η λύση σε αυτό το πρόβλημα έρχεται με τη χρήση του **Logical Volume Manager (LVM)** (RedHat-LVM, n.d.).

Το LVM ορίζει τις παρακάτω καινούργιες οντότητες όπως αυτές παρουσιάζονται στην Εικόνα 3- 9:

- **Physical Volumes (PV):** Κάθε PV είναι ένας φυσικός δίσκος. Μπορεί να είναι είτε ολόκληρος ο δίσκος είτε μία κατάτμηση είτε ακόμα και ένα σύστημα RAID. Για να μπορεί μία κατάτμηση να συμμετέχει σε LVM πρέπει να δηλωθεί στο partition table σαν Linux LVM Filesystem.
- **Logical Volumes (LV):** Είναι ένα λογικό ισοδύναμο μιας κανονικής κατάτμησης για το λειτουργικό σύστημα. Στην πραγματικότητα μπορεί να αντιστοιχεί σε τμήματα από παραπάνω από μια κανονικές κατατμήσεις σε φυσικούς δίσκους.
- **Volume Groups (VG):** Είναι μια μορφή οργάνωσης όπου «ομαδοποιεί» πολλά PV και LV σε μία οργανωτική μονάδα. Για να μπορεί να αναθέτει ένα PV σε πολλά LV χωρίζει το κάθε PV σε πολλά μικρά Physical Entities PE



Εικόνα 3- 9: Logical Volume Manager

Υπάρχουν αρκετές εντολές που σχετίζονται με τη λειτουργία και τη διαχείριση του LVM (RedHat-LVM, n.d.) και αυτές παρουσιάζονται οργανωμένα στον Πίνακα 3- 11.

| Λειτουργία      | Physical Volume (PV)    | Volume Group (VG)       | Logical Volume (LV)     |
|-----------------|-------------------------|-------------------------|-------------------------|
| Εμφάνιση        | <i><b>pvs</b></i>       | <i><b>vgs</b></i>       | <i><b>lvs</b></i>       |
| Εμφάνιση (αναλ) | <i><b>pvdisplay</b></i> | <i><b>vgdisplay</b></i> | <i><b>lvdisplay</b></i> |
| Δημιουργία      | <i><b>pvcreate</b></i>  | <i><b>vgcreate</b></i>  | <i><b>lvcreate</b></i>  |
| Επέκταση        |                         | <i><b>vgextend</b></i>  | <i><b>lvextend</b></i>  |
| Μείωση          |                         | <i><b>vgreduce</b></i>  | <i><b>lvreduce</b></i>  |
| Διαγραφή        | <i><b>pvremove</b></i>  | <i><b>vgremove</b></i>  | <i><b>lvremove</b></i>  |
| Μετονομασία     |                         |                         | <i><b>lvrename</b></i>  |
| Μετακίνηση      | <i><b>pvmove</b></i>    |                         |                         |

Πίνακας 3- 11: Εντολές διαχείρισης του LVM

Για παράδειγμα αν θέλουμε να εμφανίσουμε σύντομες πληροφορίες για τα physical volumes, τα volume groups, και τα logical volumes που υπάρχουν στο σύστημα μας μπορούμε να χρησιμοποιήσουμε τις παρακάτω εντολές.

```
[root@ekdda ~]# pvs
PV          VG          Fmt Attr PSize  PFree
 /dev/sda2  fedora_localhost-live lvm2 a-- <19.00g  0
[root@ekdda ~]# vgs
VG          #PV #LV #SN Attr   VSize  VFree
 fedora_localhost-live  1   2   0 wz--n- <19.00g  0
[root@ekdda ~]# lvs
LV  VG          Attr      LSize  Pool Origin Data%
Meta% Move Log Cpy%Sync Convert
root fedora_localhost-live -wi-ao---- <17.00g
swap fedora_localhost-live -wi-ao----  2.00g data
```

Παρατηρούμε δηλαδή ότι έχουμε ένα physical volume το **/dev/sda2** που ανήκει σε ένα volume group (**fedora\_localhost-live**) και έχουμε δύο logical volumes το **root** και το **swap**. Αυτά είναι και τα προεπιλεγμένα partitions που δημιουργεί η Fedora κατά την εγκατάσταση της.

Στη συνέχεια θα παρουσιάσουμε ένα παράδειγμα χρήσης του LVM. Αρχικά θα δημιουργήσουμε ένα volume group. Σε αυτό το volume group θα αναθέσουμε ένα physical volume και τέλος μέσα σε αυτό το volume group θα δημιουργήσουμε και ένα logical volume. Θα χρησιμοποιήσουμε την κατάτμηση **/dev/sdb2** την οποία είχαμε δημιουργήσει σε προηγούμενο παράδειγμα και θα το ενημερώσουμε ότι θα χρησιμοποιηθεί σε LVM με τη χρήση της παρακάτω εντολής στο εργαλείο parted.

```
(parted) select /dev/sdb
Using /dev/sdb
(parted) set 2 lvm on
```

Στη συνέχεια αρχικά δημιουργούμε ένα καινούργιο physical volume από την κατάτμηση **/dev/sdb2**. Μετά δημιουργούμε ένα καινούργιο volume group με το όνομα **data** στο οποίο αναθέτουμε το physical volume που δημιουργήσαμε. Και τέλος δημιουργούμε ένα καινούργιο logical volume με το όνομα **images** που ανήκει στο volume group **data**. Οι ενέργειες μας αυτές παρουσιάζονται στο παρακάτω πλαίσιο.

```
[root@ekdda ~]# pvcreate /dev/sdb2
Physical volume "/dev/sdb2" successfully created.
[root@ekdda ~]# vgcreate data /dev/sdb2
Volume group "data" successfully created
[root@ekdda ~]# lvcreate -n images -L 2GB data
Logical volume "images" created.
```



Πλέον έχουμε ένα καινούργιο logical volume με το όνομα **data-images** που το χειριζόμαστε μέσα από το LVM. Αυτό όμως για το λειτουργικό σύστημα είναι σαν μία κανονική κατάτμηση που μπορούμε να διαμορφώσουμε με ό,τι σύστημα αρχείων θέλουμε και να την προσαρτήσουμε σε έναν κατάλογο του συστήματος. Η μεγάλη διαχειριστική διαφορά είναι ότι σε αυτή την κατάτμηση μπορούμε να αυξήσουμε πολύ εύκολα το μέγεθος αναθέτοντας καινούριο αποθηκευτικό χώρο από το volume group.

### 3.3.5 Διαχείριση δικαιωμάτων αρχείων / καταλόγων

Κάθε αρχείο και κατάλογος στο Linux έχει δικαιώματα ανάγνωσης (read), εγγραφής (write) και εκτέλεσης (execute) για τρεις κατηγορίες χρηστών οι οποίες είναι:

- **Ιδιοκτήτης (owner):** είναι τα δικαιώματα που έχει στο αρχείο / κατάλογο ο ιδιοκτήτης του (όπως φαίνεται από την εντολή ls)
- **Ομάδα ιδιοκτήτη (group owner):** είναι τα δικαιώματα που έχουν στο αρχείο / κατάλογο τα μέλη που ανήκουν στο group στο οποίο ανήκει το αρχείο / κατάλογος (όπως φαίνεται από την εντολή ls)
- **Όλοι οι υπόλοιποι (all users):** είναι τα δικαιώματα που έχουν όλοι οι υπόλοιποι χρήστες

Τα δικαιώματα πρόσβασης για κάθε αρχείο / κατάλογο εμφανίζονται με τη χρήση της παραμέτρου `-l` στην εντολή `ls`. Είναι οι πρώτοι εννέα χαρακτήρες που εμφανίζονται μετά τον πρώτο, ο οποίος παρουσιάζει και τον τύπο του αρχείου όπως καταγράφηκε και παρουσιάστηκε στην Ενότητα 3.3.3. Οι χαρακτήρες αυτοί έχουν πάντα σταθερά σειρά εμφάνισης και είναι με τη σειρά τα δικαιώματα ανάγνωσης εγγραφής, εκτέλεσης για τον ιδιοκτήτη, το ομάδα ιδιοκτήτη του ιδιοκτήτη και τους υπόλοιπους χρήστες.

```
[user@ekdda ~]$ ls -l testfile
-rw-r--r--. 1 user user 0 Dec  3 16:02 testfile
```

Παρατηρούμε ότι το αρχείο **testfile** έχει ιδιοκτήτη τον χρήστη **user** και ομάδα ιδιοκτήτη το user private group του χρήστη user. Τα δικαιώματα που έχει ο user είναι εγγραφή και ανάγνωση ενώ όλοι οι υπόλοιποι έχουν μόνο ανάγνωση.

Η αλλαγή των δικαιωμάτων σε ένα αρχείο / κατάλογο μπορεί να γίνει με τη χρήση της εντολής **chmod**. Η εντολή chmod μπορεί να χρησιμοποιηθεί είτε με την αριθμητική είτε με τη λεκτική μορφή της.

Στη λεκτική της μορφή η εντολή chmod παίρνει σαν παραμέτρους τις παρακάτω:

- Έναν ή περισσότερους χαρακτήρες που καθορίζουν σε ποια από τις κατηγορίες δικαιωμάτων θα γίνει η αλλαγή (**u**=ιδιοκτήτη, **g**=ομάδα ιδιοκτήτη, **o**=υπόλοιπους, **a**=σε όλες τις κατηγορίες). Αν παραληφθεί ο χαρακτήρας που καθορίζει την κατηγορία τότε εφαρμόζεται σε όλους (**a**).
- Τι είδους αλλαγή πρέπει να γίνει (+ προσθήκη δικαιωμάτων, - αφαίρεση δικαιωμάτων, = ανάθεση δικαιωμάτων)
- Ποιο από τα δικαιώματα θα μεταβληθούν (**r**= ανάγνωση, **w**=εγγραφή, **x**=εκτέλεση)

Για παράδειγμα η εντολή που προσθέτει το δικαίωμα εκτέλεσης στον ιδιοκτήτη και στην ομάδα ιδιοκτήτη για το αρχείο **testfile** παρουσιάζεται στο παρακάτω πλαίσιο.

```
[user@ekdda ~]$ chmod ug+x testfile
[user@ekdda ~]$ ls -l testfile
-rwxr-xr--. 1 user user 0 Dec  3 16:02 testfile
```

Με τη χρήση της αριθμητικής μορφής της `chmod` μπορούμε μόνο να αναθέσουμε δικαιώματα, δεν μπορούμε να προσθέσουμε ή να αφαιρέσουμε από τα υπάρχοντα. Συγκεκριμένα αντιστοιχούμε έναν αριθμό σε κάθε μία από τις κατηγορίες δικαιωμάτων (ιδιοκτήτη, ομάδα ιδιοκτήτη και υπόλοιποι). Ο αριθμός αυτός προκύπτει από το άθροισμα των αριθμών των δικαιωμάτων που θέλουμε να έχει η κάθε μία κατηγορία. Η ανάγνωση έχει τον αριθμό 4 η εγγραφή τον αριθμό 2 και η εκτέλεση το 1. Για παράδειγμα θέλουμε για το αρχείο ο ιδιοκτήτης να έχει όλα τα δικαιώματα, η ομάδα ιδιοκτήτη εγγραφή και εκτέλεση και οι άλλοι μόνο ανάγνωση πρέπει να δώσουμε τον αριθμό **7** (4+2+1) **3** (2+1) και **4** (4). Αυτό παρουσιάζεται και στο παρακάτω πλαίσιο:

```
[user@ekdda ~]$ chmod 734 testfile
[user@ekdda ~]$ ls -l testfile
-rwx-wxr--. 1 user user 0 Dec  3 16:02 testfile
```

Μία σημαντική παρατήρηση όσο αφορά τα δικαιώματα είναι το γεγονός ότι τα δικαιώματα δεν είναι αθροιστικά αλλά συγκεκριμένα. Για να γίνει πιο κατανοητό το παραπάνω ας δούμε ένα αρχείο **strange** με σχετικά «περίεργα» δικαιώματα όπως παρουσιάζεται στο παρακάτω πλαίσιο

```
[user@ekdda ~]$ ls -l strange
-r--rw----. 1 user user 0 Dec 16 17:13 strange
```

Το αρχείο αυτό παρατηρούμε ότι έχει ιδιοκτήτη τον `user` και ομάδα ιδιοκτήτη το `user private group` του `user`. Όσο αφορά τα δικαιώματα ο ιδιοκτήτης μπορεί μόνο να το διαβάσει αλλά η ομάδα ιδιοκτήτη μπορεί να το διαβάσει και να το γράψει. Ο `user` που ανήκει στην ομάδα `user` μπορεί να το γράψει ή όχι;

```
[user@ekdda ~]$ whoami
user
[user@ekdda ~]$ echo "Test write" >> strange
bash: strange: Permission denied
```

Όπως βλέπουμε τα δικαιώματα του `user` είναι τα δικαιώματα του ιδιοκτήτη (μόνο ανάγνωση) και όχι τα δικαιώματα της ομάδας `user`. Τα δικαιώματα της ομάδας `user` σε αυτό το αρχείο έχουν όλοι οι άλλοι χρήστες που ανήκουν στην ομάδα `user` εκτός από τον `user`.

### Προκαθορισμένα δικαιώματα

Όταν δημιουργείται ένα καινούργιο αρχείο ή κατάλογος αυτό έχει κάποια προκαθορισμένα δικαιώματα. Τα δικαιώματα αυτά καθορίζονται με τη χρήση αυτής εντολής **umask**. Αν δοθεί η εντολή **umask** χωρίς καμία παράμετρο εμφανίζει έναν αριθμό.

```
[user@ekdda ~]$ umask  
0002
```

Ο αριθμός αυτός περιγράφει, με την αριθμητική μορφή, ποια είναι τα δικαιώματα που ΔΕΝ έχουν οι τρεις κατηγορίες δικαιωμάτων σε ένα καινούργιο αρχείο κατάλογο που δημιουργείται. Στη συγκεκριμένη περίπτωση ο ιδιοκτήτης έχει όλα τα δικαιώματα (μηδέν), το γκρουπ ιδιοκτήτη πάλι όλα τα δικαιώματα (μηδέν) και οι υπόλοιποι δεν έχουν το δικαίωμα εγγραφής (δύο). Το αρχικό μηδενικό είναι για να περιγράψει τα ειδικά δικαιώματα που θα αναφερθούν παρακάτω. Αξίζει να σημειωθεί ότι ακόμα και με μηδέν στην **umask** για λόγους ασφαλείας ένα καινούργιο αρχείο που δημιουργείται δεν έχει το δικαίωμα εκτέλεσης (x).

### Δικαιώματα σε καταλόγους

Ενώ η λειτουργικότητα του δικαιώματος της ανάγνωσης, της εγγραφής και της εκτέλεσης ενός αρχείου είναι προφανής, δεν ισχύει το ίδιο και για τα δικαιώματα σε έναν κατάλογο. Μία παρατήρηση που μπορεί να διευκολύνει την κατανόηση της λειτουργικότητας των δικαιωμάτων σε καταλόγους είναι το γεγονός ότι στην ουσία οι καταλόγοι στο Linux είναι αρχεία που έχουν σαν περιεχόμενο τα αρχεία και τους καταλόγους. Επομένως το δικαίωμα ανάγνωσης σε έναν κατάλογο σημαίνει ότι είναι δυνατή η παράθεση των περιεχομένων του, δηλαδή εκτέλεση της εντολής **ls**. Ακόλουθα, το δικαίωμα εγγραφής σημαίνει ότι είναι δυνατή η μετατροπή των περιεχομένων του καταλόγου, δηλαδή η δημιουργία καινούργιων αρχείων ή διαγραφή τους μέσα σε αυτόν, και τέλος η εκτέλεση σημαίνει ότι είναι δυνατή η πρόσβαση σε αυτόν τον κατάλογο με τη χρήση της εντολής **cd**.

Ένα ωραίο παράδειγμα για την κατανόηση της λειτουργικότητας των δικαιωμάτων σε καταλόγους είναι το παρακάτω. Έστω ότι έχουμε το αρχείο **strong** στο οποίο έχουμε όλα τα δικαιώματα όπως εμφανίζεται στο παρακάτω πλαίσιο. Μπορούμε να το διαγράψουμε; Αν δοκιμάσουμε θα δούμε ότι δεν μπορούμε!

```
[user@ekdda test]$ ls -la strong  
-rwxrwxrwx. 1 user user 0 Dec 16 17:30 strong  
[user@ekdda test]$ rm strong  
rm: cannot remove 'strong': Permission denied
```

Αυτό οφείλεται στο γεγονός ότι για να διαγράψουμε ένα αρχείο, όπως και να το δημιουργήσουμε, δεν μας ενδιαφέρουν τα δικαιώματα που έχουμε στο αρχείο αλλά τα δικαιώματα που έχουμε στον κατάλογο που περιέχεται. Αν στον κατάλογο `test`, που περιέχεται το αρχείο `strong`, δεν έχουμε το δικαίωμα εγγραφής δεν μπορούμε να μετατρέψουμε τα περιεχόμενα του καταλόγου (δηλαδή να γράψουμε ή να διαγράψουμε αρχεία ή καταλόγους).

### Ειδικά δικαιώματα

Εκτός από τα δικαιώματα της ανάγνωσης της εγγραφής και της εκτέλεσης υπάρχουν και τρία ειδικά δικαιώματα (Carigan, 2020) που είναι τα παρακάτω:

**Set User Id (SUID):** Όταν εκτελούμε ένα αρχείο, π.χ. ένα `script`, τότε αυτό εκτελείται με τα δικαιώματα που έχουμε εμείς σαν χρήστες. Όμως ένα αρχείο που έχει ενεργό το SUID δικαίωμα επιτρέπει σε αυτόν που θα το εκτελέσει να το εκτελέσει με τα δικαιώματα του ιδιοκτήτη του αρχείου και όχι τα δικά του. Η ενεργοποίηση του SUID σε ένα αρχείο γίνεται με τη λεκτική μορφή της `chmod u+s`.

```
[user@ekdda ~]$ touch strange
[user@ekdda ~]$ chmod u+s strange
[user@ekdda ~]$ ls -l strange
-rwSrwx-r--. 1 user user 0 Dec 16 17:46 strange
```

Όπως παρατηρούμε το δικαίωμα SUID παρουσιάζεται με τον χαρακτήρα `s` στη θέση του δικαιώματος εκτέλεσης (`x`) του ιδιοκτήτη. Αν το δικαίωμα εκτέλεσης υπάρχει τότε το `s` είναι κεφαλαίο αλλιώς είναι μικρό.

**Set Group Id (SGID):** Η λειτουργικότητα του SGID είναι ίδια με αυτή του SUID με τη διαφορά ότι αυτός που εκτελεί το αρχείο έχει τα δικαιώματα της ομάδας ιδιοκτήτη του αρχείου. Ενδιαφέρον έχει η λειτουργικότητα που έχει ένας κατάλογος με το δικαίωμα SGID. Σε αυτή την περίπτωση τα αρχεία που δημιουργούνται σε αυτόν τον κατάλογο θα έχουν ομάδα ιδιοκτήτη την ομάδα ιδιοκτήτη του καταλόγου και όχι του χρήστη που τα δημιούργησε. Η ενεργοποίηση του SUID σε ένα αρχείο ή κατάλογο γίνεται με τη λεκτική μορφή της `chmod g+s`.

```
[root@ekdda ~]# mkdir /strangedir
[root@ekdda ~]# chmod 777 /strangedir
[root@ekdda ~]# chmod g+s /strangedir
[root@ekdda ~]# chown :web /strangedir/
[root@ekdda ~]# ls -ld /strangedir
drwxrwsrwx. 2 root web 4096 Jan  4 17:02 /strangedir
# εδώ χρήση σαν user
[user@ekdda ~]$ cd /strangedir
[user@ekdda strandedir]$ id
uid=1000(user) gid=1000(user) groups=1000(user),10(wheel)
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[user@ekdda strandedir]$ touch newfile
[user@ekdda strandedir]$ ls -l newfile
-rw-rw-r--. 1 user web 0 Jan  4 17:06 newfile
```

Επίσης σε αντιστοιχία με το SUID έτσι και το SGID εμφανίζεται με τον χαρακτήρα *s* στη θέση του δικαιώματος εκτέλεσης για την ομάδα ιδιοκτήτη.

**Sticky Bit:** Το δικαίωμα sticky bit έχει νόημα μόνο για τους καταλόγους. Σε έναν κατάλογο που έχει ενεργοποιημένο το δικαίωμα sticky bit μπορούν να δημιουργήσουν ή να διαγράψουν αρχεία όλοι οι χρήστες που έχουν δικαίωμα εγγραφής σε αυτόν. Όμως ο κάθε χρήστης μπορεί να διαγράψει μόνο τα δικά του αρχεία και όχι των άλλων χρηστών. Ένα χαρακτηριστικό παράδειγμα καταλόγου με ενεργοποιημένο το δικαίωμα sticky bit είναι ο κατάλογος **/tmp**.

```
[user@ekdda ~]$ ls -ld /tmp
drwxrwxrwt. 19 root root 420 Dec 16 18:03 /tmp
```



Η χρησιμοποίηση των SUID και των SGID δικαιωμάτων είναι απαραίτητη σε κάποια αρχεία συστήματος του Linux αλλά δημιουργούν σημαντικά κενά ασφαλείας όταν χρησιμοποιούνται σε άλλα αρχεία (Tevault, 2018).

## 4 Εγκατάσταση - Διαχείριση Βασικών Εξυπηρετητών - Διαχείριση Υπηρεσιών και System Monitoring

### 4.1 Χρήσιμες Εντολές Διαχείρισης

Πριν ξεκινήσουμε την περιγραφή των διαφόρων εξυπηρετητών (Servers) και των διαδικασιών εγκατάστασης θα πρέπει να αναφέρουμε κάποιες χρήσιμες εντολές οι οποίες επαναλαμβάνονται σε όλη τη διάρκεια των εγκαταστάσεων

Η πρώτη εντολή που θα δούμε είναι η **systemctl** (είναι σύντμηση των λέξεων system control)

Με τις παραμέτρους **start**, **stop**, **restart**

```
[root@ekdda ~]# systemctl start <υπηρεσία>
[root@ekdda ~]# systemctl stop <υπηρεσία>
[root@ekdda ~]# systemctl restart <υπηρεσία>
```

Γίνεται εκκίνηση, διακοπή και επανεκκίνηση μιας υπηρεσίας

Με την παράμετρο **status**

```
[root@ekdda ~]# systemctl status <υπηρεσία>
```

Μπορούμε να δούμε την κατάσταση στην οποία βρίσκεται μια υπηρεσία. Για παράδειγμα με την εκτέλεση της εντολής

```
[root@ekdda ~]# systemctl status httpd
```

βλέπουμε την κατάσταση στην οποία βρίσκεται η υπηρεσία httpd

```
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Sat 2020-11-28 11:25:21 EET; 8h ago
   Process: 5705 ExecReload=/usr/sbin/httpd $OPTIONS -k graceful (code=exited, status=0/SUCCESS)
  Main PID: 804 (httpd)
    Status: "Total requests: 26; Idle/Busy workers 100/0; Requests/sec: 0.000867; Bytes served/sec: 7 B/s"
     Tasks: 53 (limit: 4915)
    CGroup: /system.slice/httpd.service
            └─ 804 /usr/sbin/httpd -DFOREGROUND
               5723 /usr/sbin/httpd -DFOREGROUND
               5724 /usr/sbin/httpd -DFOREGROUND
               5725 /usr/sbin/httpd -DFOREGROUND
               5726 /usr/sbin/httpd -DFOREGROUND
               5727 /usr/sbin/httpd -DFOREGROUND
               5988 /usr/sbin/httpd -DFOREGROUND

Nov 28 14:42:16 mail.ekdda01.linux ownCloud[5724]: {core} Trusted domain error. "192.168.1.149" tried to
Nov 28 14:57:16 mail.ekdda01.linux ownCloud[5988]: {core} Trusted domain error. "192.168.1.149" tried to
Nov 28 15:12:16 mail.ekdda01.linux ownCloud[5725]: {core} Trusted domain error. "192.168.1.149" tried to
Nov 28 15:27:16 mail.ekdda01.linux ownCloud[5726]: {core} Trusted domain error. "192.168.1.149" tried to
Nov 28 15:42:16 mail.ekdda01.linux ownCloud[5727]: {core} Trusted domain error. "192.168.1.149" tried to
Nov 28 15:57:16 mail.ekdda01.linux ownCloud[5723]: {core} Trusted domain error. "192.168.1.149" tried to
Nov 28 16:12:16 mail.ekdda01.linux ownCloud[5724]: {core} Trusted domain error. "192.168.1.149" tried to
```

Εικόνα 4- 1: Αποτέλεσμα της εκτέλεσης της εντολής **systemctl status** της υπηρεσίας **httpd**

Στην παραπάνω Εικόνα 4- 1 βλέπουμε ότι η υπηρεσία είναι active από το Σάββατο 28/11/2020 (και ώρα 11:25:21) και επίσης βλέπουμε τα IDs των διεργασιών της υπηρεσίας. Τα σχετικά με τα IDs ή PIDs των διεργασιών θα τα δούμε αναλυτικά στην παράγραφο 5.3

Αν εκτελέσουμε τις εντολές

```
[root@ekdda ~]# systemctl stop httpd
[root@ekdda ~]# systemctl status httpd
```

Θα δούμε την Εικόνα 4- 2

```
root@ekdda ~]# systemctl stop httpd
root@ekdda ~]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Active: inactive (dead) since Sat 2020-11-28 19:51:31 EET; 1s ago
     Process: 5705 ExecReload=/usr/sbin/httpd $OPTIONS -k graceful (code=exited, status=0/SUCCESS)
     Process: 804 ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND (code=exited, status=0/SUCCESS)
    Main PID: 804 (code=exited, status=0/SUCCESS)
   Status: "Total requests: 26; Idle/Busy workers 100/0;Requests/sec: 0.000856; Bytes served/sec: 7 B/s"

Nov 28 15:12:16 mail.ekdda01.linux ownCloud[5725]: {core} Trusted domain error. "192.168.1.149" tried to
Nov 28 15:27:16 mail.ekdda01.linux ownCloud[5726]: {core} Trusted domain error. "192.168.1.149" tried to
Nov 28 15:42:16 mail.ekdda01.linux ownCloud[5727]: {core} Trusted domain error. "192.168.1.149" tried to
Nov 28 15:57:16 mail.ekdda01.linux ownCloud[5723]: {core} Trusted domain error. "192.168.1.149" tried to
Nov 28 16:12:16 mail.ekdda01.linux ownCloud[5724]: {core} Trusted domain error. "192.168.1.149" tried to
```

Εικόνα 4- 2: Διακοπή και έλεγχος λειτουργίας της υπηρεσίας httpd

Η παράμετρος **enable**

```
[root@ekdda ~]# systemctl enable <υπηρεσία>
```

επιτρέπει στην υπηρεσία να εκκινείται κατά τη διαδικασία της εκκίνησης του συστήματος (**system boot**).

Αντίστοιχα η παράμετρος **disable** σταματά την εκκίνηση της υπηρεσίας κατά τη διαδικασία εκκίνησης του συστήματος.

```
[root@ekdda ~]# systemctl disable <υπηρεσία>
```

Η δεύτερη εντολή σχετίζεται με το τείχος προστασίας (firewall) το οποίο αναφέρουμε πιο αναλυτικά στην παράγραφο 7.1.2. Η εντολή αυτή είναι η **firewall-cmd** (αντίστοιχα σύντμηση των λέξεων firewall command)

Με τις εντολές

```
[root@ekdda ~]# firewall-cmd --add-port=<port service>/tcp --
permanent
[root@ekdda ~]# firewall-cmd --remove-port=<port service>/tcp --
permanent
```

προσθέτουμε ή αφαιρούμε μια πόρτα υπηρεσίας από το firewall (για τις πόρτες διαβάστε την παράγραφο 6.4). Η παράμετρος **permanent** κάνει μόνιμη την προσθήκη ή την αφαίρεση αυτή.

Με τις εντολές

```
[root@ekdda ~]# firewall-cmd --add-service=<service> --permanent
[root@ekdda ~]# firewall-cmd --remove-service=<service> --
permanent
```

προσθέτουμε ή αφαιρούμε μια υπηρεσία από το firewall (για τις υπηρεσίες διαβάστε τις παραγράφους 5.1 και 6.4).

Με τις εντολές

```
[root@ekdda ~]# firewall-cmd --list-services
[root@ekdda ~]# firewall-cmd --list-ports
```

βλέπουμε μια λίστα με τις επιτρεπτές από το firewall (ανοικτές) πόρτες και υπηρεσίες. Μια εκτέλεση των παραπάνω εντολών βλέπουμε στην ακόλουθη Εικόνα 4- 3

```
[root@ekdda ~]# firewall-cmd --list-services
dhcprv6-client ssh samba-client mdns samba http smtp pop3 imap ftp nfs dhcp vnc-server telnet squid
[root@ekdda ~]# firewall-cmd --list-ports
1025-65535/udp 1025-65535/tcp 3389/tcp 80/tcp 443/tcp 9090/tcp
```

Εικόνα 4- 3: Εμφάνιση των υπηρεσιών των θυρών που είναι ανοικτές στο firewall

Με την εντολή

```
[root@ekdda ~]# firewall-cmd -reload
```

επαναφορτώνουμε τους κανόνες του firewall. Χρειάζεται επαναφόρτωση του firewall κάθε φορά που κάνουμε οποιαδήποτε προσθήκη ή αφαίρεση.

Τέλος με την εντολή

```
[root@ekdda ~]# systemctl stop/start firewalld
```

σταματούμε/ξεκινούμε την υπηρεσία firewall

Το firewall πλέον αποτελεί ξεχωριστή υπηρεσία (την firewalld) στη fedora και για το λόγο αυτό μπορούν να γίνουν όλες οι λειτουργίες επί αυτής (start/stop/restart/enable /disable).

### Ασκήσεις - Δραστηριότητες

Εκτελέστε τις εντολές της παραγράφου και δείτε τα αποτελέσματα που φαίνονται στις εικόνες



## 4.2 Εγκατάσταση και Διαχείριση X Window Server

Ο **X Window Server** ή **X Window System** ή **X Server** ή **Xorg** ή απλά **X** (διαφορετικές ονομασίες με τις οποίες είναι γνωστός και αναφέρεται στη βιβλιογραφία), είναι ο εξυπηρετητής (Server) που φιλοξενεί τα γραφικά περιβάλλοντα διεπαφής του χρήστη με το Linux. Λειτουργεί στη λογική του Client – Server, με την έννοια ότι υπάρχει ένας εξυπηρετητής ο οποίος μπορεί να εξυπηρετήσει διάφορους χρήστες, τοπικούς και απομακρυσμένους, διαθέτοντας σε αυτούς ένα γραφικό εύχρηστο περιβάλλον.



Οι διαφορετικές εκδόσεις του Linux ονομάζονται **διανομές**

Ο X εκτελείται σε μια μηχανή και εξυπηρετεί τόσο τους τοπικούς χρήστες όσο και τους απομακρυσμένους. Η υπηρεσία που παρέχει, είναι η δυνατότητα για μια γραφική διεπαφή (GUI – Graphical User Interface). Διαχειρίζεται τις συσκευές εισόδου (input devices), δηλαδή το πληκτρολόγιο και το ποντίκι (keyboard και mouse), καθώς και την κάρτα γραφικών (VGA card) καθώς και την οθόνη του υπολογιστή.

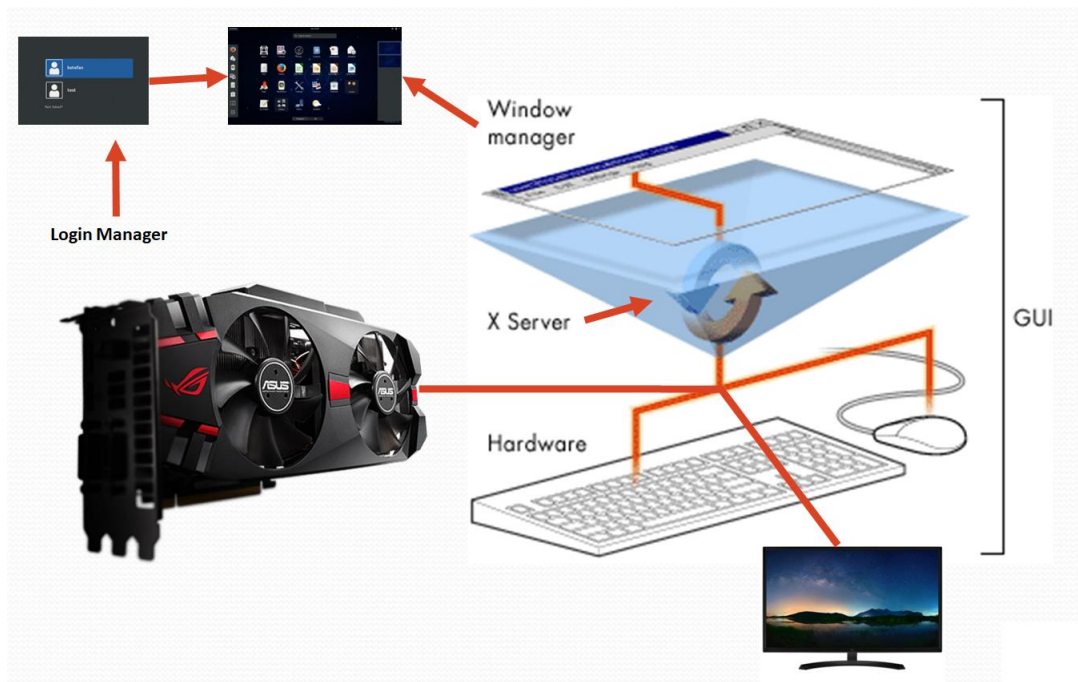
Διαχειρίζεται χρώματα και αναλύσεις, ανάλογα με το υλικό που διαθέτει ο Server. Η υλοποίηση λειτουργίας του X Server είναι μέσω ενός πρωτοκόλλου το οποίο είναι γνωστό με το όνομα X11 protocol. Ανεξάρτητα αν ο Client είναι τοπικός ή απομακρυσμένος η επικοινωνία ανάμεσα σε Server και Client γίνεται μέσω αυτού του πρωτοκόλλου.

Βασικά συστατικά που αποτελούν το γραφικό μέρος του X είναι :

- ο **X Window Manager (XWM)**, ο οποίος αποτελείται από
  - τον **Window Manager** το οποίο είναι ένα σύστημα που ελέγχει την εμφάνιση των παραθύρων, και γενικότερα όλες τις γραφικές λειτουργίες των παραθύρων (τη θέση τους, τα borders, αν θα έχουν στοιχεία ελέγχου όπως minimize close κλπ)
  - το **Desktop Environment** το οποίο περιλαμβάνει αυτό που ξέρουμε ως εφαρμογές συστήματος, τον file manager, την εμφάνιση των μενού στις εφαρμογές κλπ
- ο **Login ή Display Manager (DM)**

Παραδείγματα τέτοιων X Window Managers είναι τα GNOME, KDE, Xfce, Sugar, Mate κλπ. Θα δούμε πιο αναλυτικά οδηγίες εγκατάστασης XWM στην παράγραφο 4.2.4

Μια οπτικοποίηση του X Window Server μαζί με τον X Window Manager και τον Display ή Login Manager είναι αυτή που φαίνεται στην παρακάτω Εικόνα 4- 4. Στην ίδια εικόνα επίσης φαίνονται οι συσκευές υλικού που ελέγχει ο X Server (ποντίκι, πληκτρολόγιο, κάρτα γραφικών, οθόνη υπολογιστή).



Εικόνα 4- 4: Οπτικοποίηση του X Window Server μαζί με τον X Window Manager

Ο X Window Server μπορεί να είναι ενσωματωμένος στη διανομή ή όχι.



Αν κάποιος θέλει η διανομή του να περιέχει γραφικό μέρος καλύτερα να επιλέξει μια διανομή με ενσωματωμένο τον X Window Server αφού η μετέπειτα εγκατάστασή του μπορεί να αποδειχθεί πολύωρη και επίπονη διαδικασία με τις περισσότερες φορές να μην επιτυγχάνονται τα καλύτερα αποτελέσματα.



Μπορεί μια διανομή να περιέχει τον X Window Server αλλά να μην χρησιμοποιηθεί ποτέ αν ο διαχειριστής κρίνει ότι επιβαρύνει τη λειτουργία του συστήματος που φιλοξενεί τον Linux Server, χρησιμοποιώντας την τεχνική των runlevels η οποία περιγράφεται πιο κάτω στην παράγραφο 4.2.2

Στην επόμενη παράγραφο θα δούμε τι κάνει ο διαχειριστής του Linux Server

- όταν η διανομή δεν περιέχει τον X Window Server και θέλει να τον εγκαταστήσει
- όταν υπάρχει ο X Window Server αλλά δεν τον χρειάζεται και θέλει να έχει τη μηχανή μόνο σε λειτουργία γραμμής εντολών (command line mode)
- Ποια πρωτόκολλα διαμοιρασμού του X Window Server υπάρχουν, ποιες οι διαφορές τους και πώς λειτουργούν

#### 4.2.1 Εγκατάσταση του X όταν δεν υπάρχει στη διανομή

Στις διανομές που δεν περιλαμβάνουν τον X Windows Server η εγκατάσταση του γίνεται με την ακόλουθη εντολή

```
[root@ekdda ~]# dnf groupinstall "Fedora Workstation"
```

ή αν δουλεύει κανείς σε παλαιότερες εκδόσεις CENTOS χρησιμοποιούμε την εντολή yum (η εντολή dnf ενσωματώθηκε στο CENTOS στην έκδοση 8).

```
[root@ekdda ~]# yum groupinstall "X Window System"
```

Μετά την εγκατάσταση, για την παραμετροποίηση του X (Xorg configuration) χρειάζεται να παραμετροποιηθούν κάποια αρχεία.

Στις παλαιότερες εκδόσεις συναντά κανείς το αρχείο

`/etc/X11/xorg.conf`

το οποίο περιέχει τμήματα που αφορούν τις συσκευές εισόδου, την κάρτα γραφικών και το monitor τα οποία πρέπει να παραμετροποιηθούν κατάλληλα ανάλογα τις συσκευές που έχει το μηχάνημα που φιλοξενεί τον Linux Server. Σε πολλές περιπτώσεις που υπάρχουν ιδιαίτερες συσκευές, πρέπει να αναζητηθούν και οι κατάλληλοι οδηγοί των συσκευών (drivers) οι οποίοι πρέπει να δηλωθούν κατάλληλα στο παραπάνω αρχείο.

Στις πιο σύγχρονες εκδόσεις Fedora πλέον δεν υπάρχει ένα αρχείο, αλλά πολλαπλά. Αντί να υπάρχουν τμήματα σε ένα αρχείο, υπάρχουν διαφορετικά αρχεία σε έναν κατάλογο. Στην παρακάτω Εικόνα 4- 5 φαίνεται ένα παράδειγμα περιεχομένου αυτών των αρχείων

```
40-monitor.conf

Section "Monitor"
    Identifier   "VGA1"
    Option      "PreferredMode" "1280x1024"
EndSection

Restart your X system.

If it doesn't work try creating the next two files in the same path, and re

30-graphic.conf

Section "Device"
    Identifier   "Intel Integrated"
    Driver       "intel"
EndSection

50-screen.conf

Section "Screen"
    Identifier   "Default Screen"
    DefaultDepth 24
    SubSection  "Display"
        Depth     24
        Modes      "1280x1024" "1024x768" "640x480"
    EndSubSection
EndSection
```

Εικόνα 4- 5: Παράδειγμα παραμετροποίησης αρχείων του X Window Server

Τα αρχεία βρίσκονται στον κατάλογο

/etc/X11/xorg.conf.d/

Ο κατάλογος αυτός εμφανίζει το παρακάτω περιεχόμενο (περίπου, μπορεί όχι ακριβώς)

```
[root@ekdda ~]# ls -l /etc/X11/xorg.conf.d/
-rw-r--r--. 1 root root 350 Oct 18 2017 00-keyboard.conf
-rw-r--r--. 1 root root 350 Oct 18 2017 20-monitor.conf
-rw-r--r--. 1 root root 435 Oct 18 2017 30-graphic.conf
-rw-r--r--. 1 root root 323 Oct 18 2017 50-screen.conf
```

Μια χρήσιμη εντολή για την εύρεση της ισχύουσας ανάλυσης της οθόνης και των δυνατών αναλύσεων της κάρτας γραφικών του συστήματος είναι η ακόλουθη

# **xrandr**

Ένα στιγμιότυπο εκτέλεσης της παραπάνω εντολής φαίνεται στην παρακάτω Εικόνα 4- 6

```
[root@ekdda ~]# xrandr
Screen 0: minimum 320 x 200, current 1280 x 768, maximum 16384 x 16384
VGA-1 connected 1280x768+0+0 (normal left inverted right x axis y axis) 0mm x 0mm
 1368x766      59.95 +
 2560x1600     59.99   59.97
 1920x1440     60.00
 1856x1392     60.00
 1792x1344     60.00
 2048x1152     60.00
 1920x1200     59.88   59.95
 1920x1080     60.00
 1600x1200     60.00
```

Εικόνα 4- 6: Στιγμιότυπο εκτέλεσης της εντολής xrandr

Πολλές και αναλυτικές πληροφορίες για την παραμετροποίηση του Xorg μπορεί κανείς να βρει στο (Freebsd, n.d.)

#### 4.2.2 Απενεργοποίηση του X (τεχνική runlevels)

Η χρήση του X κάποιες φορές είναι θέμα αισθητικής του διαχειριστή του Server, κάποιες όμως φορές είναι και θέμα δυνατοτήτων του υπολογιστικού συστήματος που φιλοξενεί τον Linux Server. Αν ένα σύστημα εγκαταστάθηκε με το γραφικό περιβάλλον αλλά ο διαχειριστής δεν επιθυμεί να ενεργοποιηθεί κατά την εκκίνηση του συστήματος, επειδή για παράδειγμα οι υπολογιστικοί του πόροι δεν επιτρέπουν την καλή λειτουργία του όταν είναι φορτωμένο το γραφικό περιβάλλον, μπορεί να χρησιμοποιήσει τα **runlevels** προκειμένου να είναι απενεργοποιημένος στην εκκίνηση του συστήματος.

Τα runlevels είναι ένας τρόπος λειτουργίας (operation mode) στα λειτουργικά συστήματα υπολογιστών που εφαρμόζουν την αρχικοποίηση τύπου Unix System V. Συμβατικά, υπάρχουν επτά επίπεδα, αριθμημένα από μηδέν έως έξι. Μόνο ένα runlevel εκτελείται κατά την εκκίνηση. Ένα runlevel καθορίζει την κατάσταση του μηχανήματος μετά την εκκίνηση. Διαφορετικά επίπεδα εκτέλεσης συνήθως αντιστοιχίζονται (όχι απαραίτητα με κάποια συγκεκριμένη σειρά) στη λειτουργία ενός χρήστη, τη λειτουργία πολλαπλών

χρηστών χωρίς εκκίνηση υπηρεσιών δικτύου, τη λειτουργία πολλαπλών χρηστών με εκκίνηση υπηρεσιών δικτύου, τον τερματισμό συστήματος και τις καταστάσεις επανεκκίνησης συστήματος.

Η ακριβής λειτουργία αυτών των επιπέδων διαφέρει μεταξύ διανομών Linux. Για παράδειγμα, το runlevel 4 μπορεί να είναι μια κατάσταση λειτουργίας (πχ δυνατότητα για καθορισμό από το χρήστη) του Server σε μια διανομή και μια άλλη κατάσταση λειτουργίας σε μια άλλη διανομή ή να μην είναι τίποτα απολύτως σε μια άλλη διανομή.

Τα runlevels ακολουθούν συνήθως τις γενικές διαμορφώσεις που περιγράφονται στον Πίνακα 4- 1. Ωστόσο, όπως είπαμε, ορισμένες διανομές χρησιμοποιούν διαφορετικές διαμορφώσεις. Σε μια τυπική μορφή λειτουργίας των runlevels, όταν ένας υπολογιστής εισέρχεται στο μηδέν runlevel, διακόπτεται και όταν μπαίνει στο επίπεδο runlevel έξι, επανεκκινείται. Τα ενδιάμεσα runlevels (1-5) διαφέρουν ως προς το ποιες υπηρεσίες ξεκινούν και ως προς το αν έχουν δίκτυο ή όχι. Τα προεπιλεγμένα επίπεδα εκτέλεσης είναι συνήθως 3 ή 5. Τα χαμηλότερα επίπεδα λειτουργίας είναι χρήσιμα για συντήρηση ή επισκευές έκτακτης ανάγκης, καθώς συνήθως δεν προσφέρουν καθόλου υπηρεσίες δικτύου. Οι συγκεκριμένες λεπτομέρειες της λειτουργίας των runlevels διαφέρουν μεταξύ των διαφορετικών διανομών.

Στον Πίνακα 4- 1 φαίνονται τα διαφορετικά runlevels :

| ID | Όνομα                               | Περιγραφή Λειτουργίας   |
|----|-------------------------------------|---|
| 0  | Halt                                | Τερματισμός   |
| 1  | Single-user mode                    | Κατάσταση λειτουργίας για διαχειριστικούς λόγους                        |
| 2  | Multi-user mode                     | Εκκίνηση με χρήση γραμμής εντολών (χωρίς γραφικό περιβάλλον)            |
| 3  | Multi-user mode with networking     | Εκκίνηση με χρήση γραμμής εντολών (χωρίς γραφικό περιβάλλον) και δίκτυο |
| 4  | Not used/user-definable             | Για ειδικές χρήσεις   |
| 5  | With GUI (Graphical User Interface) | Εκκίνηση με χρήση γραφικού περιβάλλοντος                                |
| 6  | Reboot                              | Επανεκκίνηση  |

Πίνακας 4- 1: Runlevels και λειτουργίες (WikiRunlevels, n.d.)

Για τις διάφορες λειτουργίες των runlevels και τη χρησιμότητά τους υπάρχουν πολλές πηγές στο διαδίκτυο όπου κανείς μπορεί να ανατρέξει (μια από αυτές και η (WikiRunlevels, n.d.))

Εμείς στα πλαίσια αυτού του κειμένου θα επικεντρωθούμε στα runlevels με IDs 3 και 5.

Η επιλογή runlevel με ID 3 οδηγεί σε εκκίνηση του συστήματος με γραμμή εντολών και δίκτυο, χωρίς γραφικό

Η επιλογή runlevel με ID 5 οδηγεί σε εκκίνηση του συστήματος με γραφικό

Για να επιλέξουμε runlevel 3 πρέπει να εκτελέσουμε την εντολή

```
[root@ekdda ~]# systemctl set-default multi-user.target
```

και στη συνέχεια επανεκκίνηση

Για να επιλέξουμε runlevel 5 πρέπει να εκτελέσουμε την εντολή (και στη συνέχεια επανεκκίνηση)

```
[root@ekdda ~]# systemctl set-default graphical.target
```

Οι αριθμητικές ονομασίες των runlevels έχουν καθιερωθεί από τον παλαιότερο τρόπο εναλλαγής τους. Σε παλιότερες εκδόσεις των διανομών η επιλογή του runlevel γινόταν από το αρχείο **inittab** του οποίου το περιεχόμενο ήταν ως εξής :

```
[root@ekdda ~]# more /etc/inittab
# Default runlevel. The runlevels used by RHS are:
# 0 - halt (Do NOT set initdefault to this)
# 1 - Single user mode
# 2 - Multiuser, without NFS (The same as 3, if you do not have
networking)
# 3 - Full multiuser mode
# 4 - unused
# 5 - X11
# 6 - reboot (Do NOT set initdefault to this)
#
id:3:initdefault
```

Το νούμερο στην τελευταία γραμμή όριζε το runlevel του συστήματος

#### Ασκήσεις - Δραστηριότητες

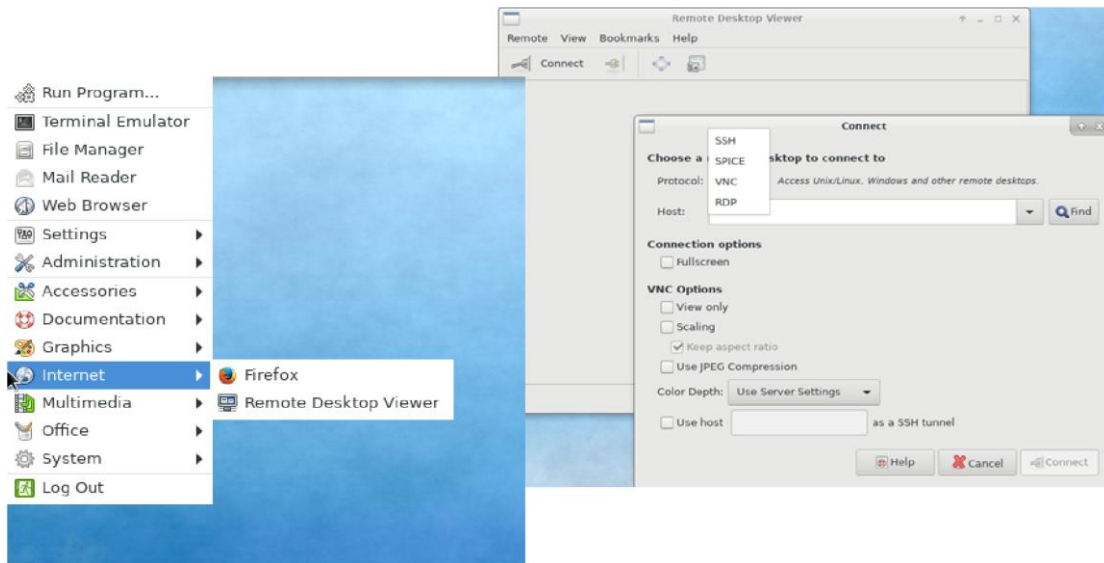
- Αλλάξτε το runlevel σε 3 και κάντε επανεκκίνηση, ώστε να δείτε τη διαφορά από την εκκίνηση σε 5
- Αλλάξτε το runlevel σε 5 και κάντε επανεκκίνηση, ώστε να δείτε τη διαφορά από την εκκίνηση σε 3

#### 4.2.3 Πρωτόκολλα απομακρυσμένης πρόσβασης στον X

Τα πρωτόκολλα απομακρυσμένης πρόσβασης με τα οποία θα ασχοληθούμε στο συγκεκριμένο κείμενο είναι δύο. Πριν δούμε τα πρωτόκολλα θα δούμε την εγκατάσταση ενός Linux Client (ένα λογισμικό που μπορεί να συνδεθεί με τον Server) ο οποίος μπορεί να συνδεθεί απομακρυσμένα χρησιμοποιώντας και τα δύο πρωτόκολλα. Ο Client που θα χρησιμοποιήσουμε είναι ο Vinagre (ή Remote Desktop Viewer). Για την εγκατάστασή του ως διαχειριστές εκτελούμε τις εντολές

```
[root@ekdda ~]# dnf -y install vinagre
```

Ο Vinagre εγκαθίσταται στο μενού, στην υποκατηγορία Internet με το όνομα Remote Desktop Viewer (Εικόνα 4- 7) για τον Xfce X Windows Manager.



Εικόνα 4- 7: Εγκατάσταση και εικόνα του Vinagre κατά το άνοιγμα

Μπορούμε να παρατηρήσουμε ότι ο Vinagre έχει δυνατότητα σύνδεσης και με τα δύο πρωτόκολλα (RDP και VNC) που θα δούμε στα επόμενα καθώς και με το πρωτόκολλο SSH.

#### 4.2.3.1 Πρωτόκολλο RDP

Όλα τα λειτουργικά συστήματα επικοινωνούν με τον υπόλοιπο κόσμο (εξωτερικό δίκτυο) μέσα από πόρτες. Σε κάθε μια από αυτές τις πόρτες μπορεί να «ακούει» και να εξυπηρετεί αιτήματα και μια υπηρεσία. Μια αναλυτική περιγραφή για υπηρεσίες, διεργασίες και πόρτες των λειτουργικών συστημάτων μπορούμε να βρούμε στις παραγράφους 5.1 και 6.4.

Μια από αυτές τις πόρτες, φιλοξενεί και την υπηρεσία MS WBT Service (Windows Based Terminal Service, η οποία είναι η επίσημη ονομασία της υπηρεσίας που χρησιμοποιεί το RDP remote desktop protocol and connection) με προεπιλεγμένη πόρτα την 3389 η οποία χρησιμοποιείται παγκοσμίως για τον Windows Remote Server.

Η 3389 είναι μια πόρτα η οποία όταν είναι ορατή στο διαδίκτυο (χωρίς προστασία από κάποιο firewall) είναι πολλές φορές στόχος για τους Hackers και ευάλωτη σε DOS attacks (είδος επίθεσης το οποίο θα το δούμε αναλυτικά στην παράγραφο 7.1.5.3).



Επειδή η πόρτα 3389 όταν είναι ορατή στο διαδίκτυο (χωρίς προστασία από κάποιο firewall) είναι πολλές φορές στόχος για τους Hackers, χρησιμοποιείται συνήθως από τους διαχειριστές σε εσωτερικά δίκτυα (intranets). Αν όμως πρέπει οπωσδήποτε να είναι ορατή στο εξωτερικό δίκτυο είναι καλύτερα να εφαρμόζεται μια τεχνική masquerating και port forwarding (την τεχνική αυτή θα τη δούμε πιο αναλυτικά στην παράγραφο 7.1.3).



Το remote desktop protocol το οποίο είναι ιδιόκτητο πρωτόκολλο (proprietary protocol) που αναπτύχθηκε και ανήκει στη Microsoft είναι ιδιαίτερα χρήσιμο όταν θέλουμε να δουλέψουμε απομακρυσμένα στο σύστημά μας με τη χρήση του γραφικού περιβάλλοντος.

Ο X Window Server μπορεί να γίνει export στα Windows μέσω του πρωτοκόλλου RDP με τη βοήθεια του **xrdp Server**. Ο xrdp χρησιμοποιεί την υπηρεσία MS WBT και την πόρτα 3389.

Για την εγκατάσταση του xrdp εκτελούμε τις παρακάτω εντολές

```
root@ekdda ~]# dnf -y install xrdp
```

Μετά την εγκατάσταση χρειάζεται εκκίνηση της υπηρεσίας με την εντολή **start**.

```
[root@ekdda ~]# systemctl start xrdp  
[root@ekdda ~]# systemctl enable xrdp
```

Όπως έχουμε δει στην παράγραφο 4.1 η εντολή **enable** επιτρέπει στην υπηρεσία να εκκινείται κατά τη διαδικασία της εκκίνησης του συστήματος (**system boot**).

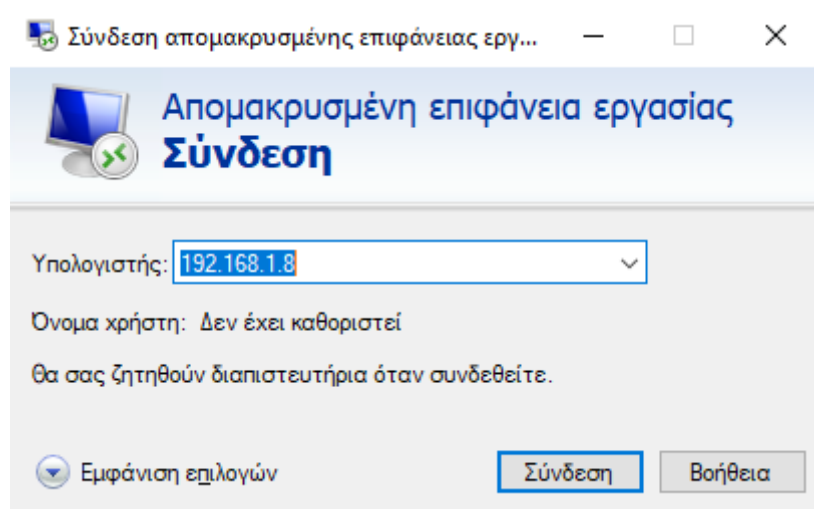
Τέλος χρειάζεται το άνοιγμα της υπηρεσίας στο firewall με την εγγραφή της πόρτας (δείτε και παραγράφους 4.1 και 7.1.2). Η παράμετρος permanent σημαίνει ότι το άνοιγμα της υπηρεσίας στο firewall θα είναι μόνιμο.

```
[root@ekdda ~]# firewall-cmd --add-port=3389/tcp --permanent  
[root@ekdda ~]# firewall-cmd --reload
```

Στη συνέχεια θα πρέπει να εγκατασταθεί ένας κατάλληλος RDP Client στο απομακρυσμένο σύστημα, από το οποίο θα έχουμε πρόσβαση στον X Server. Αυτό το απομακρυσμένο σύστημα (πελάτης – Client) μπορεί να έχει οποιοδήποτε λειτουργικό σύστημα. Στη συνέχεια περιγράψουμε σύνδεση από Windows και Linux Client.

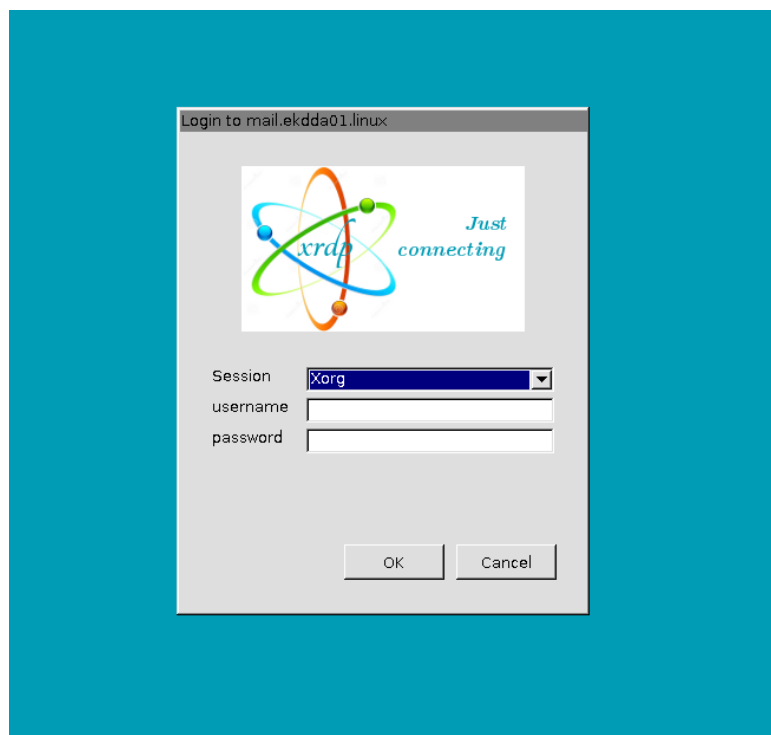
Αν θέλουμε να χρησιμοποιήσουμε Windows Client για να συνδεθούμε στον RDP Server μπορούμε να χρησιμοποιήσουμε τη «Σύνδεση Απομακρυσμένης Επιφάνειας Εργασίας» (Remote Desktop Connection - Εικόνα 4- 8). Το λογισμικό αυτό περιέχεται δωρεάν σε όλες τις καινούριες εκδόσεις των Windows.



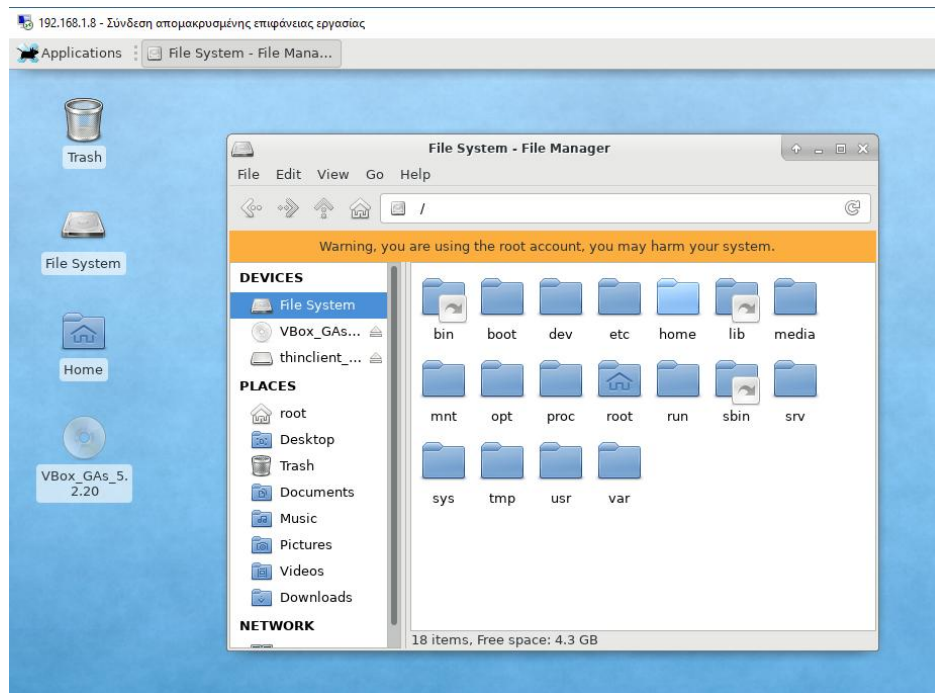


Εικόνα 4- 8: Παράθυρο σύνδεσης στην «Απομακρυσμένη Επιφάνειας Εργασίας»

Αφού πατήσουμε σύνδεση εμφανίζεται το παράθυρο σύνδεσης στον απομακρυσμένο RDP SERVER, όπου αφού επιλέξουμε το Xorg πρωτόκολλο, συνδεόμαστε με τον κατάλληλο χρήστη και βλέπουμε τις Εικόνα 4- 9 και Εικόνα 4- 10.



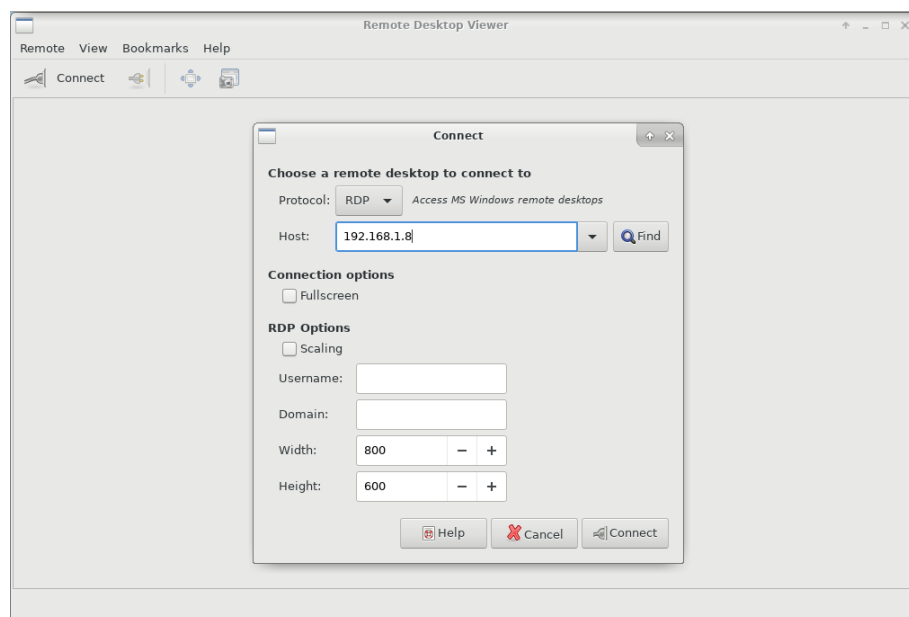
Εικόνα 4- 9: Παράθυρο σύνδεσης στον απομακρυσμένο RDP Server από Windows Client



Εικόνα 4- 10: Διαμοιρασμένο γραφικό session από τον RDP Server σε Windows Client

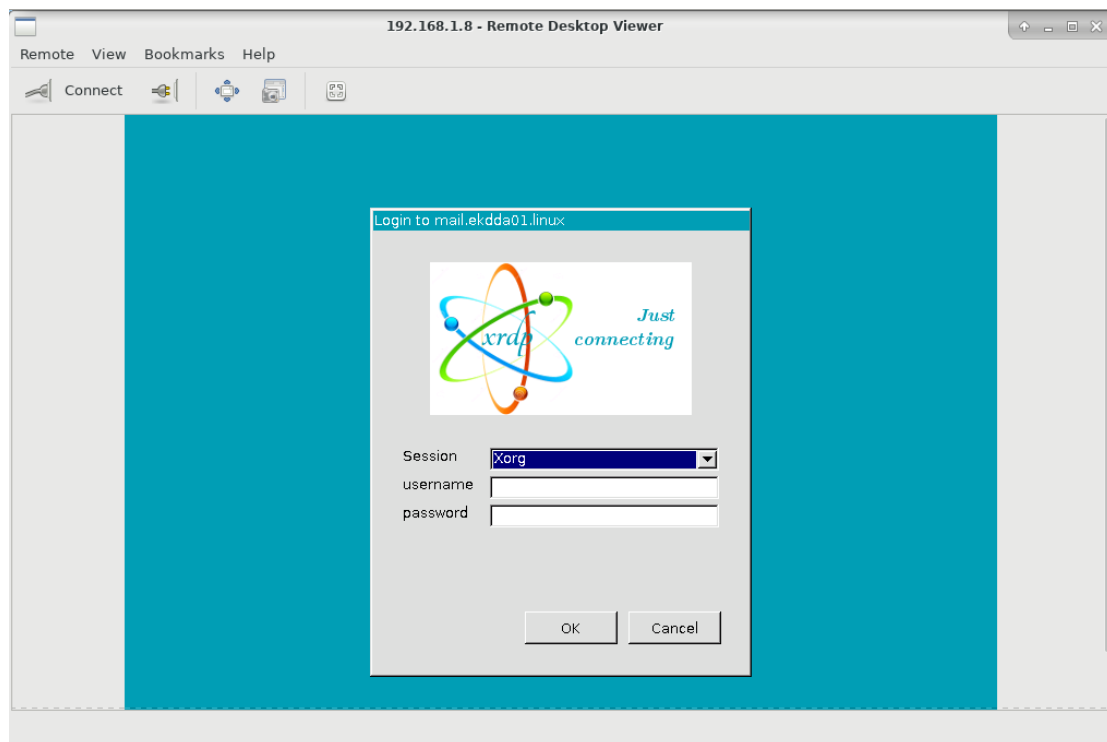
Στη συνέχεια αν θέλουμε να χρησιμοποιήσουμε σύστημα Linux ώστε να συνδεθούμε στον RDP Server απομακρυσμένα, μπορούμε να χρησιμοποιήσουμε τον Vnagre (ή Remote Desktop Viewer) του οποίου την εγκατάσταση περιγράψαμε στην προηγούμενη παράγραφο 4.2.3

Στην Εικόνα 4- 11 βλέπουμε το παράθυρο σύνδεσης του Vnagre στον RDP Server (επιλεγμένο πρωτόκολλο σύνδεσης το RDP).

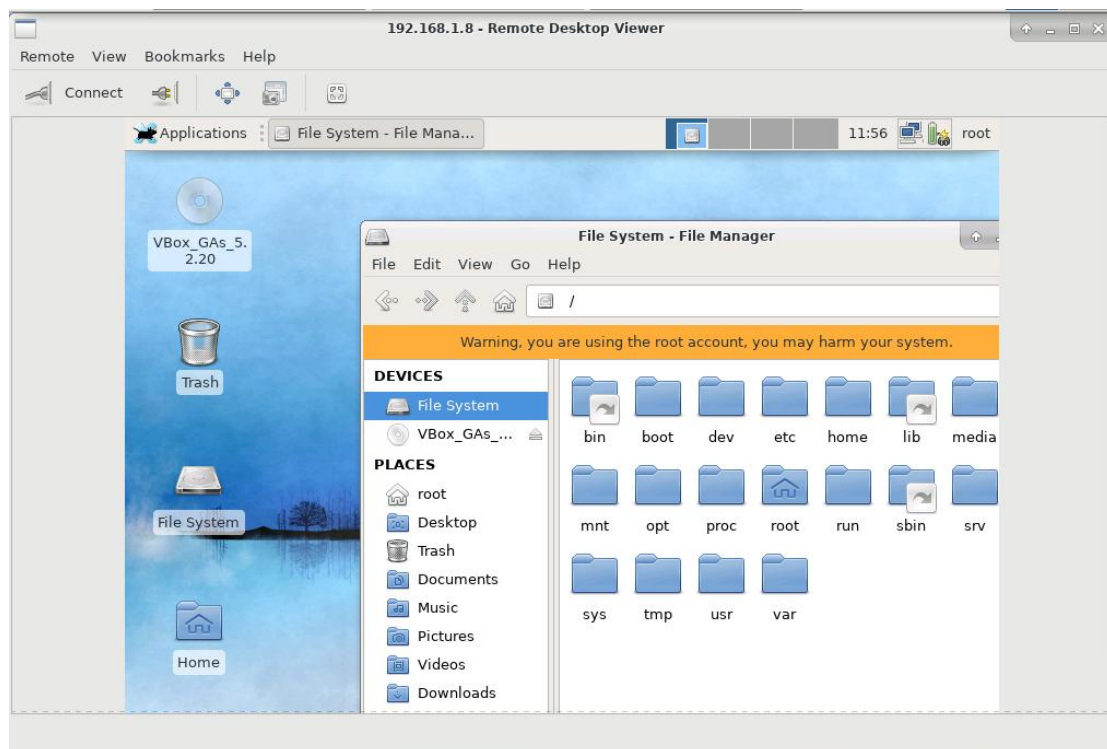


Εικόνα 4- 11: Παράθυρο σύνδεσης του Vnagre στον RDP Server

Πατώντας το κουμπί Connect (σύνδεση) εμφανίζεται το παράθυρο σύνδεσης στον απομακρυσμένο RDP SERVER, όπου αφού επιλέξουμε το Xorg πρωτόκολλο, συνδεόμαστε με τον κατάλληλο χρήστη και βλέπουμε τις Εικόνα 4- 12 και Εικόνα 4- 13



Εικόνα 4- 12: Παράθυρο σύνδεσης στον απομακρυσμένο RDP Server από Linux Client



Εικόνα 4- 13: Διαμοιρασμένο γραφικό session από τον RDP Server σε Linux Client



Ο RDP Server δίνει διαφορετικό DISPLAY (GUI session) σε κάθε απομακρυσμένο χρήστη. Για τον όρο DISPLAY μπορούμε να δούμε περισσότερα ακριβώς πιο κάτω.

### Ασκήσεις - Δραστηριότητες

- Εκτελέστε τις εντολές της παραγράφου και δείτε τα αποτελέσματα που φαίνονται στις εικόνες
- Κάντε μια σύνδεση από Windows στον xrdp
- Πραγματοποιήστε μια σύνδεση από Linux στον xrdp

#### 4.2.3.2 Πρωτόκολλο VNC

Το **VNC** είναι ακρωνύμιο του Virtual Network Computing. Είναι μια ανοικτή πλατφόρμα ανεξάρτητη από το γραφικό σύστημα που διαμοιράζεται, σχεδιασμένη για απομακρυσμένο έλεγχο ενός άλλου υπολογιστικού συστήματος. Εκπρόσωποι τέτοιων ανοικτών τεχνολογιών είναι : RealVNC, TightVNC, TurboVNC, UltraVNC, κ.τ.λ.

Το πρωτόκολλο **VNC** συνδέει έναν απομακρυσμένο χρήστη διαμοιράζοντας ένα DISPLAY (GUI session) του X Window Server (άρα οθόνη, πληκτρολόγιο και ποντίκι). Επομένως όσοι είναι συνδεδεμένοι στο ίδιο αυτό DISPLAY, ακόμα και αν μια σύνδεση είναι τοπικά στη φυσική θέση του υπολογιστικού συστήματος που φιλοξενεί τον Linux Server, βλέπουν όλοι τα ίδια. Μια χρήση είναι να δουλεύει κάποιος την ίδια οθόνη από διάφορες θέσεις ή αν υπάρχουν δύο – τρεις χρήστες και διαχειριστές και θέλει κάποιος να προσφέρει μια βοήθεια σε κάποιον άλλον σε πραγματικό χρόνο. Είναι δυνατόν όμως να υπάρξουν και διαφορετικά sessions αν υπάρξει σύνδεση σε διαφορετικά DISPLAYS.



κάθε DISPLAY στο Linux είναι ένα GUI session του X Window Server με «δικά του» ποντίκι, οθόνη, πληκτρολόγιο. Αυτό το DISPLAY μπορεί να γίνει EXPORT σε απομακρυσμένα συστήματα. Σε διαφορετικές βιβλιογραφίες, συναντούμε τον όρο DISPLAY και ως SCREEN ή MONITOR. Εμείς σε αυτό το κείμενο θα χρησιμοποιήσουμε τον όρο DISPLAY, επειδή έτσι χρησιμοποιείται ξεκάθαρα από το πρωτόκολλο VNC

Η εγκατάσταση του VNC Server γίνεται με τις παρακάτω εντολές, μαζί με το άνοιγμα της υπηρεσίας στο firewall και την επανεκκίνηση του firewall.

```
[root@ekdda ~]# dnf -y install tigervnc-server
[root@ekdda ~]# firewall-cmd --add-service=vnc-server --permanent
[root@ekdda ~]# firewall-cmd --reload
```


Το **TigerVNC** είναι μια δωρεάν εφαρμογή ανοιχτού κώδικα, υψηλής απόδοσης, ανεξάρτητη από πλατφόρμα υλοποίησης του Server (έχει εκδόσεις και για Linux και για Unix και για Windows). Είναι μια εφαρμογή Client / Server που επιτρέπει στους χρήστες να χρησιμοποιούν και να αλληλεπιδράσουν με γραφικές εφαρμογές σε απομακρυσμένα συστήματα. Ο tigervnc-server χρησιμοποιεί έναν μηχανισμό που διαμορφώνει μια

αυτόνομη εικονική επιφάνεια εργασίας για κάθε χρήστη. Είναι ικανό να εκτελεί εφαρμογές 3D και βίντεο και προσπαθεί να διατηρήσει σταθερή διεπαφή χρήστη και να επαναχρησιμοποιήσει στοιχεία, όπου είναι δυνατόν, σε διάφορες πλατφόρμες που υποστηρίζει. Επιπλέον, προσφέρει ασφάλεια μέσω μιας σειράς επεκτάσεων που εφαρμόζουν προηγμένες μεθόδους ελέγχου ταυτότητας και κρυπτογράφησης TLS (TigerVNC, n.d.).

Στη συνέχεια χρειάζεται να οριστεί ένα password για τον VNC Server.

```
[root@ekdda ~]# vncpasswd
Password:
Verify:
Would you like to enter a view-only password (y/n)? n
```

Στο παραπάνω τμήμα κώδικα βλέπουμε ότι ζητάει ένα password για τα vnc sessions και στη συνέχεια ζητάει από το χρήστη να ορίσει εάν θέλει και ένα password για view-only mode, όπου ο διασυνδεδεμένος χρήστης θα έχει μόνο πρόσβαση για να βλέπει, χωρίς να μπορεί να επέμβει. Εάν δε χρειαζόμαστε τέτοιο mode χρήσης τότε απαντάμε **n (no)** αλλιώς απαντάμε **y (yes)**.

 το password του vnc αποτελείται από 6 τουλάχιστον χαρακτήρες ή ψηφία

Στη συνέχεια χρειάζεται να εκκινήσουμε το service του vnc ορίζοντας το DISPLAY στο οποίο κάποιος θα μπορεί να συνδεθεί και με ποια γεωμετρία. Η γεωμετρία στην ουσία αποτελεί τον ορισμό της ανάλυσης της οθόνης. Στον παρακάτω κώδικα ορίζεται ότι ο VNC Server θα «ακούει» στο DISPLAY 1 με γεωμετρία (ανάλυση) 800x600 pixels.

```
[root@ekdda ~]# vncserver :1 -geometry 800x600
```

Μόλις εκτελεστεί η παραπάνω εντολή δημιουργούνται ένας κρυφός υποκατάλογος (dot folder) και κάποια αρχεία στο home directory του χρήστη ο οποίος εκτέλεσε την παραπάνω εντολή τα οποία επιτρέπουν μια σχετική παραμετροποίηση του VNC Server. Ο κατάλογος είναι ο

**/home/user/.vnc**

Και τα αρχεία είναι τα

**/home/user/.vnc/xstartup**

**/home/user/.vnc/config**

Επίσης δημιουργούνται στον ίδιο κατάλογο αρχεία καταγραφής δραστηριότητας στο συγκεκριμένο DISPLAY του συγκεκριμένου Server. Για παράδειγμα :

**/home/user/.vnc/192.168.1.11:1.log**

Στην περίπτωση που θέλουμε να σταματήσουμε τη συγκεκριμένη υπηρεσία στο συγκεκριμένο DISPLAY, χρησιμοποιούμε την παράμετρο kill, ως εξής

```
[root@ekdda ~]# vncserver -kill :1  
  
Killing Xvnc process ID 1721
```

Στην περίπτωση που θέλουμε να αλλάξουμε τον X Window Manager που φορτώνει κατά τη χρήση του VNC Server παραμετροποιούμε το αρχείο xstartup που έχει πλέον δημιουργηθεί στον κρυφό κατάλογο .vnc

```
[root@ekdda ~]# gedit ~/.vnc/xstartup
```

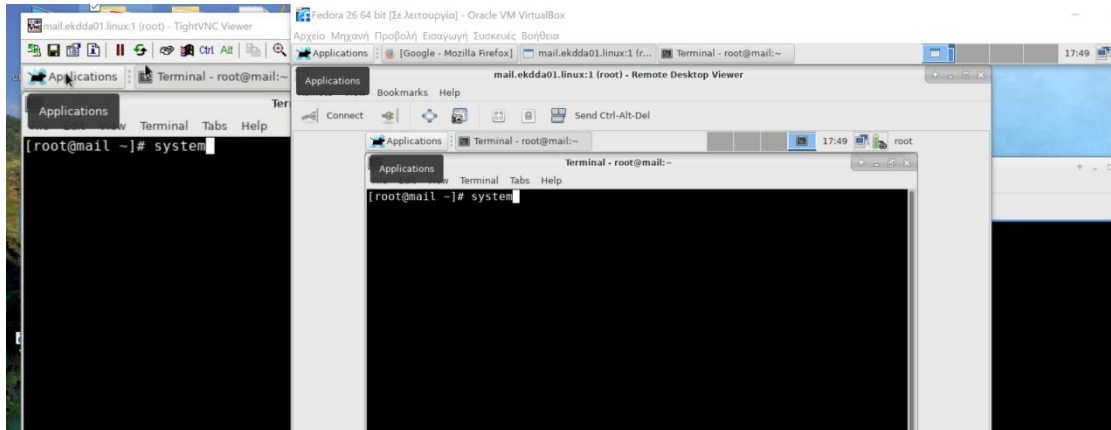
και στη συνέχεια προσθέτουμε (ανάλογα τον X Window Manager) εντολές με τον ακόλουθο τρόπο

```
#exec /etc/X11/xinit/xinitrc  
exec /usr/bin/xfce4-session
```

Η παραπάνω προσθήκη στο συγκεκριμένο αρχείο καθορίζει ότι αν κάποιος συνδεθεί με VNC Server σε οποιοδήποτε DISPLAY θα εξυπηρετηθεί με Xfce X Window Manager.

Στη συνέχεια θα δούμε εικόνες από τη σύνδεση Windows και Linux Client στον VNS Server.

Στην παρακάτω Εικόνα 4- 14 μπορούμε να δούμε δυο χρήστες που είναι συνδεδεμένοι από διαφορετικά σημεία και από διαφορετικά λειτουργικά στο ίδιο DISPLAY (στο ίδιο GUI session) και βλέπουν ακριβώς την ίδια εικόνα.

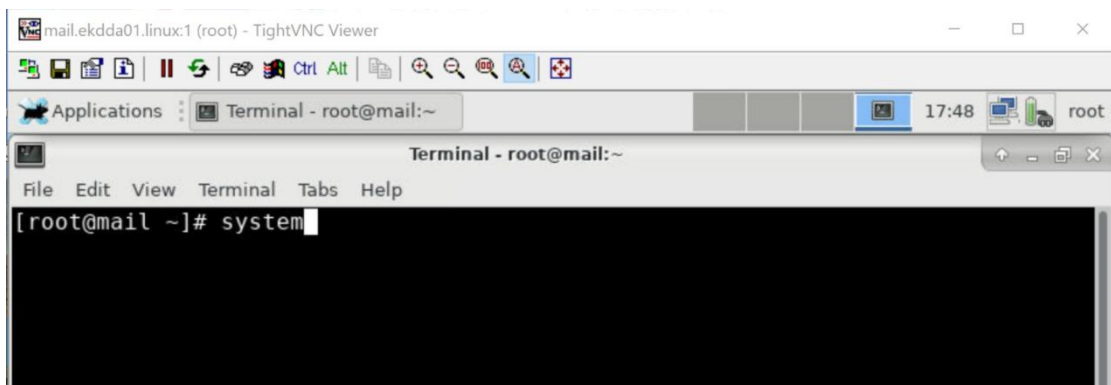


Εικόνα 4- 14: Δύο συνδέσεις VNC στο ίδιο Display (1) ταυτόχρονα και από Windows και από Linux (UltraVNC και VInagre αντίστοιχα). Βλέπουν ακριβώς το ίδιο

Για να φτάσουμε στην Εικόνα 4- 14 πρέπει πρώτα να ζητήσουμε σύνδεση με τον κατάλληλο Client. Στη συγκεκριμένη εικόνα έχει γίνει σύνδεση με τον TightVNC Client από Windows (Εικόνα 4- 15 και Εικόνα 4- 16).

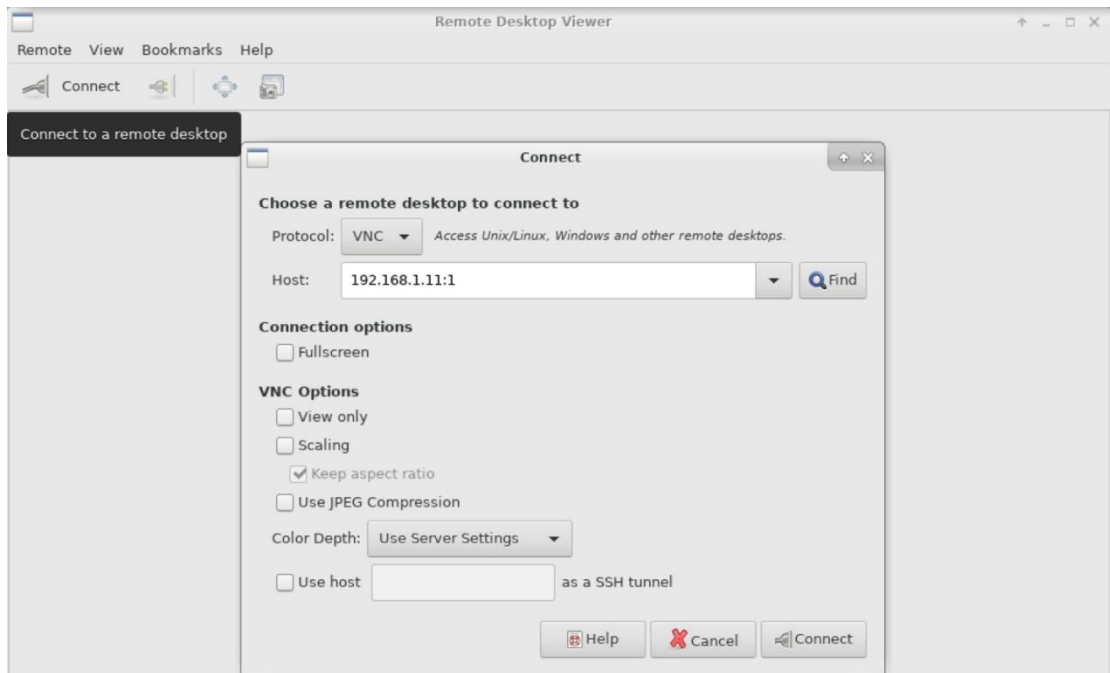


Εικόνα 4- 15: Οθόνη σύνδεσης στο Display 1 στον 192.168.1.11 Linux Fedora Server με VNC από Client σε Windows

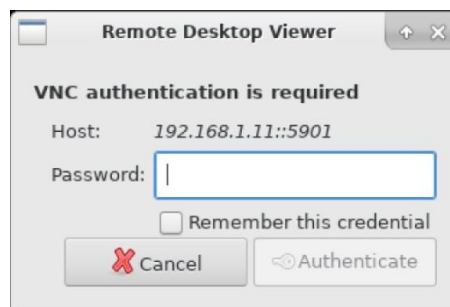


Εικόνα 4- 16: Οθόνη με το session στο Display 1 στον 192.168.1.11 Linux Fedora Server με VNC από Client σε Windows

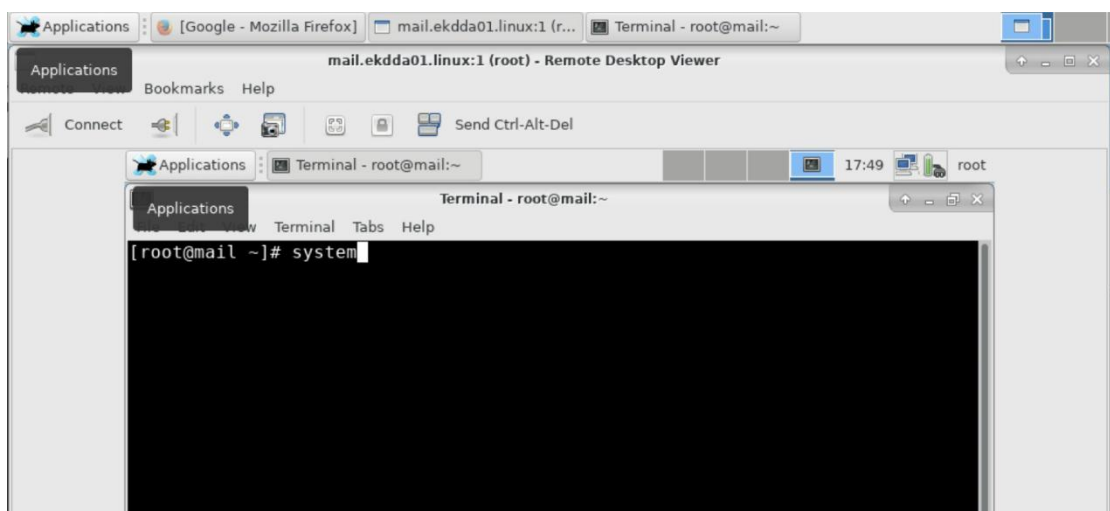
Στη συνέχεια γίνεται σύνδεση στον ίδιο Server και στο ίδιο Display (1) από Linux VNC Client (Εικόνα 4- 17, Εικόνα 4- 18 και Εικόνα 4- 19), πιο συγκεκριμένα τον VInagre του οποίου την εγκατάσταση περιγράψαμε στην προηγούμενη παράγραφο 4.2.3.



Εικόνα 4- 17: Οθόνη διασύνδεσης του remote desktop viewer (Vinagre) στον Server 192.168.1.11 στο DISPLAY 1



Εικόνα 4- 18: Οθόνη login του remote desktop viewer (Vinagre)



Εικόνα 4- 19: Οθόνη με το session στο Display 1 στον 192.168.1.11 Linux Fedora Server με VNC από Client σε Linux



### Ασκήσεις - Δραστηριότητες

- Εκτελέστε τις εντολές της παραγράφου και δείτε τα αποτελέσματα που φαίνονται στις εικόνες
- Κάντε μια σύνδεση από Windows στον vnc Server σε κάποιο display
- Πραγματοποιήστε μια σύνδεση από Linux στον vnc Server στο ίδιο display
- Πραγματοποιήστε μια σύνδεση από Linux στον vnc Server στο άλλο display
- «Σκοτώστε» ως διαχειριστές από τον Linux Server τα display sessions

#### 4.2.4 Εγκατάσταση X Window Manager (WM)

ο X Window Manager (WM), κατ' ουσία είναι το γραφικό περιβάλλον, ή αλλιώς η διεπαφή (interface) του χρήστη με το λειτουργικό σύστημα. Αποτελείται από τον Window Manager το οποίο είναι ένα σύστημα που ελέγχει την εμφάνιση των παραθύρων, και γενικότερα όλες τις γραφικές λειτουργίες των παραθύρων (τη θέση τους, τα όρια, αν θα έχουν στοιχεία ελέγχου όπως ελαχιστοποίηση, κλείσιμο κτλ) και το Desktop Environment το οποίο περιλαμβάνει αυτό που ξέρουμε ως εφαρμογές συστήματος, τον διαχειριστή αρχείων, την εμφάνιση των μενού στις εφαρμογές κλπ. Αυτά τα δύο είναι ενσωματωμένα σε ένα πακέτο.

Παραδείγματα τέτοιων X Window Manager είναι τα GNOME, KDE, Xfce, Sugar, Mate, Hawaii κλπ. Η επιλογή του κάθε χρήστη για τον X Window Manager που θα δουλέψει είναι κάτι πολύ προσωπικό και τις περισσότερες φορές σχετίζεται με το τι έχει συνηθίσει ο κάθε χρήστης.

Η εγκατάσταση ενός τέτοιου γραφικού περιβάλλοντος γίνεται με τις εντολές (εγκατάσταση των kde, xfce, hawaii και sugar).

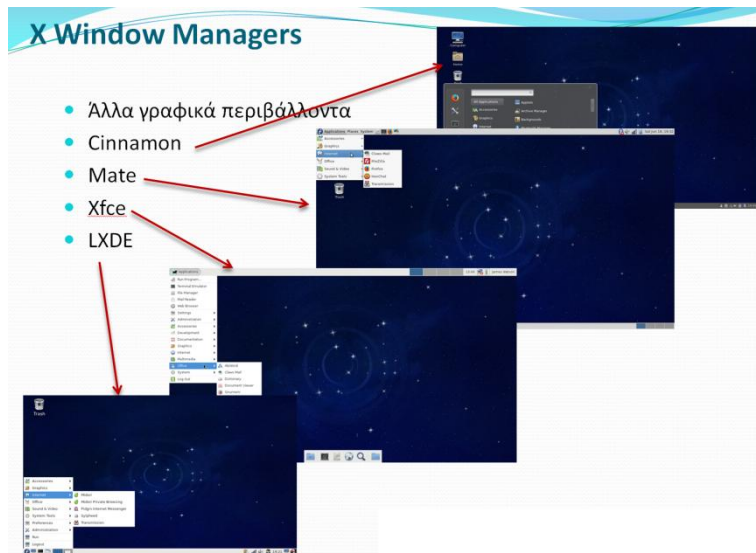
```
[root@ekdda ~]# dnf -y install @kde-desktop
[root@ekdda ~]# dnf -y install @xfce-desktop
[root@ekdda ~]# dnf -y install @hawaii-desktop
[root@ekdda ~]# dnf -y install @sugar-desktop
```

Μια εντολή για να βρει κάποιος τους διαθέσιμους X Window Managers (XWM) είναι η

```
[root@ekdda ~]# dnf grouplist -v
```

Με την παραπάνω εντολή εμφανίζονται όχι μόνο η διαθέσιμοι XWM αλλά και όλα τα grouplists όπως για παράδειγμα το Libre Office, Administration Tools κ.α.

Στην Εικόνα 4- 20 φαίνονται κάποιοι XWMs



Εικόνα 4- 20: Εικόνες από τα Desktops διαφόρων X Window Managers

Ο αναγνώστης μπορεί να βρει έναν γενικό κατάλογο μαζί με συγκριτικά στοιχεία στο (Wiki\_X\_Window\_Managers, n.d.).

Επίσης στην ίδια βιβλιογραφική αναφορά, ο αναγνώστης μπορεί να βρει συγκριτικά στοιχεία με τα προεγκατεστημένα πακέτα (λογισμικά) που έχει κάθε X Window Manager.

#### **Ασκήσεις - Δραστηριότητες**

Εγκαταστήστε διάφορους X Windows Managers σύμφωνα με τις εντολές της παραγράφου και δείτε τα διαφορετικά περιβάλλοντα και τι διαφορετικές εφαρμογές που έχει ο κάθε ένας από αυτούς.

### **4.3 Εγκατάσταση και Διαχείριση Samba Server**

Ο **Samba Server** (Sambaproject, n.d.) είναι εξυπηρετητής ο οποίος επιτρέπει το διαμοιρασμό αρχείων ανάμεσα σε δύο ή περισσότερους υπολογιστές ανεξαρτήτως λειτουργικού (Unix, Linux, Windows, MacOS).



Εικόνα 4- 21: Εμπορικό σήμα του Samba Server

Τα πρωτόκολλα στα οποία στηρίζεται η λειτουργία του είναι το SMB και το NMB. Ο Samba υποστηρίζει τα παρακάτω δικτυακά πρωτόκολλα και υπηρεσίες : NetBIOS, CIFS, MSRPC, WINS, NT Domain, Security Accounts Manager (SAM) Satabase, Local Security Authority (LSA) service, Active Directory Logon και άλλα.

Για τη λειτουργία του χρειάζεται να εγκατασταθούν τα εξής πακέτα :

- samba-common
- samba

Με το samba εγκαθίσταται ο Samba Server, ενώ με το πακέτο samba-common εγκαθίστανται επιπρόσθετα αρχεία τα οποία χρειάζονται και για τον Samba Server αλλά και για τους Samba Client. Ως προαιρετικά χαρακτηρίζονται τα παρακάτω πακέτα

- samba-Client
- system-config-samba

Το πακέτο samba-Client δίνει στο λειτουργικό, λειτουργικότητα ως προς την πλευρά του πελάτη (Client-side). Δηλαδή μπορεί το σύστημά μας να συνδεθεί με κάποιον samba Server. Το system-config-samba package παρέχει γραφική διεπαφή (graphical user interface - GUI) για την παραμετροποίηση του Samba Server. Προσοχή, το τελευταίο πακέτο δεν υποστηρίζεται από τη διανομή Fedora 25 και μετά.

#### 4.3.1 Εγκατάσταση

Η εγκατάσταση των πακέτων γίνεται με την παρακάτω εντολή

```
[root@ekdda ~]# dnf install samba samba-Client
```

#### 4.3.2 Παραμετροποίηση

Το βασικό αρχείο παραμετροποίησης είναι το smb.conf το οποίο βρίσκεται στην ακόλουθη διαδρομή

```
[root@ekdda ~]# vi /etc/samba/smb.conf
```

Οι αλλαγές και οι προσθήκες που πρέπει να γίνουν είναι οι ακόλουθες (FedoraWiki, Samba Server, n.d.) (ServerWorld, n.d.).

στη γραμμή 7: προσθέτουμε το όνομα με το οποίο θα φαίνεται μέσα από τον explorer του λειτουργικού συστήματος καθώς και την κωδικοποίηση των χαρακτήρων.

```
netbios name = LINUX SERVER  
unix charset = UTF-8
```

στη γραμμή 9: αλλάζουμε το όνομα του workgroup (εδώ προσθέτουμε το workgroup στο οποίο ανήκουν οι μηχανές με λειτουργικό Windows οι οποίες θέλουμε να έχουν πρόσβαση στον Samba Server).

```
workgroup = WORKGROUP
```

στη γραμμή 11: ορίζουμε τις διευθύνσεις IP που θα μπορούν να έχουν πρόσβαση στον Server. Στην παρακάτω γραμμή ορίζεται ότι πρόσβαση στον Server μας μπορούν να έχουν όλοι οι υπολογιστές που έχουν IPs της μορφής 127.x.y.z (εδώ απλά εννοούμε το μηχάνημα στο οποίο είναι εγκατεστημένος ο Server) όλοι οι υπολογιστές που έχουν IPs της μορφής 10.x.y.z καθώς και όλοι οι υπολογιστές που έχουν IPs της μορφής 192.168.y.z (εδώ προφανώς επιλέγουμε ως διαχειριστές τα υποδίκτυα (δείτε και παράγραφο 6.3) στα οποία θέλουμε να επιτρέψουμε την πρόσβαση στον Samba Server μας)

```
hosts allow = 127. 10. 192.168.
```

στη γραμμή 12: προσθέτουμε το ακόλουθο το οποίο σημαίνει ότι οι συνδέσεις χρήστη με μη έγκυρο κωδικό πρόσβασης απορρίπτονται, εκτός εάν δεν υπάρχει το όνομα χρήστη, οπότε αντιμετωπίζεται ως είσοδος επισκέπτη και αντιστοιχίζεται στον λογαριασμό επισκέπτη (guest account) (SambaOrgSmbConf, n.d.)

```
map to guest = Bad User
```

Στο τέλος του αρχείου θα πρέπει να προσθέσουμε τον ακόλουθο κώδικα, για κάθε διαφορετικό διαμοιρασμό καταλόγου που κάνουμε. Στο παρακάτω παράδειγμα διαμοιράζουμε τον κατάλογο

```
/home/samba_shared
```

```
# any Share name you like

[linux01_Share]

# shared directory
path = /home/samba_shared

# writeable and browseable
writeable = yes
browseable = yes

# guest ok
guest ok = yes

# guest only
guest only = yes

# user create permissions
create mode = 0644
```

Η εγγραφή

```
create mode = 0644
```

σημαίνει ότι ο χρήστης που έχει πρόσβαση στον κατάλογο μέσω Samba μπορεί να δημιουργήσει αρχεία με δικαιώματα 644. Δηλαδή rw για τον ιδιοκτήτη και r για το group και όλους τους άλλους. Να αναφέρουμε ότι τα default δικαιώματα δημιουργίας είναι 744.

Με την εγγραφή

```
guest ok = yes
```

δίνουμε τη δυνατότητα πρόσβασης στον χρήστη μέσω Samba (πχ ο χρήστης από λειτουργικό Windows) να έχει πρόσβαση στον διαμοιραζόμενο κατάλογο χωρίς να απαιτείται κάποιο password. Η παραπάνω εγγραφή συνδυάζεται και με την

```
guest only = yes
```

με την οποία καθορίζουμε ότι οι μόνες συνδέσεις στον Samba θα είναι μέσω του λογαριασμού επισκέπτη (guest account). Τα δικαιώματα (privileges) θα είναι αυτά του λογαριασμού επισκέπτη (SambaOrgSmbConf, n.d.).

Με την εγγραφή

```
writeable = yes  
browseable = yes
```

επιτρέπουμε στους χρήστες μέσω Samba (πχ οι χρήστες από λειτουργικό Windows) να προσπελαίνουν τον διαμοιραζόμενο φάκελο και να μπορούν να γράψουν σε αυτόν. Όμως θα πρέπει τα δικαιώματα του φακέλου όπως τα έχει ορίσει ο διαχειριστής του Linux Server να είναι τέτοια ώστε να επιτρέπεται η εγγραφή στον φάκελο από τους other users (πχ 755).



Η παραμετροποίηση του Samba Server που περιγράφουμε σε αυτήν την παράγραφο είναι μια πρόταση παραμετροποίησης με συγκεκριμένη φιλοσοφία. Φυσικά δεν είναι ο μοναδικός τρόπος. Κάθε διαχειριστής μπορεί να εφαρμόσει και άλλους τρόπους ανάλογα το σκεπτικό του.

Μια πλήρη επεξήγηση της παραμετροποίησης του smb.conf μπορεί ο αναγνώστης να βρει στο (SambaOrgSmbConf, n.d.).

Αφού τελειώσει η παραμετροποίηση του αρχείου

```
smb.conf
```

θα πρέπει να ξεκινήσει η τόσο η υπηρεσία smb όσο και η υπηρεσία nmb.

```
[root@ekdda ~]# systemctl start smb
[root@ekdda ~]# systemctl start nmb
```

Επιπλέον ως διαχειριστές μπορούμε να κάνουμε enable την υπηρεσία πράγμα που σημαίνει ότι η υπηρεσία θα ξεκινά με την εκκίνηση του λειτουργικού συστήματος

```
[root@ekdda ~]# systemctl enable smb
[root@ekdda ~]# systemctl enable nmb
```

Τέλος ως διαχειριστές πρέπει να ανοίξουμε την υπηρεσία στο firewall

```
[root@ekdda ~]# firewall-cmd --add-service=samba --permanent
[root@ekdda ~]# firewall-cmd --reload
```

Η παράμετρος permanent (όπως είδαμε) σημαίνει ότι η υπηρεσία θα είναι ανοικτή στο firewall μόνιμα.

Εάν το SELinux είναι ενεργό (περισσότερα για το SELinux στην παράγραφο 7.1.1), πρέπει να προστεθούν και οι επόμενες εντολές οι οποίες ενεργοποιούν τους κατάλληλους διακόπτες, ώστε να μπορούν τα αιτήματα για την υπηρεσία Samba να προσπερνούν την εμπλουτισμένη ασφάλεια που προσφέρει το συγκεκριμένο σύστημα.

```
[root@ekdda ~]# setsebool -P samba_enable_home_dirs on permanent
[root@ekdda ~]# setsebool -P samba_enable_home_dirs on
[root@ekdda ~]# restorecon -R /home/samba_shared
```

Εάν εκτελεσθούν οι προηγούμενες εντολές χρειάζεται επανεκκίνηση των υπηρεσιών smb και nmb.

```
[root@ekdda ~]# systemctl restart smb
[root@ekdda ~]# systemctl restart nmb
```

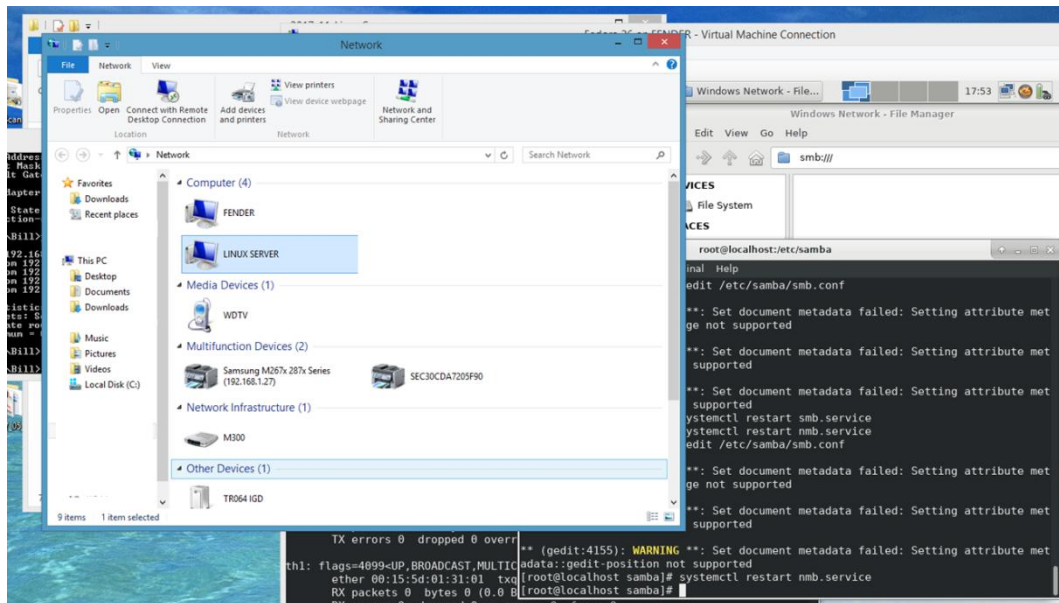
Με τις παρακάτω εντολές μπορεί τόσο ο διαχειριστής όσο και οι χρήστες του Server να δουν σε ποια κατάσταση βρίσκεται η λειτουργία των υπηρεσιών του Server.

```
[root@ekdda ~]# systemctl status smb nmb
[root@ekdda ~]# service smb status
[root@ekdda ~]# service nmb status
```

### 4.3.3 Χρήση

Στην παρούσα παράγραφο θα δούμε τι βλέπει ο χρήστης εφόσον έχει γίνει η παραμετροποίηση σύμφωνα με την παράγραφο 4.3.2

Ένας χρήστης των Windows από τον explorer των Windows στην επιλογή Network θα δει την παρακάτω Εικόνα 4- 22



Εικόνα 4- 22: Οθόνη των Windows που βλέπει ο χρήστης όταν ο Samba είναι ενεργοποιημένος

Εναλλακτικά μπορεί κάποιος να αναζητήσει τον Samba Server με την IP του μηχανήματος που τον φιλοξενεί, για παράδειγμα [\\192.168.1.49](http://192.168.1.49) στη γραμμή διευθύνσεων του explorer.

Στο παράθυρο του explorer εμφανίζεται το όνομα που έδωσε ο διαχειριστής στο τμήμα

```
# any Share name you like
```

Που αναφέρθηκε αναλυτικά πιο πάνω

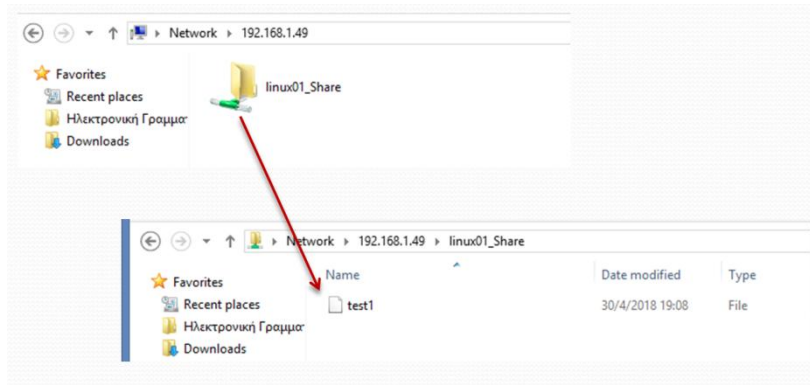
```
# any Share name you like

[linux01_Share]
.....
```

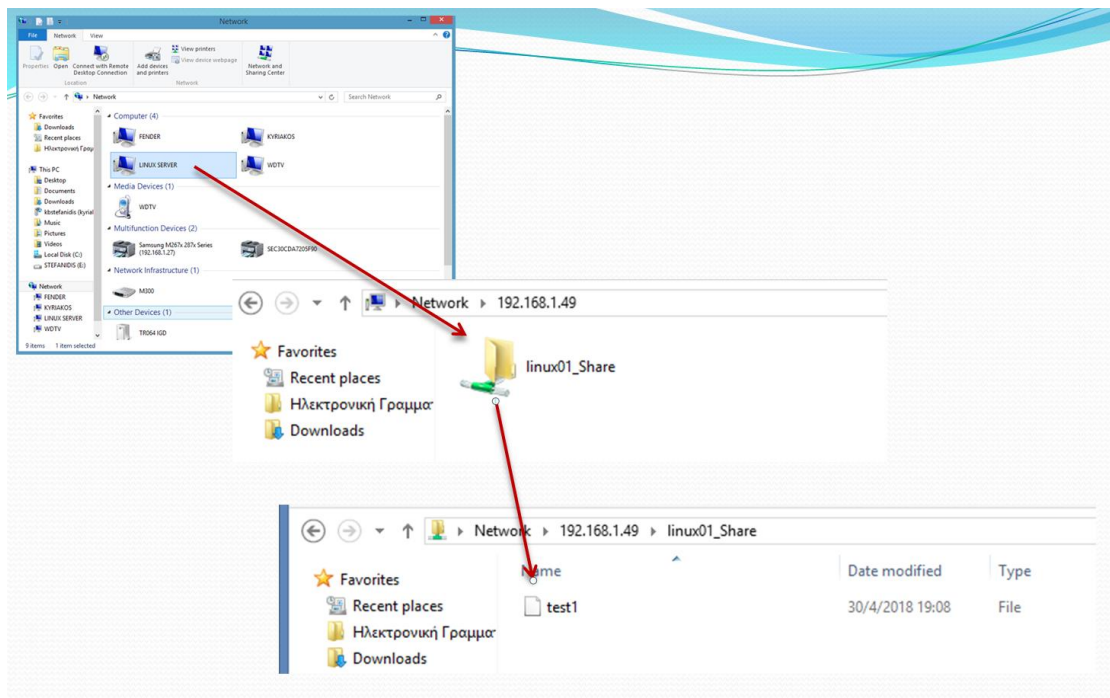
και το οποίο στην πραγματικότητα δείχνει στον διαμοιραζόμενο κατάλογο του Server

```
/home/samba_shared
```





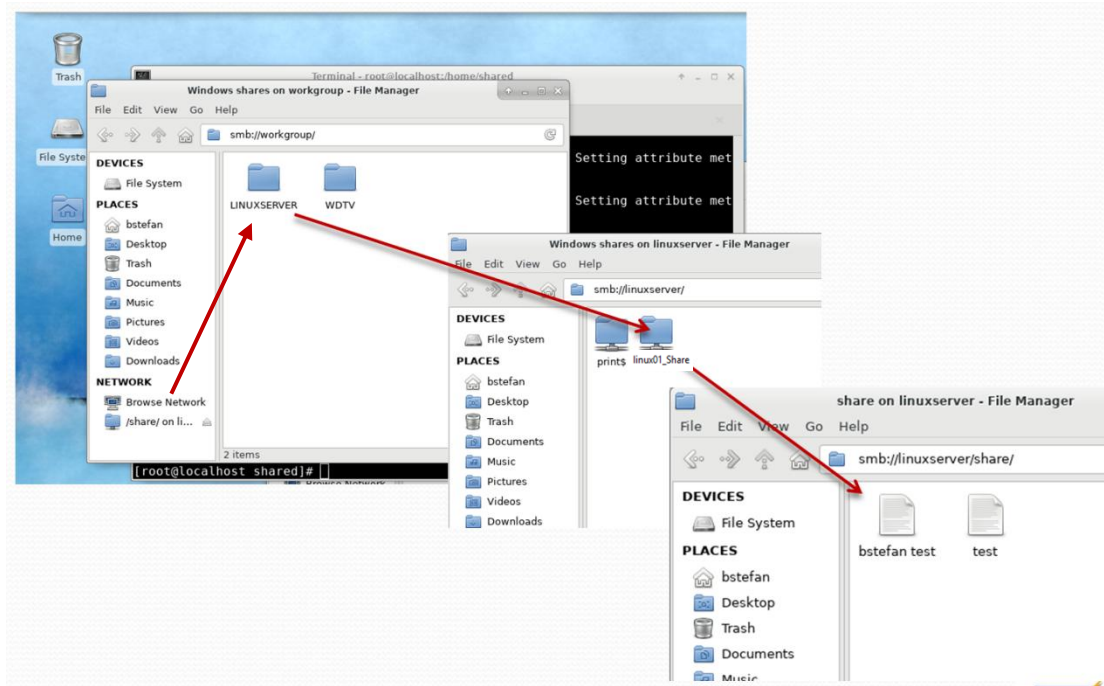
Εικόνα 4- 23: Εικόνα από τον explorer των Windows με το όνομα και το περιεχόμενο του διαμοιραζόμενου φακέλου



Εικόνα 4- 24: Συνολική εικόνα διαμοιραζόμενου αρχείου και φακέλου σε Windows

Ένας χρήστης Linux θα δει μια αντίστοιχη εικόνα στον «Nautilus» στην επιλογή Network (Εικόνα 4- 25).





Εικόνα 4- 25: Συνολική εικόνα διαμοιραζόμενου αρχείου και φακέλου σε Linux

### Ασκήσεις - Δραστηριότητες

- Πραγματοποιήστε την εγκατάσταση του Samba Server σύμφωνα με τις οδηγίες και εκτελέστε τις εντολές της παραγράφου και δείτε τα αποτελέσματα που φαίνονται στις εικόνες
- Διαμοιράστε δυο καταλόγους με διαφορετικά δικαιώματα
- Δημιουργήστε αρχεία με διαφορετικά δικαιώματα μέσα στους καταλόγους
- Ως χρήστες του Samba (πχ από τον explorer των windows) προσπαθήστε να αλλάξετε το περιεχόμενο ενός αρχείου που είναι read only και ενός που είναι read write
- Αλλάξτε τα δικαιώματα ενός καταλόγου σε read only και προσπαθήστε να αλλάξετε ένα αρχείο με δικαιώματα read write που βρίσκεται μέσα σε αυτόν.

## 5 Διαχείριση Υπηρεσιών και System Monitoring

Στις επόμενες παραγράφους θα δούμε τις υπηρεσίες σε έναν Linux Server και θέματα διαχείρισης και παρακολούθησής τους (monitoring).

### 5.1 Διεργασίες – Υπηρεσίες - Deamons - Sockets

**Υπηρεσίες ή Διεργασίες Εξυπηρετητή** είναι οι διεργασίες του λειτουργικού συστήματος οι οποίες προορίζονται να εξυπηρετήσουν τους χρήστες. Τοπικούς και απομακρυσμένους. Κάθε διεργασία δημιουργείται από την εκτέλεση κάποιου προγράμματος, μιας εφαρμογής ή προέρχεται από την εκτέλεση ενός μεγαλύτερου συνόλου οδηγιών και προγραμμάτων. Δεδομένου ότι το Linux είναι ένα λειτουργικό σύστημα πολλαπλής επεξεργασίας, οι διαδικασίες μπορούν να εκτελούνται ταυτόχρονα και ανεξάρτητα η μία από την άλλη. Κάθε διαδικασία έχει τα δικά της δικαιώματα, χώρο διευθύνσεων εικονικής μνήμης και ούτω καθεξής. Μια διαδικασία μπορεί να έχει πρόσβαση και να χρησιμοποιεί πόρους συστήματος όπως μνήμη, CPU, αρχεία στο σύστημα αρχείων και φυσικές συσκευές. Για τη διαχείριση και τον έλεγχο της πρόσβασης σε αυτούς τους πόρους, ο πυρήνας πρέπει να παρακολουθεί ποιες διαδικασίες εκτελούνται και ποιοι πόροι χρησιμοποιούνται. Ο πυρήνας το κάνει χρησιμοποιώντας μια δομή δεδομένων για να αντιπροσωπεύσει τις διαδικασίες. Αυτή η δομή δεδομένων είναι αρκετά μεγάλη και περίπλοκη.



**Δαίμονες** (από τον αγγλικό **daemons**) ονομάζονται οι διεργασίες που εκτελούνται στο παρασκήνιο (background) και δεν είναι στον άμεσο έλεγχο κάποιου χρήστη. Παραδοσιακά, τα ονόματα της διεργασίας - υπηρεσίας που είναι δαίμονας τελειώνουν με το γράμμα d, για να διευκρινίζεται ότι η διαδικασία είναι στην πραγματικότητα ένας δαίμονας και για τη διαφοροποίηση μεταξύ ενός δαίμονα και ενός κανονικού προγράμματος υπολογιστή. Για παράδειγμα η sshd είναι μια υπηρεσία - δαίμονας η οποία εξυπηρετεί αιτήματα σύνδεσης ασφαλούς κελύφους ssh

Πολλές φορές όμως ο όρος δαίμονας χρησιμοποιείται ανεξαιρέτως για όλες τις υπηρεσίες που είναι στη διάθεση του χρήστη.

Υπάρχουν διεργασίες οι οποίες χρειάζονται στο σύστημα για να λειτουργήσει, όπως η υπηρεσία διαχείρισης συστήματος **systemd**, η υπηρεσία διαχείρισης χρηστών αλλά και υπηρεσίες οι οποίες χρησιμοποιούνται από το σύστημα για να εξυπηρετήσουν δικτυακά και διαδικτυακά αιτήματα (υπάρχει εκτενής αναφορά στο κεφάλαιο 6), όπως για παράδειγμα ένα αίτημα ανάγνωσης ιστοσελίδας ή ένα αίτημα αποστολής αλληλογραφίας (FedoraDocsSystemdProcess, n.d.)

Η **systemd** (υπάρχει από τη Fedora 15 και έπειτα) είναι η πρώτη διεργασία που ξεκινά από τον πυρήνα (Kernel) του Linux Server κατά τη διαδικασία της εκκίνησης (Boot). Έχει αντικαταστήσει την πιο παλιά διεργασία SysVinit (γνωστή και ως init) και το νεότερο σύστημα init Upstart (FedoraDocsInitProcess, n.d.). Η systemd συντονίζει την υπόλοιπη διαδικασία εκκίνησης και διαμορφώνει το περιβάλλον για τον χρήστη. Ξεκινά τη διαδικασία φόρτωσης όλων των προγραμμάτων άμεσα και διαχειρίζεται πληροφορίες μεταξύ

αλληλεξαρτώμενων προγραμμάτων καθώς αυτά φορτώνονται στη μνήμη. Διαχωρίζοντας τα προγράμματα και τα μέσα επικοινωνίας τους, κάθε πρόγραμμα μπορεί να φορτωθεί στη μνήμη χωρίς να περιμένει πρώτα να φορτωθούν άσχετα ή ακόμη και εξαρτώμενα προγράμματα.



**Sockets** από την πλευρά των υπηρεσιών του συστήματος είναι συνδέσεις μεταξύ μιας υπηρεσίας του Linux Server μας και άλλων υπηρεσιών ή προγραμμάτων χρηστών. Επίσης είναι συνδέσεις με άλλα Sockets που διασυνδέουν άλλες υπηρεσίες και εφαρμογές.

Καθώς οι διεργασίες φορτώνονται στη μνήμη, συνδέονται σταδιακά στα διάφορα sockets τους για να λαμβάνουν μηνύματα αναμονής και να επικοινωνούν με άλλες διεργασίες και sockets.

Η systemd χειρίζεται εξαρτήσεις μεταξύ προγραμμάτων, αλλά δεν χρειάζεται προκαθορισμένη σειρά εκκίνησης. Τα προγράμματα και οι προτιμήσεις του χρήστη (διαχειριστή στην περίπτωση του Linux Server) φορτώνονται καθώς οι συσκευές και οι υπηρεσίες από τις οποίες εξαρτώνται γίνονται διαθέσιμες.

Στο τέλος όλης αυτής της διαδικασίας, παρουσιάζεται μια οθόνη σύνδεσης χρήστη.

## 5.2 Καταστάσεις διεργασιών

Ένα μεγάλο πρόβλημα που προκύπτει ειδικά όταν το υπολογιστικό σύστημα που φιλοξενεί τον Linux Server είναι για πολύ χρόνο σε λειτουργία είναι οι «ορφανές» διεργασίες και οι διεργασίες «ζόμπι». Οι «ορφανές» θεωρητικά δε δημιουργούν πρόβλημα. Όμως οι «ζόμπι» διεργασίες δεν κάνουν τίποτα αλλά φαίνονται ενεργές στο σύστημα. Υπάρχει μια μικρή ποσότητα μνήμης που χρησιμοποιείται για κάθε διαδικασία ζόμπι. Τεχνικά, θα χρειαστούν δεκάδες χιλιάδες, ίσως εκατοντάδες χιλιάδες διεργασίες ζόμπι για να προκαλέσουν σημαντική εξάντληση των πόρων του συστήματος. (RedhatOrphanZombiProcess, n.d.)

Η φυσιολογική πορεία τερματισμού ενός δέντρου διεργασιών είναι να τερματίζονται πρώτα τα «παιδιά» και μετά οι «γονείς». Ένα «παιδί» για να τερματιστεί πρέπει να πάρει τη συγκατάθεση του «γονέα» του. Ο τερματισμός είναι φυσιολογικός όταν φτάσει να τερματιστεί και η πρώτη διεργασία που δημιούργησε όλες τις άλλες και αφού έχουν τερματιστεί όλα τα «παιδιά» της με τη συγκατάθεσή της.

Τι γίνεται όμως όταν τερματιστεί μια διεργασία «γονέας» χωρίς να έχουν τερματισθεί πρώτα τα «παιδιά» της ; Επιπλέον τι γίνεται εάν τερματιστεί μια διεργασία «παιδί» χωρίς να έχει δώσει τη συγκατάθεση ο «γονέας» ;



η «συγκατάθεση» της διεργασίας «γονέας» για τον τερματισμό μιας διεργασίας παιδί είναι γνωστή ως κωδικός εξόδου (exit code).

Στην πρώτη περίπτωση που τερματίζεται ο «γονέας» χωρίς να έχουν τερματισθεί πρώτα τα «παιδιά» της, οι διεργασίες «παιδιά» μετατρέπονται σε ορφανές διεργασίες. Στη δεύτερη περίπτωση όπου ο «γονέας» δεν έδωσε τη συγκατάθεσή του για να τερματιστεί το «παιδί», το «παιδί» μετατρέπεται σε «ζόμπι». Οι «ορφανές» διεργασίες συνεχίζουν να θεωρούνται φυσιολογικές αφού αναλαμβάνει ως γονέας η «πρώτη των όλων» διεργασία η οποία όπως είδαμε είναι η systemd. Οι διεργασίες «ζόμπι» όμως θεωρούνται προβληματικές και πρέπει να τερματίζονται από το διαχειριστή.

Η κατάσταση των διεργασιών παρουσιάζεται στη στήλη «STAT» της εντολή ps που θα δούμε στην αμέσως επόμενη παράγραφο 5.3.1. Οι καταστάσεις που μπορεί να βρεθεί μια διεργασία είναι :

- **R** που σημαίνει ότι η διεργασία εκτελείται κανονικά,
- **S** που σημαίνει ότι βρίσκεται σε κατάσταση αναμονής,
- **Z** που σημαίνει ότι είναι σε κατάσταση "ζόμπι",
- **D** που σημαίνει ότι κλείνει μόνο με επανεκκίνηση του συστήματος και
- **T** που σημαίνει ότι έχει σταματήσει.

### 5.3 Διεργασίες Συστήματος και Εργαλεία Παρακολούθησης Διεργασιών (process monitoring)

Οι διεργασίες (FedoraDocsProcesses, n.d.) όπως είδαμε δεν είναι τίποτε άλλο από κώδικα που εκτελείται στο υπολογιστικό μας σύστημα (Linux Server). Κάθε φορά που ανοίγουμε μία εφαρμογή στον υπολογιστή μας, ξεκινούν διάφορες τέτοιες διεργασίες για να την εμφανίσουν στην οθόνη μας. Υπεύθυνος για τις διεργασίες αυτές είναι ο πυρήνας Linux (Linux kernel), ο οποίος κατά τη διάρκεια της εκκίνησης του συστήματος αλλά και κατά τη διάρκεια εκτέλεσης των εφαρμογών του συστήματος προσδίδει σε αυτές υπολογιστικούς πόρους. Δηλαδή φορτώνει τον κώδικά τους στη μνήμη και τους δίνει ένα μέρος του επεξεργαστή για να εκτελεστούν. Όταν φυσικά κλείσουν οι αντίστοιχες εφαρμογές, ο πυρήνας τερματίζει τις διεργασίες που την αφορούν και απελευθερώνει τη μνήμη και το ποσοστό του επεξεργαστή που καταλάμβαναν.

Προκειμένου να διαμορφώσουμε το σύστημά μας, ως διαχειριστές συστήματος πρέπει να παρακολουθούμε αυτούς τους υπολογιστικούς πόρους. Πόσοι χρησιμοποιούνται ή πόσο διαθέσιμοι είναι. Δηλαδή την ποσότητα της ελεύθερης μνήμης, τον διαθέσιμο ελεύθερο χώρο στο δίσκο, τον τρόπο κατάτμησης του σκληρού δίσκου ή ποιες διαδικασίες εκτελούνται. Επίσης το ποσοστό της CPU που χρησιμοποιούν.

Στην παράγραφο αυτή θα δούμε εργαλεία για την παρακολούθηση των διεργασιών κυρίως πού θα τις βρούμε, πώς μπορούμε να τις ξεχωρίσουμε, τι πληροφορίες μας δίνουν και πώς μπορούμε να τις τερματίσουμε («σκοτώσουμε») αν κάτι «δεν πάει καλά».

#### 5.3.1 Εντολή ps

Όπως ο κάθε χρήστης και η κάθε ομάδα χρηστών έχουν ένα μοναδικό αναγνωριστικό (UID και GID αντίστοιχα), έτσι και οι διεργασίες έχουν το δικό τους μοναδικό αναγνωριστικό με τη μορφή του PID (**P**rocess **I**dentification number – **Α**ριθμός **Α**ναγνώρισης **Δ**ιεργασίας). Ο

διαχειριστής μπορεί (και οφείλει) να ελέγχει συχνά τις διάφορες διεργασίες που εκτελούνται στο σύστημά του, είτε από τον πυρήνα, είτε από τον ίδιο (root) είτε από τους χρήστες. Για να δούμε τις διεργασίες με τα PID τους στο τερματικό, γράφουμε την εντολή:

```
[root@ekdda ~]# ps -aux | more
```

Η εκτέλεση της παραπάνω εντολής έχει αποτέλεσμα κάτι σαν το παρακάτω

```

bstefan@mail:~
File Edit View Terminal Tabs Help
bstefan@mail:~ x bstefan@mail:~ x bstefan@mail:~
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
--system --deserialize 24
root         1  0.4  0.1 30048  7948 ?        Ss   18:06   0:03 /usr/lib/systemd/sy
root         2  0.0  0.0     0     0 ?        S    18:06   0:00 [kthreadd]
root         3  0.0  0.0     0     0 ?        I    18:06   0:00 [kworker/0:0]
root         4  0.0  0.0     0     0 ?        I<   18:06   0:00 [kworker/0:0H]
root         5  0.0  0.0     0     0 ?        I    18:06   0:00 [kworker/u4:0]
root         6  0.0  0.0     0     0 ?        I<   18:06   0:00 [mm_percpu_wq]
root         7  0.1  0.0     0     0 ?        S    18:06   0:00 [ksoftirqd/0]
root         8  0.1  0.0     0     0 ?        I    18:06   0:00 [rcu_sched]
root         9  0.0  0.0     0     0 ?        I    18:06   0:00 [rcu_bh]
root        10  0.0  0.0     0     0 ?        S    18:06   0:00 [migration/0]
root        11  0.0  0.0     0     0 ?        S    18:06   0:00 [watchdog/0]
root        12  0.0  0.0     0     0 ?        S    18:06   0:00 [cpuhp/0]
root        13  0.0  0.0     0     0 ?        S    18:06   0:00 [cpuhp/1]
root        14  0.0  0.0     0     0 ?        S    18:06   0:00 [watchdog/1]
root        15  0.0  0.0     0     0 ?        S    18:06   0:00 [migration/1]
root        16  0.0  0.0     0     0 ?        S    18:06   0:00 [ksoftirqd/1]
root        18  0.0  0.0     0     0 ?        I<   18:06   0:00 [kworker/1:0H]
root        19  0.0  0.0     0     0 ?        S    18:06   0:00 [kdevtmpfs]
root        20  0.0  0.0     0     0 ?        I<   18:06   0:00 [netns]
root        21  0.0  0.0     0     0 ?        S    18:06   0:00 [rcu_tasks_kthre]
root        22  0.0  0.0     0     0 ?        S    18:06   0:00 [kauditd]
--More--

```

Εικόνα 5- 1: Έξοδος της εντολής ps

Στην παραπάνω Εικόνα 5- 1 μπορούμε να δούμε μια λίστα με ένα στιγμιότυπο της κατάστασης που βρίσκονται οι υπηρεσίες. Υπάρχουν διάφορες ετικέτες που δίνουν πληροφορίες για τις υπηρεσίες :

- η ετικέτα **USER** δείχνει σε ποιον χρήστη ανήκει η διεργασία
- η ετικέτα **PID** δείχνει τον μοναδικό αριθμό αναγνώρισης της διεργασίας
- η ετικέτα **%CPU** δείχνει το ποσοστό χρήσης του επεξεργαστή από τη διεργασία
- η ετικέτα **%MEM** δείχνει το ποσοστό χρήσης της μνήμης από τη διεργασία
- η ετικέτα **VSZ** δείχνει το μέγεθος που καταλαμβάνει η διεργασία στην εικονική μνήμη σε KB
- η ετικέτα **RSS** δείχνει το μέγεθος που καταλαμβάνει η διεργασία στην non-swapped φυσική μνήμη σε KB
- η ετικέτα **TTY** δείχνει το τερματικό στο οποίο εκτελείται η διεργασία
- η ετικέτα **STAT** είναι ο κωδικός κατάστασης της διεργασίας. Την είδαμε πιο αναλυτικά στην παράγραφο **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.**, όπου μιλάμε για τις «ορφανές» διεργασίες και τις διεργασίες «ζόμπι»
- η ετικέτα **START** δείχνει πότε ξεκίνησε η διεργασία

- η ετικέτα **TIME** δείχνει πόση ώρα εκτελείται η διεργασία
- η ετικέτα **COMMAND** που σχετίζεται με το όνομα του εκτελέσιμου αρχείου, script ή της εντολής με την οποία ξεκινά η διεργασία

Αν θέλουμε να δούμε τις διεργασίες που τρέχει κάποιος συγκεκριμένος χρήστης μπορούμε να γράψουμε την εντολή

```
[root@ekdda ~]# ps -aux | grep <user> | more
```

Στη θέση του <user> μπορεί να είναι ο οποιοσδήποτε χρήστης του συστήματος. Η έξοδος που παίρνουμε σε μια τέτοια περίπτωση είναι η ακόλουθη Εικόνα 5- 2:

```
[root@mail ~]# ps aux | grep bstefan | more
bstefan 1741 0.0 0.1 20164 7588 ? Ss 18:09 0:00 /usr/lib/systemd/systemd --user
bstefan 1743 0.0 0.0 55932 1676 ? S 18:09 0:00 (sd-pam)
bstefan 1758 0.0 0.2 48684 10148 tty2 Ssl+ 18:09 0:00 /usr/libexec/gdm-x-session --run-script
startxfce4
bstefan 1760 0.3 1.3 156424 55120 tty2 Sl+ 18:09 0:04 /usr/libexec/Xorg vt2 -displayfd 3 -auth
/run/user/1000/gdm/Xauthority -nolisten tcp -background none -noreset -keeptty -verbose 3
bstefan 1780 0.0 0.1 25496 4972 ? Ss 18:09 0:00 /usr/bin/dbus-daemon --session --address
=systemd: --nofork --nopidfile --systemd-activation --syslog-only
bstefan 1783 0.0 0.0 5788 3156 tty2 S+ 18:09 0:00 /bin/sh /etc/xdg/xfce4/xinitrc -- vt
bstefan 1799 0.0 0.0 10700 532 ? Ss 18:09 0:00 /usr/bin/ssh-agent /bin/sh -c exec -l /b
in/bash -c "startxfce4"
bstefan 1835 0.0 0.3 60656 15524 tty2 Sl+ 18:09 0:00 xfce4-session
bstefan 1839 0.0 0.1 9996 5136 ? S 18:09 0:00 /usr/lib/xfce4/xfconf/xfconfd
bstefan 1842 0.0 0.0 24460 536 ? Ss 18:09 0:00 /usr/bin/gpg-agent --sh --daemon --write
-env-file /home/bstefan/.cache/gpg-agent-info
bstefan 1843 0.0 0.4 39656 20164 tty2 S+ 18:09 0:00 xfwm4 --display :0.0 --sm-client-id 2bc8
6bb35-cf2f-4938-9008-750189441075
bstefan 1844 0.0 0.5 61876 24100 tty2 Sl+ 18:09 0:00 xfce4-panel --display :0.0 --sm-client-i
d 2c8fe8e5c-ef01-45e7-8c94-fe76e42a7b20
bstefan 1852 0.0 0.4 58416 16824 ? Ssl 18:09 0:00 xfsettingsd --display :0.0 --sm-client-i
d 27ef937d5-d07f-4a40-96e8-4a00bc9cec
```

Εικόνα 5- 2: Εμφάνιση διεργασιών χρήστη με την ps -aux

Μια ακόμη χρήσιμη εκτέλεση της ps είναι με τις παραμέτρους -ejf

```
[root@ekdda ~]# ps -ejf | more
```

Με την οποία εκτός από το **PID** λαμβάνουμε και το **PPID** (Parent Process ID), το οποίο μας δείχνει τον γονέα της διεργασίας. Η πληροφορία είναι πολύ χρήσιμη σε περίπτωση που θέλουμε να σταματήσουμε μια διεργασία, μπορούμε να δούμε από ποια εξαρτάται. Στην παρακάτω Εικόνα 5- 3 βλέπουμε την έξοδο της παραπάνω εντολής

```
[root@ekdda ~]# ps -ejf | more
UID      PID PPID PGID  SID  C STIME TTY      TIME CMD
root      1   0    1    1   0 18:06 ?        00:00:03 /usr/lib/systemd/sys
temd --switched-root --system --deserialize 24
root      2   0    0    0   0 18:06 ?        00:00:00 [kthreadd]
root      4   2    0    0   0 18:06 ?        00:00:00 [kworker/0:0H]
root      6   2    0    0   0 18:06 ?        00:00:00 [mm_percpu_wq]
root      7   2    0    0   0 18:06 ?        00:00:01 [ksoftirqd/0]
root      8   2    0    0   0 18:06 ?        00:00:02 [rcu_sched]
root      9   2    0    0   0 18:06 ?        00:00:00 [rcu_bh]
root     10   2    0    0   0 18:06 ?        00:00:00 [migration/0]
```

Εικόνα 5- 3: Εμφάνιση διεργασιών χρήστη με την ps -ejf



### Ασκήσεις - Δραστηριότητες

Εκτελέστε τις εντολές της παραγράφου και δείτε τα αποτελέσματα που φαίνονται στις εικόνες

#### 5.3.2 Εντολή pstree

Μια πιο γραφική εντολή για να δούμε το «γενεαλογικό δέντρο» των διεργασιών είναι η εντολή **pstree**

```
[root@ekdda ~]# pstree | less
```

Με την εντολή **pstree** μπορούμε να έχουμε μια άμεση γραφική εποπτεία των εξαρτήσεων κάθε διεργασίας. Δηλαδή ποια είναι η διεργασία «γονέας» της, καθώς και ποια διεργασία έχει αυτήν ως «γονέα». Η εντολή (παραμέτρος εδώ) **less** επιτρέπει την κίνηση πάνω-κάτω με τα βελάκια του πληκτρολογίου καθώς και τη χρήση του **PageUp** και **PageDown**. Στην παρακάτω Εικόνα 5- 4 βλέπουμε την έξοδο της παραπάνω εντολής

```
systemd--ModemManager--{gdbus}
|   `--{gmain}
|   -NetworkManager--dhclient
|   |   -{gdbus}
|   |   `--{gmain}
|   -abrt-dbus--{gdbus}
|   |   `--{gmain}
|   -3*[abrt-dump-journ]
|   -abrt-d--{gdbus}
|   |   `--{gmain}
|   -accounts-daemon--{gdbus}
|   |   `--{gmain}
```

Εικόνα 5- 4: Έξοδος της εντολής **pstree**

Στην : Έξοδος της εντολής **pstree**. Μπορούμε να δούμε πολύ καθαρά αυτό που περιγράψαμε στην παράγραφο 5.1. Ότι η διεργασία **systemd** είναι ο γονέας όλων των διεργασιών. Επίσης μπορούμε να δούμε ότι η διεργασία **NetworkManager** έχει γονέα τη **systemd** και έχει «παιδί» την **dhClient**

### Ασκήσεις - Δραστηριότητες

Εκτελέστε τις εντολές της παραγράφου και δείτε τα αποτελέσματα που φαίνονται στις εικόνες

#### 5.3.3 Εντολή top

Το μειονέκτημα της εντολής **ps** είναι ότι εκτελώντας την, βλέπουμε μια λίστα με ένα στιγμιότυπο των διεργασιών με την κατάσταση που βρίσκονταν τη στιγμή που εκτελέσαμε την εντολή.

Εκτός από τη στατική κατάσταση που βλέπουμε με την εντολή **ps** μπορούμε να έχουμε και μια πιο δυναμική εποπτεία των διεργασιών με την εντολή **top** (από το **table of processes**).

```
[root@ekdda ~]# top
```

Με την εντολή `top` έχουμε μια «ζωντανή» αναπαράσταση της λίστας των διεργασιών. Βλέπουμε σε πραγματικό χρόνο πώς μεταβάλλουν οι διεργασίες την κατάστασή τους και πώς μεταβάλλεται η απαίτησή τους σε υπολογιστικούς πόρους κάθε δευτερόλεπτο. Οι δυο παρακάτω εικόνες έχουν παρθεί με διαφορά μερικών δευτερολέπτων

```
top - 18:43:54 up 37 min, 1 user, load average: 0.07, 0.04, 0.14
Tasks: 250 total, 1 running, 207 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.3 us, 1.3 sy, 0.0 ni, 96.6 id, 0.0 wa, 1.5 hi, 0.2 si, 0.0 st
KiB Mem : 4133340 total, 2460488 free, 696916 used, 975936 buff/cache
KiB Swap: 2097148 total, 2097148 free, 0 used. 3286808 avail Mem
```

| PID         | USER        | PR        | NI       | VIRT         | RES         | SHR         | S        | %CPU       | %MEM       | TIME+          | COMMAND        |
|-------------|-------------|-----------|----------|--------------|-------------|-------------|----------|------------|------------|----------------|----------------|
| 1760        | bstefan     | 20        | 0        | 156424       | 55120       | 27896       | S        | 1.7        | 1.3        | 0:05.11        | Xorg           |
| 3138        | root        | 20        | 0        | 0            | 0           | 0           | I        | 1.3        | 0.0        | 0:09.66        | kworker/0:1    |
| 1844        | bstefan     | 20        | 0        | 61876        | 24100       | 21036       | S        | 0.3        | 0.6        | 0:00.40        | xfce4-panel    |
| 1886        | bstefan     | 20        | 0        | 181424       | 52612       | 42968       | S        | 0.3        | 1.3        | 0:01.51        | xfce4-terminal |
| 3382        | root        | 20        | 0        | 0            | 0           | 0           | I        | 0.3        | 0.0        | 0:00.03        | kworker/1:2    |
| <b>3490</b> | <b>root</b> | <b>20</b> | <b>0</b> | <b>16896</b> | <b>3948</b> | <b>3560</b> | <b>R</b> | <b>0.3</b> | <b>0.1</b> | <b>0:00.04</b> | <b>top</b>     |
| 1           | root        | 20        | 0        | 30048        | 7948        | 6428        | S        | 0.0        | 0.2        | 0:03.22        | systemd        |
| 2           | root        | 20        | 0        | 0            | 0           | 0           | S        | 0.0        | 0.0        | 0:00.01        | kthreadd       |

Εικόνα 5- 5: Εικόνα από την εντολή `top`

```
top - 18:44:30 up 38 min, 1 user, load average: 0.04, 0.03, 0.13
Tasks: 250 total, 1 running, 207 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.2 us, 1.2 sy, 0.0 ni, 97.7 id, 0.0 wa, 0.8 hi, 0.2 si, 0.0 st
KiB Mem : 4133340 total, 2460612 free, 696780 used, 975948 buff/cache
KiB Swap: 2097148 total, 2097148 free, 0 used. 3286932 avail Mem
```

| PID  | USER | PR | NI  | VIRT  | RES  | SHR  | S | %CPU | %MEM | TIME+   | COMMAND      |
|------|------|----|-----|-------|------|------|---|------|------|---------|--------------|
| 3138 | root | 20 | 0   | 0     | 0    | 0    | I | 2.0  | 0.0  | 0:10.30 | kworker/0:1  |
| 1    | root | 20 | 0   | 30048 | 7948 | 6428 | S | 0.0  | 0.2  | 0:03.22 | systemd      |
| 2    | root | 20 | 0   | 0     | 0    | 0    | S | 0.0  | 0.0  | 0:00.01 | kthreadd     |
| 4    | root | 0  | -20 | 0     | 0    | 0    | I | 0.0  | 0.0  | 0:00.00 | kworker/0:0H |
| 5    | root | 20 | 0   | 0     | 0    | 0    | I | 0.0  | 0.0  | 0:00.12 | kworker/u4:0 |
| 6    | root | 0  | -20 | 0     | 0    | 0    | I | 0.0  | 0.0  | 0:00.00 | mm_percpu_wq |
| 7    | root | 20 | 0   | 0     | 0    | 0    | S | 0.0  | 0.0  | 0:00.95 | ksoftirqd/0  |

Εικόνα 5- 6: Εικόνα από την εντολή `top` μετά από μερικά δευτερόλεπτα

Μπορούμε να διακρίνουμε ότι υπάρχει διαφορά στη σειρά με την οποία εμφανίζονται οι διεργασίες ανάλογα με το ποσοστό της CPU που καταλαμβάνουν. Μπορούμε να ταξινομήσουμε τις διεργασίες στην οθόνη `top` ως προς το ποσοστό χρήσης της CPU πιέζοντας το συνδυασμό `shift+P` οπότε βλέπουμε κάτι τέτοιο (Εικόνα 5- 7) :

```
top - 18:56:54 up 50 min, 1 user, load average: 0.03, 0.01, 0.05
Tasks: 250 total, 1 running, 207 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.3 us, 0.8 sy, 0.0 ni, 97.5 id, 0.0 wa, 1.2 hi, 0.2 si, 0.0 st
KiB Mem : 4133340 total, 2456432 free, 697968 used, 978940 buff/cache
KiB Swap: 2097148 total, 2097148 free, 0 used. 3286140 avail Mem
scroll coordinates: y = 1/250 (tasks), x = 1/12 (fields)
```

| PID  | USER    | PR | NI | VIRT   | RES    | SHR   | S | %CPU | %MEM | TIME+   | COMMAND        |
|------|---------|----|----|--------|--------|-------|---|------|------|---------|----------------|
| 1760 | bstefan | 20 | 0  | 156424 | 55120  | 27896 | S | 2.0  | 1.3  | 0:06.68 | Xorg           |
| 1886 | bstefan | 20 | 0  | 181424 | 52612  | 42968 | S | 0.7  | 1.3  | 0:02.29 | xfce4-terminal |
| 678  | root    | 20 | 0  | 2484   | 560    | 508   | S | 0.3  | 0.0  | 0:00.40 | rngd           |
| 1900 | bstefan | 20 | 0  | 810288 | 252088 | 99888 | S | 0.3  | 6.1  | 0:18.71 | firefox        |
| 3138 | root    | 20 | 0  | 0      | 0      | 0     | I | 0.3  | 0.0  | 0:14.28 | kworker/0:1    |
| 1    | root    | 20 | 0  | 30048  | 7948   | 6428  | S | 0.0  | 0.2  | 0:03.31 | systemd        |

Εικόνα 5- 7: Ταξινόμηση της `top` ως προς το ποσοστό χρήσης της CPU



ή μπορούμε να ταξινομήσουμε ως προς το ποσοστό χρήσης της Μνήμης πιέζοντας το συνδυασμό shift+M οπότε βλέπουμε κάτι τέτοιο (Εικόνα 5- 8) :

```
top - 18:56:01 up 49 min, 1 user, load average: 0.08, 0.02, 0.06
Tasks: 253 total, 3 running, 208 sleeping, 0 stopped, 0 zombie
%Cpu(s): 18.2 us, 4.7 sy, 0.0 ni, 70.3 id, 0.0 wa, 6.2 hi, 0.5 si, 0.0 st
KiB Mem : 4133340 total, 2439172 free, 715276 used, 978892 buff/cache
KiB Swap: 2097148 total, 2097148 free, 0 used. 3268836 avail Mem
scroll coordinates: y = 1/253 (tasks), x = 1/12 (fields)
  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM    TIME+  COMMAND
 1900 bstefan   20   0  810288 252088 99888 S   0.0   6.1   0:18.65 firefox
 1205 gdm       20   0  891100 167124 84940 S   0.0   4.0   0:11.64 gnome-shell
 1018 mysql     20   0  737560 101032 17968 S   0.0   2.4   0:06.03 mysqld
   842 root      20   0  248276 60044  52768 S   0.0   1.5   0:00.70 httpd
 1760 bstefan   20   0  156424 55120  27896 S   0.3   1.3   0:06.51 Xorg
 1886 bstefan   20   0  181424 52612  42968 S   0.0   1.3   0:02.20 xfce4-terminal
 1905 bstefan   20   0  141788 43764  37432 S   0.0   1.1   0:00.39 xfce4-power-man
 1990 bstefan   20   0  238640 40700  34676 S   0.0   1.0   0:00.47 evolution-alarm
 1533 gdm       20   0  131184 38560  32696 S   0.0   0.9   0:00.22 ibus-x11
```

Εικόνα 5- 8: ταξινόμηση της top ως προς το ποσοστό χρήσης της Μνήμης

Όπως και με την ps έτσι και με την top υπάρχουν διάφορες ετικέτες που δίνουν πληροφορίες για τις υπηρεσίες :

- η ετικέτα **PID** δείχνει τον μοναδικό αριθμό αναγνώρισης της διεργασίας
- η ετικέτα **USER** δείχνει σε ποιον χρήστη ανήκει η διεργασία
- η ετικέτα **PR** δείχνει την προτεραιότητα της διεργασίας
- η ετικέτα **NI** δείχνει τη **nice value** της διεργασίας
- η ετικέτα **VIRT** δείχνει την τελική εικονική μνήμη που καταλαμβάνει η διεργασία
- η ετικέτα **RES** δείχνει τη συνολική μνήμη RAM που καταλαμβάνει η διεργασία
- η ετικέτα **SHR** δείχνει τη συνολική διαμοιραζόμενη μνήμη που καταλαμβάνει η διεργασία
- η ετικέτα **%CPU** δείχνει το ποσοστό χρήσης της CPU κάθε στιγμή
- η ετικέτα **%MEM** δείχνει το ποσοστό χρήσης της Μνήμης RAM κάθε στιγμή
- η ετικέτα **TIME+** δείχνει τη συνολικό χρόνο που απασχολεί τη CPU



Από την οθόνη top βγαίνουμε πατώντας το **q**

### Ασκήσεις - Δραστηριότητες

Εκτελέστε τις εντολές της παραγράφου και δείτε τα αποτελέσματα που φαίνονται στις εικόνες

#### 5.3.4 Εντολή htop

Μια πιο γραφική και πιο ευχάριστη έκδοση της top είναι η htop. Η htop συνήθως χρειάζεται εγκατάσταση πρώτα. Η εγκατάσταση και η εκκίνησή της γίνεται με τις εντολές

```
[root@ekdda ~]# dnf -y install htop
[root@ekdda ~]# htop
```

Η εφαρμογή htop έχει την ακόλουθη διεπαφή :

```

1  [|||||]          5.3%]   Tasks: 175, 406 thr; 1 running
2  [|||||]          4.6%]   Load average: 0.07 0.05 0.01
Mem[|||||]         785M/3.94G] Uptime: 01:47:55
Swp[|]             0K/2.00G]

PID USER      PRI  NI  VIRT   RES   SHR  S  CPU% MEM%   TIME+  Command
5186 root        20   0 14332  3924  3372  R  0.7  0.1   0:00.30 htop
1760 bstefan     20   0 167M  56504 30052  S  0.0  1.4   0:12.89 /usr/libexec/Xorg
4190 bstefan     20   0 122M  35136 28640  S  0.0  0.9   0:01.66 /usr/bin/xfce4-te
1143 apache      20   0 292M  21860 14564  S  0.0  0.5   0:01.01 /usr/sbin/httpd -
1781 bstefan     20   0 167M  56504 30052  S  0.0  1.4   0:01.32 /usr/libexec/Xorg
1873 bstefan     20   0 74264 27760 21348  S  0.0  0.7   0:01.99 xfdesktop --displ
1132 apache      20   0 292M  21860 14564  S  0.0  0.5   0:01.00 /usr/sbin/httpd -
2469 bstefan     20   0 791M  247M  99888  S  0.0  6.1   0:02.45 /usr/lib/firefox/
1900 bstefan     20   0 791M  247M  99888  S  0.0  6.1   0:24.22 /usr/lib/firefox/
1075 apache      20   0 421M  21916 14620  S  0.0  0.5   0:00.98 /usr/sbin/httpd -
1110 apache      20   0 292M  21856 14560  S  0.0  0.5   0:00.99 /usr/sbin/httpd -
 720 root        20   0 20664  9232  7964  S  0.0  0.2   0:01.06 /usr/libexec/sssd
2434 bstefan     20   0 791M  247M  99888  S  0.0  6.1   0:02.99 /usr/lib/firefox/
1196 mysql       20   0 720M   99M 17968  S  0.0  2.5   0:00.22 /usr/libexec/mysq
1142 apache      20   0 292M  21860 14564  S  0.0  0.5   0:01.01 /usr/sbin/httpd -
1124 apache      20   0 292M  21856 14560  S  0.0  0.5   0:00.97 /usr/sbin/httpd -
F1Help  F2Setup  F3Search F4Filter F5Tree   F6SortBy F7Nice  -F8Nice +F9Kill  F10Quit

```

Εικόνα 5- 9: Διεπαφή της εφαρμογής htop

Η htop είναι μια πιο εξελιγμένη top, παρέχει και αυτή δυναμική αναπαράσταση των υπολογιστικών πόρων και επίσης έχει γραφική αναπαράσταση της χρήσης των επεξεργαστών (ξεχωριστή απεικόνιση για κάθε έναν), γραφική αναπαράσταση της χρήσης της μνήμης RAM και της εικονικής μνήμης (SWAP Memory)

Στην Εικόνα 5- 9 παρατηρούμε ότι η htop διαθέτει μενού επιλογών εμφανές στη βάση του παραθύρου της. Το μενού λειτουργεί με τα function keys (F1, F2, ..., F10)

### Ασκήσεις - Δραστηριότητες

Πραγματοποιήστε την εγκατάσταση της εφαρμογής σύμφωνα με τις οδηγίες και εκτελέστε τις εντολές της παραγράφου και δείτε τα αποτελέσματα που φαίνονται στις εικόνες

#### 5.3.5 Διεργασίες και Τερματικό

Οι διεργασίες εκτός από το ότι δημιουργούνται και ανήκουν σε κάποιον χρήστη, ανήκουν και στο τερματικό στο οποίο εκτελούνται. Αν για παράδειγμα εκτελέσουμε μια εντολή «wget» για να κατεβάσουμε ένα αρχείο από το Internet και κλείσουμε το τερματικό πριν ολοκληρωθεί, η διεργασία θα διακοπεί μαζί του.

Στην εντολή **ps** είδαμε την ετικέτα **TTY** η οποία μας ενημερώνει σε ποιο τερματικό τρέχει η κάθε διεργασία. Στην παρακάτω Εικόνα 5- 10 (η οποία προήλθε από την εκτέλεση μιας **ps aux** που είδαμε στην παράγραφο 5.3.1) μπορούμε να δούμε τη στήλη TTY η οποία έχει διάφορες ενδείξεις (ερωτηματικό «?», pts, tty κτλ)

Όσες γράφουν "TTY#" τρέχουν σε κάποιο «κανονικό» τερματικό (regular terminal). Αν για παράδειγμα η ένδειξη είναι tty1, τότε η διεργασία εκτελείται στο πρώτο τερματικό του συστήματος στο οποίο μπορούμε να μπούμε πατώντας Ctrl+Alt+F1.

Ο συνδυασμός CTRL+ALT+<Function Number> εναλλάσσει μεταξύ των διαφορετικών tty sessions. Το πρωτεύον tty (αυτό στο οποίο γίνεται η εκκίνηση) είναι το tty1. Υπάρχουν άλλα πέντε τερματικά στα οποία μπαίνουμε με τα αντίστοιχα "F" πλήκτρα, (από Ctrl+Alt+F2 έως Ctrl+Alt+F6) (FedoraDocsTty, n.d.) ενώ για να επανέλθουμε στο περιβάλλον στο οποίο ξεκινά το σύστημα πατάμε Ctrl+Alt+F1. Αυτά τα έξι λέγονται «κανονικά» τερματικά (regular terminals). Εκτός από τα «κανονικά τερματικά», υπάρχουν και αυτά που χρησιμοποιούμε ανοίγοντας την εφαρμογή «Τερματικό» στον υπολογιστή μας, τα οποία ονομάζονται "ψευδοτερματικά" (pseudoterminals). Όταν στην ετικέτα TTY υπάρχει pts/# τότε η διεργασία εκτελείται σε κάποιο τέτοιο pseudoterminal (Εικόνα 5- 10).

Όταν στην ετικέτα TTY υπάρχει ερωτηματικό («?») τότε η διεργασία δε σχετίζεται με κανένα τερματικό. Αυτές οι διεργασίες είναι συνήθως οι απολύτως απαραίτητες για να τρέχει το λειτουργικό μας σύστημα και σε καμία περίπτωση δεν πρέπει να τις πειράζουμε.

| USER    | PID  | %CPU | %MEM | VSZ    | RSS   | TTY   | STAT | START | TIME | COMMAND                 |
|---------|------|------|------|--------|-------|-------|------|-------|------|-------------------------|
| root    | 2955 | 0.0  | 0.0  | 0      | 0     | ?     | I    | 18:25 | 0:00 | [kworker/u4:1]          |
| bstefan | 4190 | 0.0  | 0.8  | 126748 | 36016 | tty2  | Sl+  | 19:14 | 0:03 | /usr/bin/xfce4-terminal |
| bstefan | 4194 | 0.0  | 0.1  | 14848  | 4220  | pts/0 | Ss   | 19:14 | 0:00 | bash                    |
| root    | 4229 | 0.0  | 0.1  | 40876  | 7912  | pts/0 | S    | 19:14 | 0:00 | su -                    |
| root    | 4234 | 0.0  | 0.1  | 14880  | 4312  | pts/0 | S    | 19:14 | 0:00 | -bash                   |
| root    | 4619 | 0.0  | 0.0  | 0      | 0     | ?     | I    | 19:28 | 0:03 | [kworker/1:0]           |
| postfix | 5068 | 0.0  | 0.1  | 67184  | 7332  | ?     | S    | 19:47 | 0:00 | pickup -l -t unix -u    |
| root    | 5347 | 0.1  | 0.0  | 0      | 0     | ?     | I    | 20:00 | 0:05 | [kworker/0:1]           |
| bstefan | 5481 | 0.0  | 0.1  | 14848  | 4240  | pts/1 | Ss   | 20:05 | 0:00 | bash                    |
| root    | 5839 | 0.0  | 0.0  | 0      | 0     | ?     | I    | 20:20 | 0:00 | [kworker/u4:3]          |

Εικόνα 5- 10: Παράδειγμα σύνδεσης διεργασιών - τερματικών

### Ασκήσεις - Δραστηριότητες

Εκτελέστε τις εντολές της παραγράφου και δείτε τα αποτελέσματα που φαίνονται στις εικόνες

#### 5.3.6 Τερματισμός διεργασιών (εντολή Kill)

Πολλές φορές ο διαχειριστής είναι αναγκασμένος να τερματίσει μια διεργασία η οποία του δημιουργεί πρόβλημα. Ο τερματισμός αυτός στις περιπτώσεις που διαπιστώνουμε ότι η διεργασία δημιουργεί πρόβλημα (είτε πρόκειται για κάποια ύποπτη διεργασία, είτε για κάποια «ζόμπι» διεργασία) γίνεται βίαια. Η διαδικασία ονομάζεται και «σκότωμα» της διεργασίας εξαιτίας της εντολής που χρησιμοποιείται. Η εντολή αυτή είναι η **Kill**.

Για παράδειγμα αν η διεργασία με PID 2375 μας δημιουργεί πρόβλημα και θέλουμε να την τερματίσουμε γράφουμε στο τερματικό την ακόλουθη εντολή

```
[root@ekdda ~]# kill -9 2375
```

Οι κωδικοί αριθμοί για την εντολή kill (στην παραπάνω εντολή ο 9) μεταφράζονται σε σήματα. Υπάρχουν πολλά σήματα της εντολής kill το καθένα από τα οποία εξυπηρετεί έναν συγκεκριμένο σκοπό. Πληκτρολογώντας το συνδυασμό kill -l μπορούμε να δούμε μια λίστα από αυτά τα kill σήματα. Όλα τα σήματα ξεκινούν από «SIG» (από το SIGnal).

```
[root@ekdda ~]# kill -l
1) SIGHUP  2) SIGINT  3) SIGQUIT  4) SIGILL  5) SIGTRAP
6) SIGABRT 7) SIGBUS  8) SIGFPE  9) SIGKILL 10) SIGUSR1
11) SIGSEGV 12) SIGUSR2 13) SIGPIPE 14) SIGALRM 15) SIGTERM
16) SIGSTKFLT 17) SIGCHLD 18) SIGCONT 19) SIGSTOP 20) SIGTSTP
21) SIGTTIN 22) SIGTTOU 23) SIGURG 24) SIGXCPU 25) SIGXFSZ
26) SIGVTALRM 27) SIGPROF 28) SIGWINCH 29) SIGIO 30) SIGPWR
31) SIGSYS 34) SIGRTMIN 35) SIGRTMIN+1 36) SIGRTMIN+2 37) SIGRTMIN+3
38) SIGRTMIN+4 39) SIGRTMIN+5 40) SIGRTMIN+6 41) SIGRTMIN+7 42) SIGRTMIN+8
43) SIGRTMIN+9 44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9 56) SIGRTMAX-8 57) SIGRTMAX-7
58) SIGRTMAX-6 59) SIGRTMAX-5 60) SIGRTMAX-4 61) SIGRTMAX-3 62) SIGRTMAX-2
63) SIGRTMAX-1 64) SIGRTMAX
```

Τα σημαντικότερα σήματα, καθώς και οι κωδικοί στους οποίους αντιστοιχούν φαίνονται στον ακόλουθο Πίνακας 5- 1.

| Κωδικός Σήματος | Όνομα Σήματος  | Αποτέλεσμα                  |
|-----------------|----------------|-----------------------------|
| 0               | SIGNULL (NULL) | Κενό (Null)                 |
| 1               | SIGHUP (HUP)   | Κλείσιμο (Hangup)           |
| 2               | SIGINT (INT)   | Διακοπή από το πληκτρολόγιο |
| 3               | SIGQUIT (QUIT) | Εγκατάλειψη (Quit)          |
| 9               | SIGKILL (KILL) | Σκότωμα (Kill)              |
| 15              | SIGTERM (TERM) | Τερματισμός (Termination)   |
| 17,19,23        | SIGSTOP (STOP) | Σταμάτημα (Stop)            |

Πίνακας 5- 1: Κωδικοί και σήματα της εντολής Kill

Μια πλήρη λίστα, με επεξήγηση των σημάτων Kill μπορεί ο αναγνώστης να βρει στο (LinuxOrgKillSignals, n.d.)

### Ασκήσεις - Δραστηριότητες

Δημιουργήστε μια διεργασία σε ένα τερματικό ως χρήστες. Για παράδειγμα σε ένα terminal δημιουργήστε ένα ατέρμων ring. Στη συνέχεια ως διαχειριστές παρακολουθήστε την με την εντολή ps, την top και την htop ώστε να δείτε τη δραστηριότητα της. Στη συνέχεια βρείτε το PID της, και σκοτώστε την

## 5.4 Εργαλεία Παρακολούθησης και Διαχείρισης Συστήματος

Στην παράγραφο αυτή θα δούμε δύο πολύ σημαντικά εργαλεία παρακολούθησης του Συστήματος (System Monitoring Tools)

### 5.4.1 Cockpit

Το διαχειριστικό εργαλείο Cockpit είναι ένα πολύ καλό εργαλείο διαχείρισης του Linux Server το οποίο λειτουργεί διαδικτυακά, από κάποιον browser, και μπορεί να προσφέρει στο διαχειριστή τόσο γραφικό περιβάλλον διαχείρισης όσο και περιβάλλον τερματικού.

Για την εγκατάσταση του Cockpit χρειάζεται να εκτελέσουμε την εντολή

```
[root@ekdda ~]# dnf -y install cockpit
```

Το cockpit είναι υπηρεσία (service), άρα μπορούμε να το κάνουμε start/restart/stop/enable/disable (δείτε και παράγραφο 4.1). Προσοχή στη λεπτομέρεια, όπου για τα start/restart/stop/ η ονομασία της υπηρεσίας είναι cockpit, ενώ για τα enable/disable η ονομασία της υπηρεσίας είναι cockpit.socket

```
[root@ekdda ~]# systemctl start cockpit  
[root@ekdda ~]# systemctl enable cockpit.socket
```

Αντίστοιχα ως υπηρεσία που έχει πρόσβαση στον εξωτερικό δίκτυο και γενικότερα στις δικτυακές διεπαφές (κάρτες δικτύου), θα πρέπει να ανοίξει και στο firewall. Το cockpit είναι από τις υπηρεσίες για τις οποίες πρέπει να κάνουμε εγγραφή τόσο για το service όσο και για την πόρτα

```
[root@ekdda ~]# firewall-cmd --add-service=cockpit --permanent  
[root@ekdda ~]# firewall-cmd --add-port=9090/tcp --permanent  
[root@ekdda ~]# firewall-cmd --reload
```

Αφού ολοκληρωθούν τα παραπάνω βήματα, μπορούμε να δούμε τη λειτουργία του σε έναν browser. Φυσικά η λειτουργία μέσω browser μπορεί να γίνει από οπουδήποτε



Προσοχή στο πρωτόκολλο της διεύθυνσης URL για το Cockpit.

Είναι το ασφαλές **https**

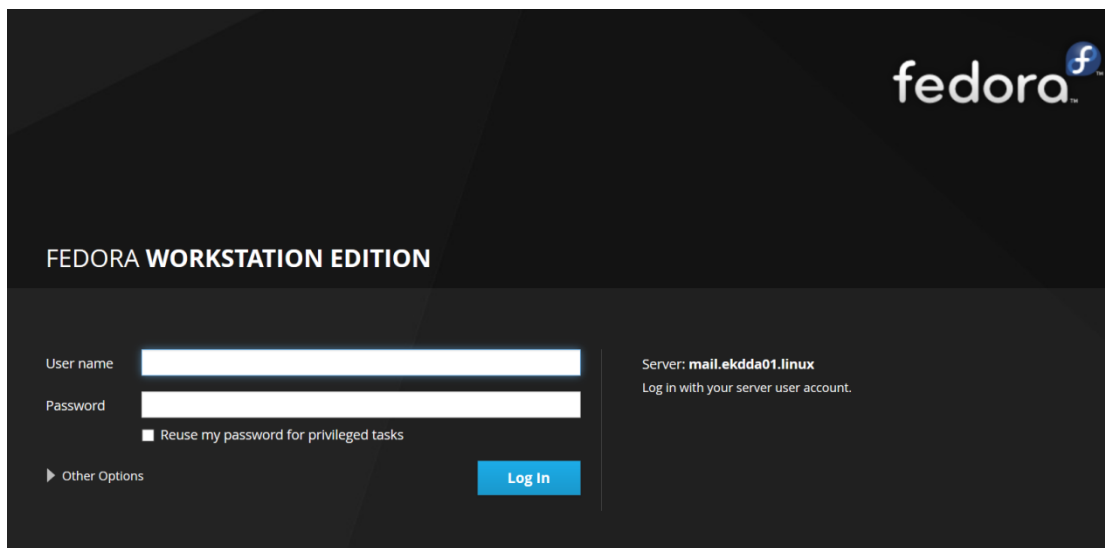
Η διεύθυνση URL στην οποία μπορούμε να δούμε το cockpit και να διαχειριστούμε τον Linux Server μας είναι :

**https://<Τοπική ή Πραγματική IP> : 9090**

Για παράδειγμα **https://192.168.1.223:9090**

Επειδή η σύνδεση είναι ασφαλής (πρωτόκολλο https), θα πρέπει να υπάρχει και κάποιο ψηφιακό πιστοποιητικό TLS το οποίο μας παρέχει ο πάροχος της διαδικτυακής μας σύνδεσης και της πραγματικής IP μας.

Όταν πλέον συνδεθούμε, βλέπουμε την παρακάτω ενδεικτική Εικόνα 5- 11 σύνδεσης.

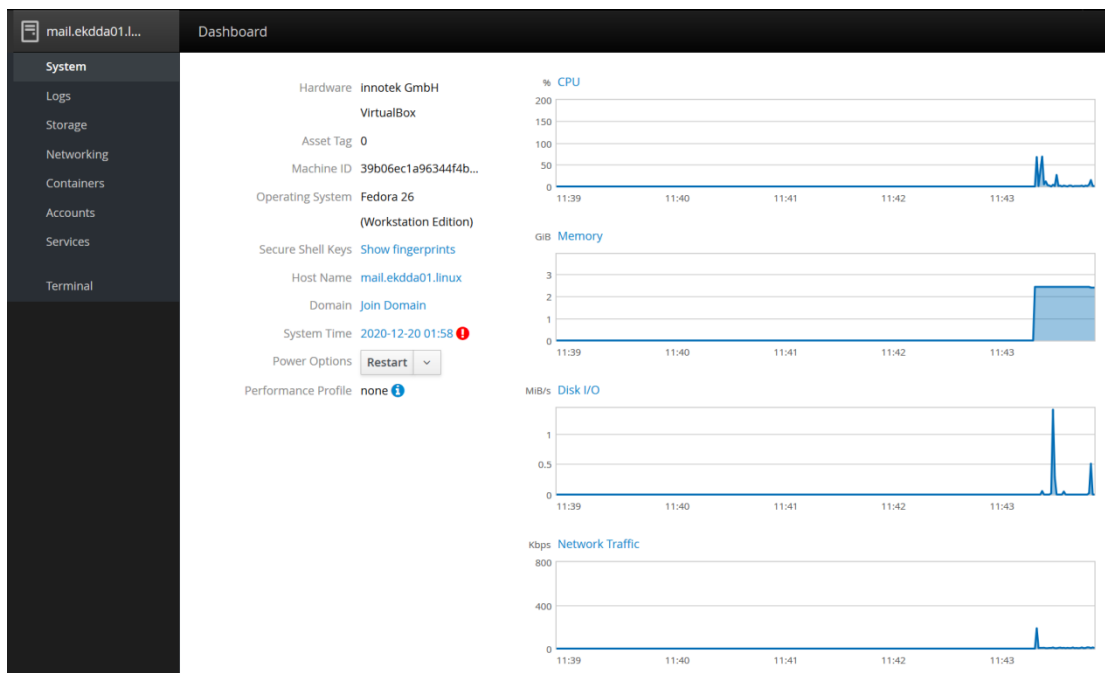


Εικόνα 5- 11: Εισαγωγική οθόνη του Cockpit

Τα στοιχεία σύνδεσης για το διαχειριστή και τους χρήστες είναι τα ίδια με αυτά που έχουν στον Linux Server.

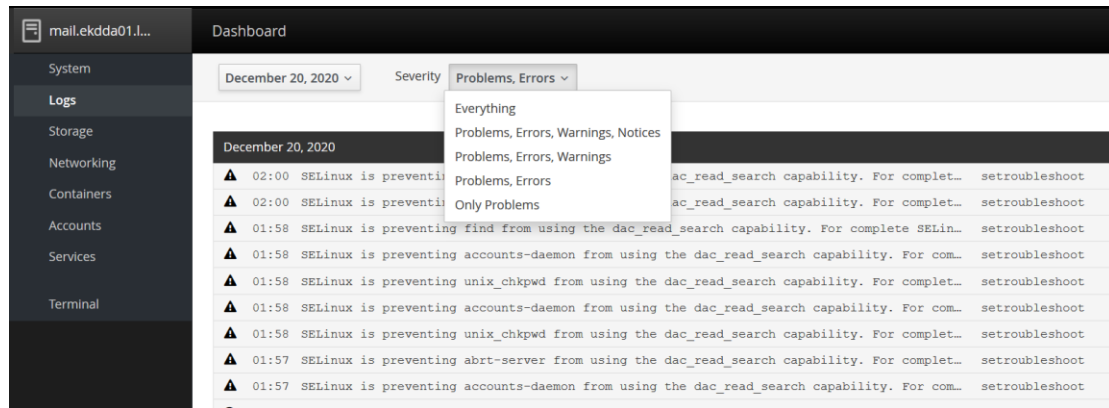
Στις επόμενες εικόνες βλέπουμε τις διάφορες οθόνες διαχείρισης του Linux Server, από το περιβάλλον του εργαλείου Cockpit.

Στην επόμενη Εικόνα 5- 12 βλέπουμε το περιβάλλον παρακολούθησης με απεικόνιση της χρήσης του επεξεργαστή, της χρήσης της μνήμης, της χρήσης των δίσκων και της κίνησης στις δικτυακές διεπαφές (κάρτες δικτύου)



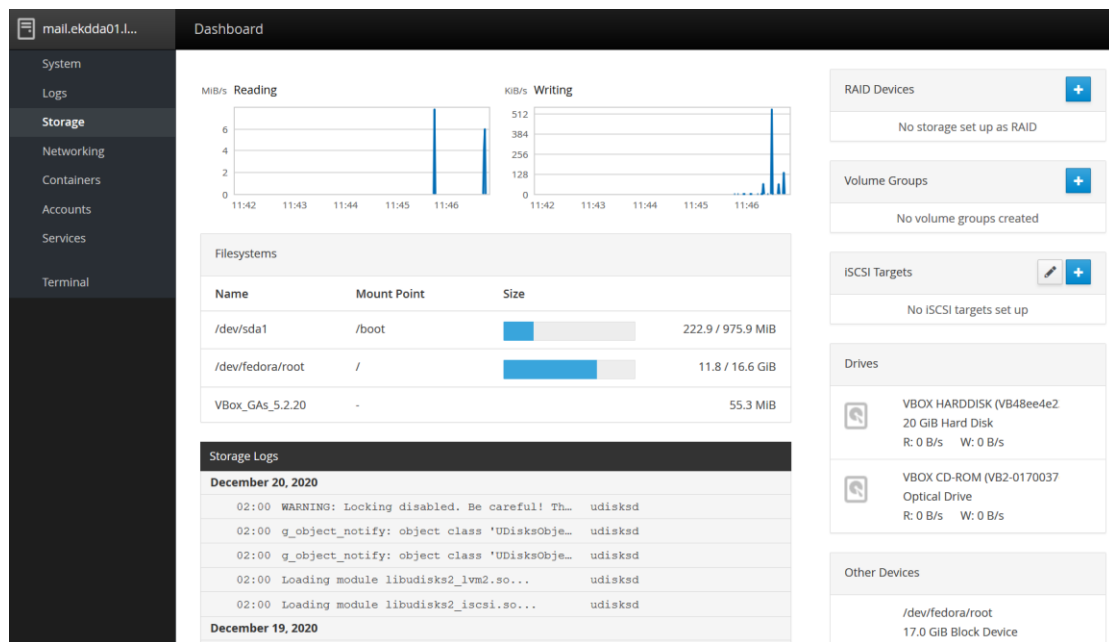
Εικόνα 5- 12: Η οθόνη system του cockpit

Στην επόμενη Εικόνα 5- 13 βλέπουμε την ενότητα με τα αρχεία καταγραφής των διαφόρων εφαρμογών και υπηρεσιών του Server



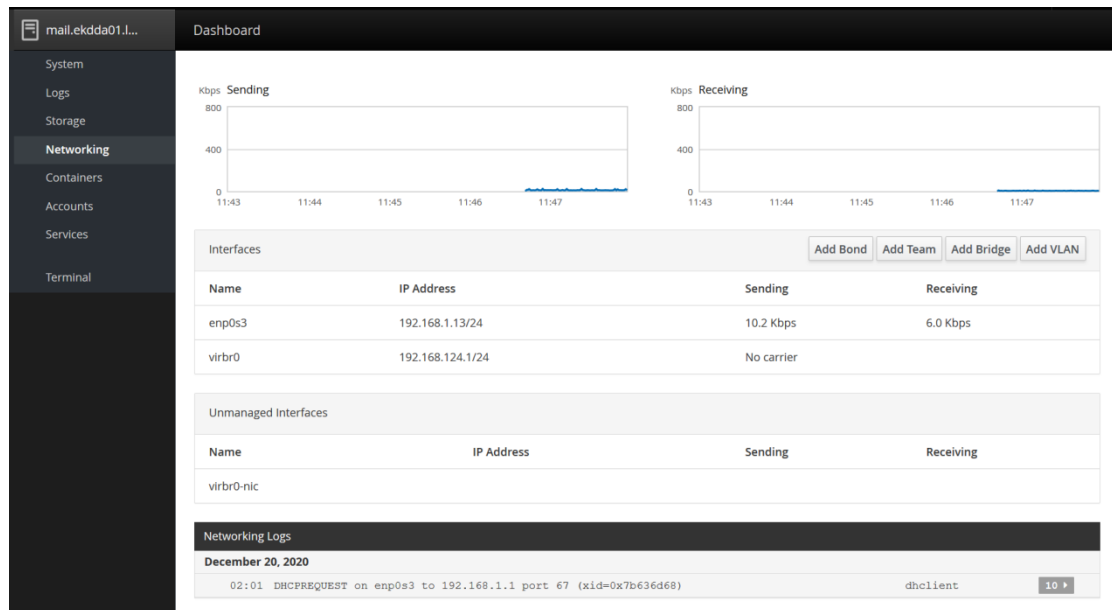
Εικόνα 5- 13: Η οθόνη logs του cockpit

Στην επόμενη Εικόνα 5- 14 βλέπουμε πληροφορίες για την αποθήκευση στα διάφορα μέσα και τις διάφορες κατατμήσεις



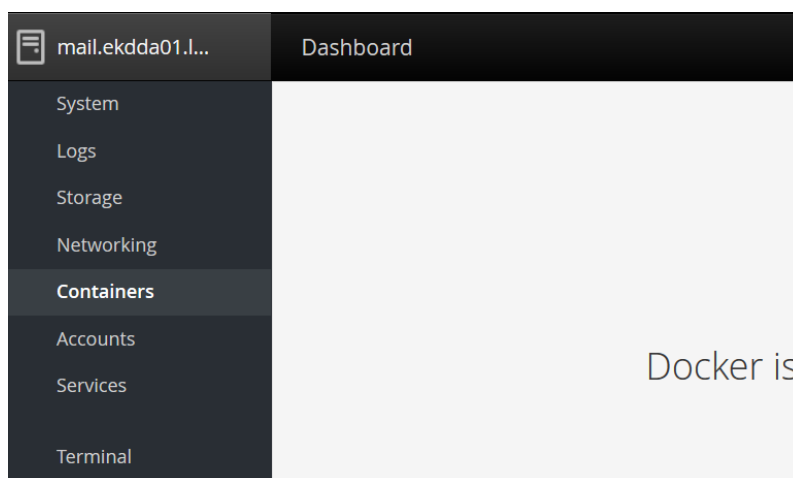
Εικόνα 5- 14: Η οθόνη storage του cockpit

Στην επόμενη Εικόνα 5- 15 βλέπουμε στοιχεία για την κατάσταση των δικτυακών διεπαφών



Εικόνα 5- 15: Η οθόνη networking του cockpit

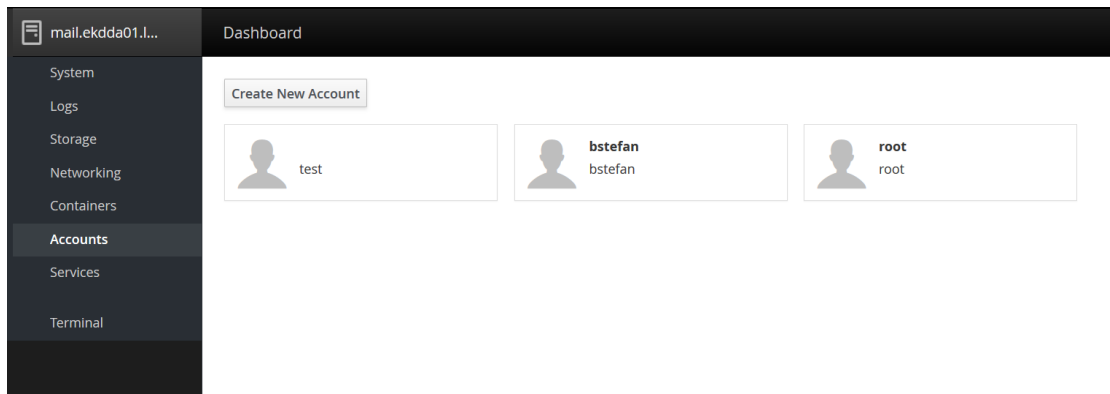
Στην επόμενη Εικόνα 5- 16 βλέπουμε την ενότητα containers. Η ενότητα αυτή παρουσιάζει στοιχεία όταν χρησιμοποιούμε το cockpit για παρακολούθηση παραπάνω από έναν Linux Server



Εικόνα 5- 16: Η οθόνη containers του cockpit

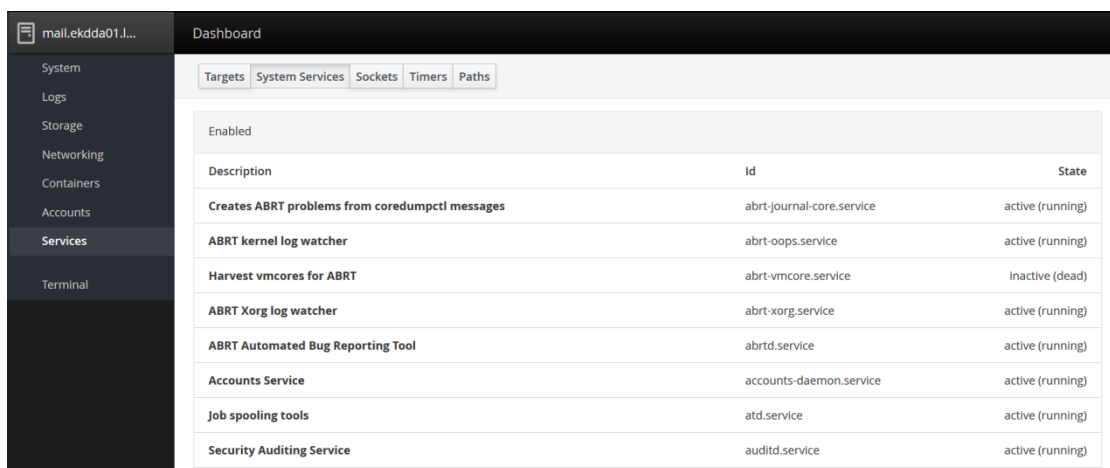
Στην επόμενη Εικόνα 5- 17 βλέπουμε την ενότητα διαχείρισης των χρηστών του Linux Server





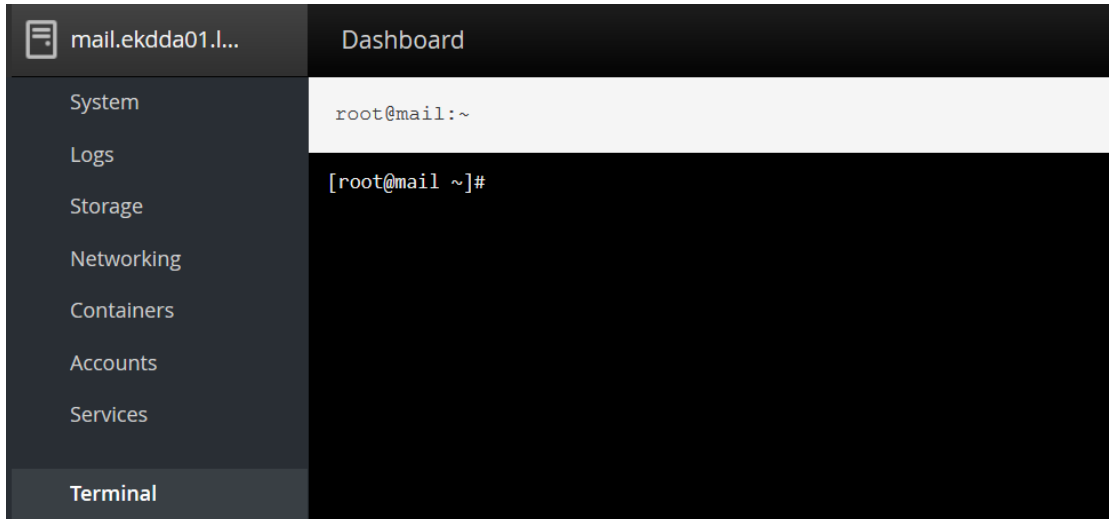
Εικόνα 5- 17: Η οθόνη accounts του cockpit

Στην επόμενη Εικόνα 5- 18 βλέπουμε την ενότητα όπου μπορούμε να παρακολουθήσουμε τις διάφορες υπηρεσίες και εφαρμογές που εκτελούνται στο σύστημά μας, καθώς και την κατάσταση στην οποία βρίσκονται (active – running κτλ.).



Εικόνα 5- 18: Η οθόνη services του cockpit

Και τέλος στην επόμενη Εικόνα 5- 19 βλέπουμε ένα καθαρό root τερματικό για τη διαχείριση του Linux Server από τη γραμμή εντολών.



Εικόνα 5- 19: Η οθόνη terminal του cockpit

### Ασκήσεις - Δραστηριότητες

Πραγματοποιήστε την εγκατάσταση της εφαρμογής σύμφωνα με τις οδηγίες και πραγματοποιήστε τις οδηγίες της παραγράφου και δείτε τα αποτελέσματα που φαίνονται στις εικόνες

#### 5.4.2 Webmin

Το διαχειριστικό εργαλείο Webmin είναι ένα ισχυρότατο εργαλείο διαχείρισης του Linux Server το οποίο λειτουργεί διαδικτυακά και μπορεί να προσφέρει στο διαχειριστή τόσο γραφικό περιβάλλον διαχείρισης όσο και περιβάλλον τερματικού.



Στο συγκεκριμένο κείμενο δε σκοπεύουμε να περιγράψουμε αναλυτικά το **webmin** ως διαχειριστικό εργαλείο. Σκοπός μας είναι να δείξουμε την εγκατάσταση και μια μικρή παρουσίαση του συγκεκριμένου συστήματος. Το webmin είναι πολύ σπουδαίο διαχειριστικό εργαλείο και θα μπορούσε να αποτελέσει από μόνο του ένα ολόκληρο σεμινάριο

Το Webmin προσφέρει πολύ όμορφο γραφικό περιβάλλον, χωρίς να χρειάζεται ο Server να έχει X Windows Server σε λειτουργία ή απλά εγκατεστημένο.

Για την εγκατάσταση του webmin χρειάζεται να κατεβάσουμε το αρχείο με την εντολή wget, επειδή μέχρι σήμερα αποτελεί για τη Fedora εφαρμογή out of the box (δεν υπάρχει στα repositories).

Η έκδοση που χρησιμοποιούμε σε αυτή τη διανομή είναι ενδεικτική. Μπορούμε να βρούμε τη νεότερη στο (Webmin\_releases, n.d.).

```
[root@ekdda ~]# wget
http://prdownloads.sourceforge.net/webadmin/webmin-1.860-
1.noarch.rpm
```

Πριν την εγκατάσταση του webim χρειάζεται να εγκατασταθούν κάποια επιπρόσθετα πακέτα perl και php.

```
[root@ekdda ~]# dnf -y install perl perl-Net-SSLeay openssl
[root@ekdda ~]# dnf -y install perl-IO-Tty perl-Encode-Detect
```

Επειδή πρόκειται για καθαρό rpm, κάνουμε εγκατάσταση του πακέτου με την εντολή

```
[root@ekdda ~]# rpm -U webmin-1.860-1.noarch.rpm
```

Ευτυχώς δεν υπάρχουν εξαρτήσεις από άλλα πακέτα, οπότε το webmin είναι έτοιμο για χρήση

Οι διαδικασίες για start/restart/stop/enable/disable δε χρειάζονται, όπως επίσης δε χρειάζεται κάποια ενέργεια στο firewall.

Αφού ολοκληρωθούν τα παραπάνω βήματα, μπορούμε να δούμε τη λειτουργία του σε έναν browser. Φυσικά η λειτουργία μέσω browser μπορεί να γίνει από οπουδήποτε.



Προσοχή στο πρωτόκολλο της διεύθυνσης URL για το Webmin. Όπως και στο Cockpit είναι το ασφαλές **https**

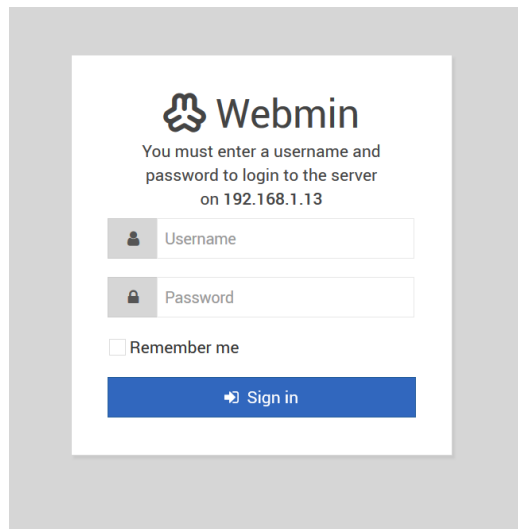
Η διεύθυνση URL στην οποία μπορούμε να δούμε το Webmin και να διαχειριστούμε τον Linux Server μας είναι :

**https://<Τοπική ή Πραγματική IP> : 10000**

**Για παράδειγμα https://192.168.1.13:10000**

Επειδή η σύνδεση είναι ασφαλής (πρωτόκολλο https), θα πρέπει να υπάρχει και κάποιο ψηφιακό πιστοποιητικό TLS το οποίο μας παρέχει ο πάροχος της διαδικτυακής μας σύνδεσης και της πραγματικής IP μας.

Όταν πλέον συνδεθούμε, βλέπουμε την παρακάτω ενδεικτική Εικόνα 5- 20 σύνδεσης.

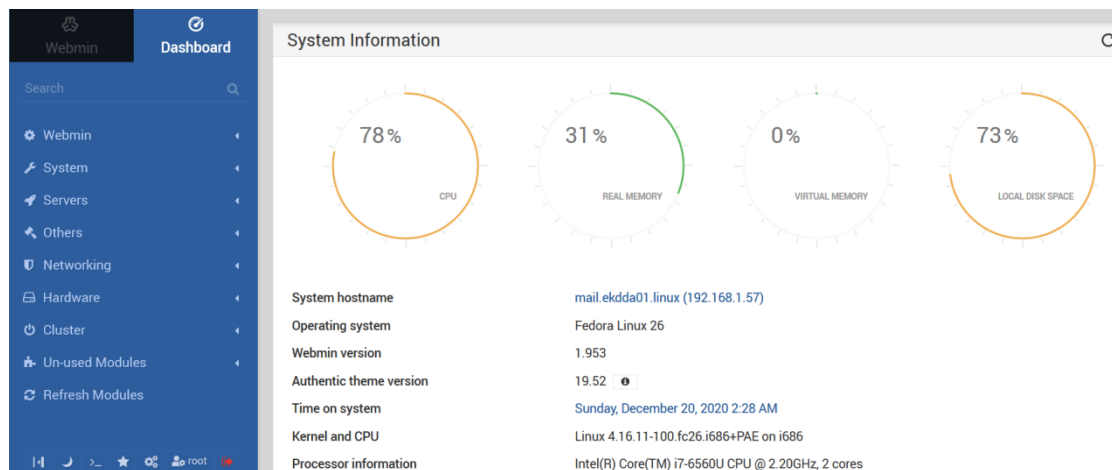


Εικόνα 5- 20: Η εισαγωγική οθόνη του webmin

Τα στοιχεία σύνδεσης για το διαχειριστή και τους χρήστες είναι τα ίδια με αυτά που έχουν στον Linux Server.

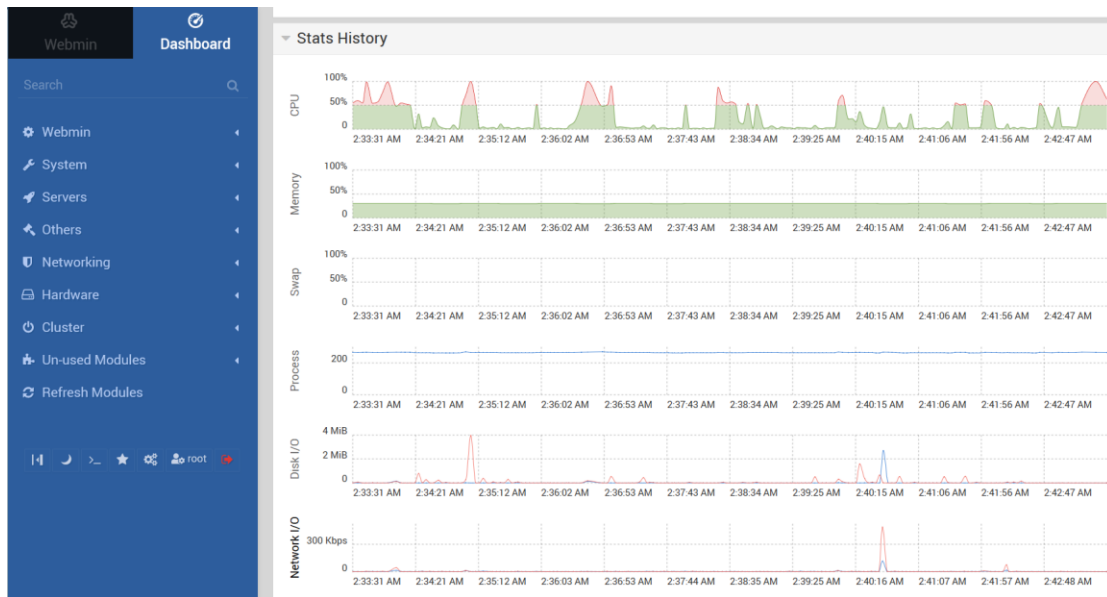
Στις επόμενες εικόνες βλέπουμε τις διάφορες οθόνες διαχείρισης του Linux Server, από το περιβάλλον του εργαλείου Webmin.

Στην επόμενη Εικόνα 5- 21 βλέπουμε το περιβάλλον παρακολούθησης με απεικόνιση της χρήσης του επεξεργαστή, της χρήσης της μνήμης, της χρήσης των δίσκων και της κίνησης στις δικτυακές διεπαφές (κάρτες δικτύου)



Εικόνα 5- 21: Το Dashboard του webmin με πληροφορίες του Server

Επίσης υπάρχει ένα πλούσιο σύνολο από γραφήματα τα οποία οπτικοποιούν την κατάσταση λειτουργίας του Linux Server μας Εικόνα 5- 21 και Εικόνα 5- 23.



Εικόνα 5- 22: Το Dashboard του webmin με πληροφορίες του Server

| Recent Logins |                       |            |
|---------------|-----------------------|------------|
| IP address    | Last active at        | State      |
| 192.168.1.7   | 12/20/2020 2:42:44 AM | This login |

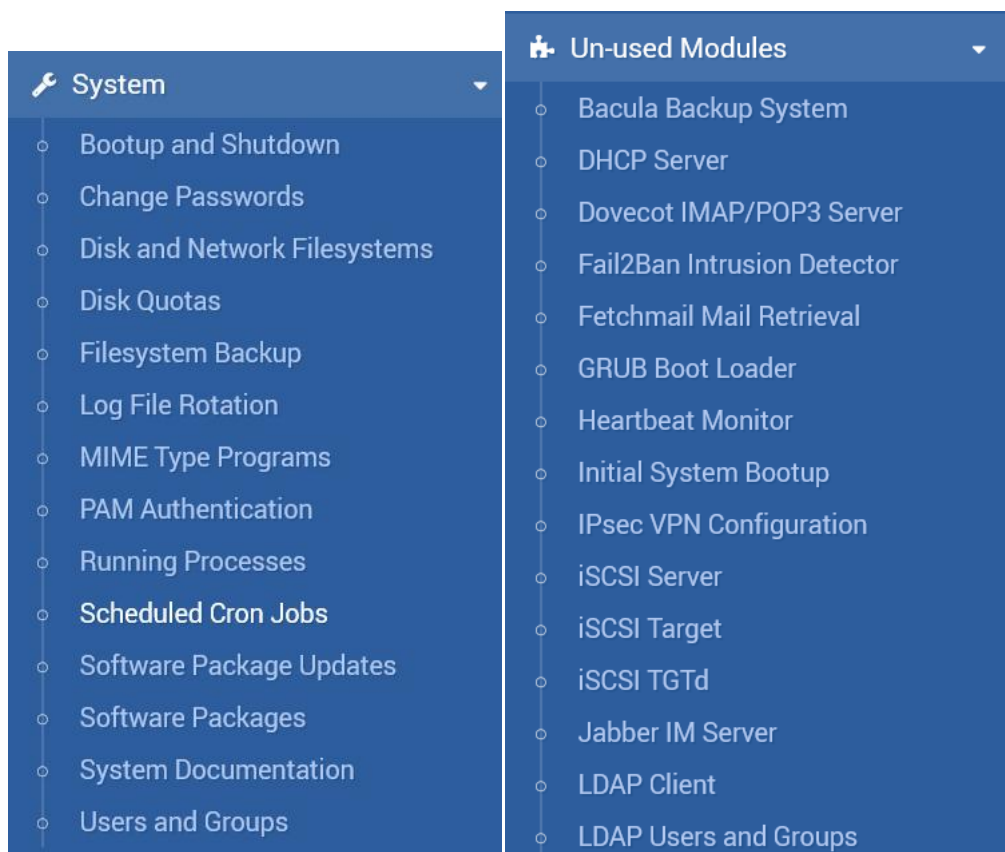
| Disk Usage |      |                  |            |  |
|------------|------|------------------|------------|--|
| Mounted As | Type | Free             | Total      | Device ID                                |
| /          | ext4 | 24% (3.93 GiB)   | 16.6 GiB   | /dev/mapper/fedora-root                  |
| /boot      | ext4 | 71% (685.77 MiB) | 975.89 MiB | UUID=d72aa77c-e233-4d45-b54a-697afb99ddf |

| Network Interfaces |          |                 |                       |                           |               |                 |        |
|--------------------|----------|-----------------|-----------------------|---------------------------|---------------|-----------------|--------|
| Name               | Type     | Interface Speed | IPv4 Address          | IPv6 Address              | Netmask       | Broadcast       | Active |
| enp0s3             | Ethernet | 1000Mb/s        | 192.168.1.13          | fe80::3eab:52eb:192c:78af | 255.255.255.0 | 192.168.1.255   | Active |
| virbr0             | Bridge   |                 | 192.168.124.1         |                           | 255.255.255.0 | 192.168.124.255 | Active |
| virbr0-nic         | Bridge   |                 | No address configured |                           | None          |                 | Down   |

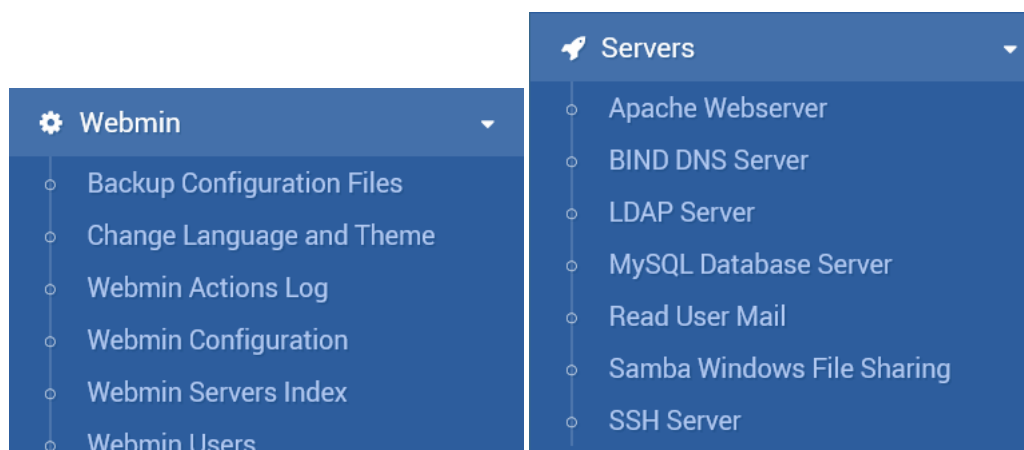
Εικόνα 5- 23: Το Dashboard του webmin με πληροφορίες του Server

Στις επόμενες εικόνες (Εικόνα 5- 24 και Εικόνα 5- 25) βλέπουμε κάποιες επιλογές από κάποια μενού



Εικόνα 5- 24: Τα μενού System και Un-used Modules του webmin

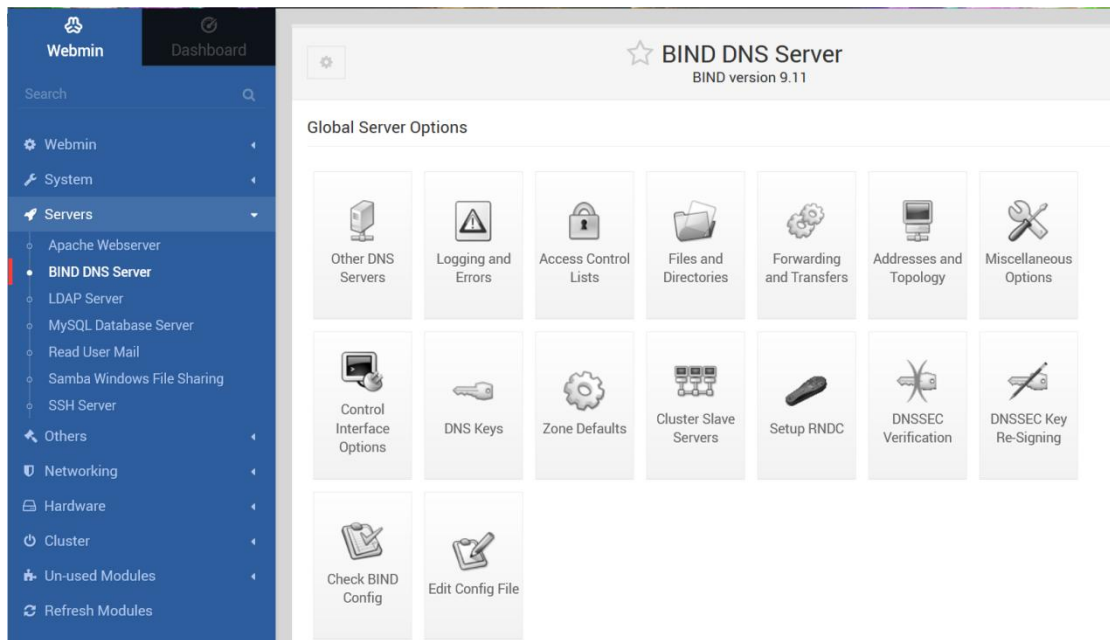
Στην Εικόνα 5- 24 βλέπουμε τους Servers που δεν είναι εγκατεστημένοι και τους οποίους μπορούμε να εγκαταστήσουμε με ένα απλό click



Εικόνα 5- 25: Τα μενού Webmin και Servers του webmin

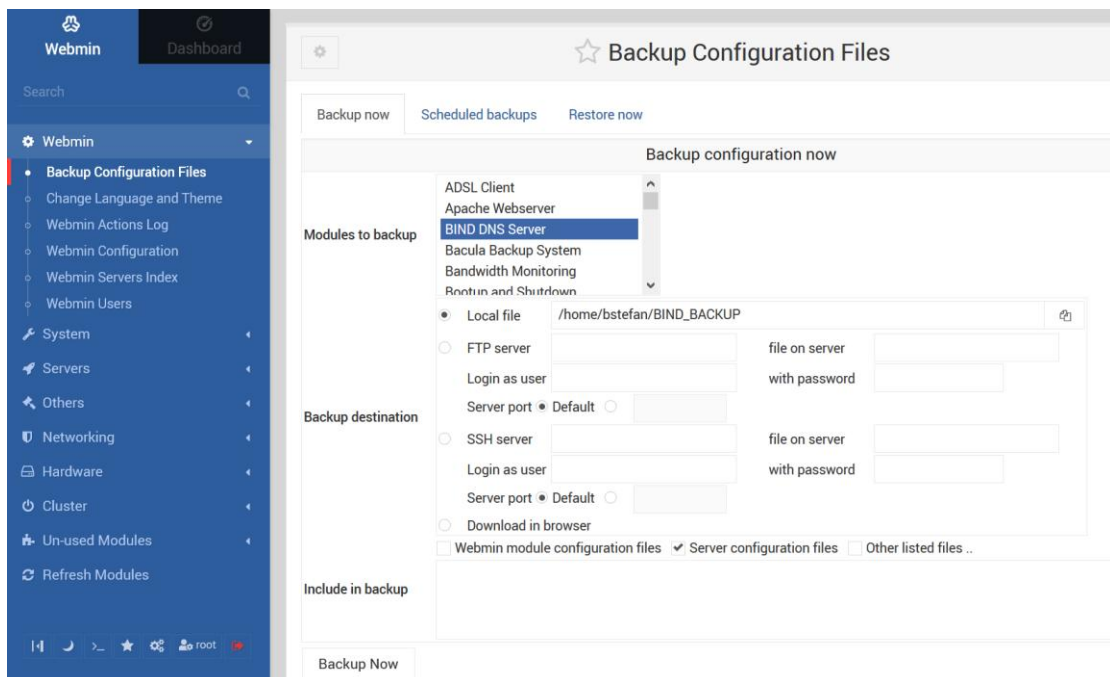
Στην Εικόνα 5- 25 βλέπουμε τους εγκατεστημένους Servers τους οποίους μπορούμε να διαχειριστούμε μέσα από ένα εύχρηστο γραφικό περιβάλλον.

Στην επόμενη Εικόνα 5- 26 βλέπουμε ενδεικτικά την οθόνη διαχείρισης ενός από του Servers (πιο συγκεκριμένα του Bind DNS Server).



Εικόνα 5- 26: Η οθόνη παραμετροποίησης του Bind DNS Server στο webmin

Στην επόμενη Εικόνα 5- 27 βλέπουμε μια από τις πολύ σημαντικές ενέργειες που πρέπει να κάνουμε συχνά ως διαχειριστές. Βλέπουμε την εύκολη και γραφική διαχείριση των αντιγράφων ασφαλείας διαφόρων εφαρμογών και υπηρεσιών του συστήματός μας.



Εικόνα 5- 27: Η οθόνη Backup των διαφόρων εφαρμογών και υπηρεσιών στο webmin

### Ασκήσεις - Δραστηριότητες

Πραγματοποιήστε την εγκατάσταση της εφαρμογής σύμφωνα με τις οδηγίες και πραγματοποιήστε τις οδηγίες της παραγράφου και δείτε τα αποτελέσματα που φαίνονται στις εικόνες.

## 6 Δικτυακές Υπηρεσίες

Στο κεφάλαιο αυτό περιγράφονται κάποιες βασικές έννοιες προκειμένου να είναι δυνατή η κατανόηση των θεμάτων των σχετικών με τη διαχείριση Linux Server. Όπως είναι προφανές δεν αποτελεί σκοπό του κεφαλαίου η πλήρης κάλυψη θεμάτων σχετικών με τα δίκτυα, όπως για παράδειγμα το πρωτόκολλο IP ή η διευθυνσιοδότηση δικτύων. Θα αναφερθούν μόνο θέματα που χρειάζονται για την κατανόηση των δικτυακών υπηρεσιών για τη διαχείριση του Linux Server.

Ως δικτυακές υπηρεσίες νοούνται αυτές οι οποίες χρησιμοποιούνται από το σύστημα για να εξυπηρετήσουν δικτυακά και διαδικτυακά αιτήματα, όπως για παράδειγμα ένα αίτημα ανάγνωσης ιστοσελίδας ή ένα αίτημα αποστολής αλληλογραφίας. Υπάρχουν διάφοροι τύποι υπηρεσιών στην από άκρο σε άκρο επικοινωνία. Οι υπηρεσίες αυτές μπορεί να είναι είτε σύγχρονες είτε ασύγχρονες (geeksforgeeks\_connection-less-service) (geeksforgeeks\_connection-oriented-service).

Μια **σύγχρονη υπηρεσία** μεταφέρει ένα συρμό (ακολουθία) από bits με σταθερή καθυστέρηση και με δεδομένο ρυθμό σφαλμάτων.

Μια **ασύγχρονη υπηρεσία** μεταφέρει συρμούς από bits ή αρχεία από bits τα οποία έχουν διαιρεθεί σε πακέτα. Η μεταφορά αυτή περιέχει τις παρακάτω λειτουργίες :

- Αναμονή (Listen)
- Σύνδεση (Connect)
- Λήψη (Receive)
- Αποστολή (Send)
- Αποσύνδεση (Disconnect)

**Η ασύγχρονη επικοινωνία** υποστηρίζει δύο κατηγορίες υπηρεσιών:

- Με σύνδεση
- Χωρίς σύνδεση

Μια υπηρεσία **με σύνδεση** (connection oriented service) εγκαθιστά πρώτα μια σύνδεση από άκρο σε άκρο, τη χρησιμοποιεί και τέλος τη λύνει (αποδεσμεύει τους συμμετέχοντες). Τα πακέτα των δεδομένων μεταφέρονται ακολουθιακά μέσα σε αγωγό, όπου ο αποστολέας εισάγει τα δεδομένα στο ένα άκρο και τα λαμβάνει ο παραλήπτης στο άλλο άκρο. Μια τέτοια υπηρεσία έχει ως πρότυπο το τηλεφωνικό δίκτυο.

Μια υπηρεσία **χωρίς σύνδεση** (connection less service) μεταφέρει τα πακέτα ανεξάρτητα το ένα από το άλλο και μάλιστα χωρίς να απαιτείται η αποκατάσταση της σύνδεσης από άκρο σε άκρο. Κάθε πακέτο φέρει τη διεύθυνση του προορισμού και η υπηρεσία αυτή μοιάζει με το κλασικό ταχυδρομείο. Είναι λοιπόν δυνατό, μηνύματα που στέλνονται σε έναν προορισμό να φθάσουν με διαφορετική σειρά, οπότε απαιτείται η αναδιάταξή τους.

Ανάλογα με την ποιότητα υπηρεσίας γίνεται η μεταφορά των δεδομένων με απουσία ή όχι λαθών, με επιβεβαίωση ή χωρίς επιβεβαίωση. Η υπηρεσία με σύνδεση θεωρείται αξιόπιστη όταν μεταφέρει ακολουθίες μηνυμάτων ή bits και μη αξιόπιστη για τη μεταφορά



ψηφιοποιημένης φωνής. Η υπηρεσία χωρίς σύνδεση μπορεί να είναι μη αξιόπιστη όταν μεταφέρει μηνύματα που δεν απαιτείται επαλήθευση και αξιόπιστη όταν απαιτείται επαλήθευση (Andrew S Tanenbaum, 1996).

## 6.1 Το Μοντέλο Αναφοράς O.S.I.

Τα πρώτα δίκτυα δε σχεδιάστηκαν με βάση κάποια πρότυπα ή γενικές αρχές, αλλά με βάση τις εκάστοτε ανάγκες και την πολιτική της κάθε εταιρείας. Με την τεχνολογική ανάπτυξη και την ευρεία χρήση των δικτύων στις ζωές μας, κρίθηκε αναγκαίο η ύπαρξη κάποιων γενικών αρχών στη σχεδίαση των δικτύων. Έτσι αναπτύχθηκαν κάποια μοντέλα αναφοράς που βασίστηκαν στα μέχρι τότε κοινά σημεία και αφορούσαν τα επίπεδα επικοινωνίας μεταξύ των διαφόρων συσκευών και λογισμικών που συμμετέχουν σε μια επικοινωνία και τον τρόπο που αυτά παραδίδουν τα δεδομένα από επίπεδο σε επίπεδο (Andrew S Tanenbaum, 1996).



Η περιγραφή των πρωτοκόλλων καθώς και τα θέματα διευθυνσιοδότησης δικτύων είναι θέματα που εξετάζονται από το πεδίο των «Υπολογιστικών Δικτύων» (Computer Networks). Σκοπός αυτής της μικρής αναφοράς είναι να θιγούν θέματα απολύτως απαραίτητα για την κατανόηση των δικτυακών υπηρεσιών σε έναν Linux Server και όχι να παρουσιαστεί μια πλήρης περιγραφή των θεμάτων αυτών.

Το 1977 ο Διεθνής Οργανισμός Τυποποίησης I.S.O. ανέπτυξε ένα Πρότυπο Διασύνδεσης Ανοικτών Συστημάτων O.S.I. (Open System Interconnection). Το μοντέλο O.S.I. αποτελεί το πλαίσιο μέσα στο οποίο κινούνται οι λεπτομερείς πλέον τυποποιήσεις, για την επίλυση των επιμέρους προβλημάτων που εμφανίζονται στις επικοινωνίες υπολογιστών. Πιο συγκεκριμένα, γίνεται ο ακριβής προσδιορισμός της αρχιτεκτονικής και του πρωτοκόλλου επικοινωνίας υπολογιστικών συστημάτων με αποτέλεσμα διαφορετικά υπολογιστικά συστήματα του εμπορίου να μπορούν να επικοινωνούν μεταξύ τους (Andrew S Tanenbaum, 1996).

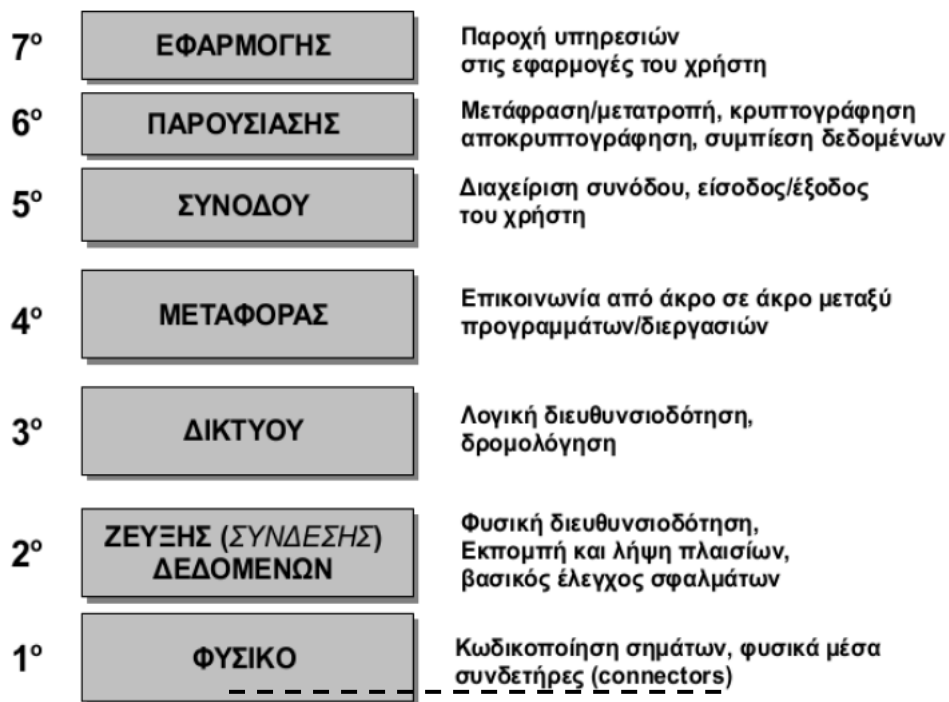
Το μοντέλο O.S.I. προσβλέπει στην ανοικτή και ελεύθερη επικοινωνία μεταξύ συστημάτων βασιζόμενο στην φιλοσοφία των επιπέδων. Όλες οι λειτουργίες που απαιτούνται για την επικοινωνία ομαδοποιούνται σε επτά επίπεδα. Οι λειτουργίες αυτές είναι ανεξάρτητες μεταξύ τους, έτσι ώστε οι αλλαγές σε ένα επίπεδο να μην έχουν επίδραση στα άλλα. Τα επίπεδα αυτά υλοποιούν τις παρακάτω διαδικασίες μιας επικοινωνίας (Wikipedia\_OSI\_model, n.d.) :

- Αναμονή για εισερχόμενες κλήσεις
- Υλοποίηση της σύνδεσης
- Λήψη εισερχόμενων μηνυμάτων
- Αποστολή ενός μηνύματος
- Λύση της σύνδεσης

Τα γνωστά πρωτόκολλα TCP/IP ήδη υπήρχαν την εποχή που είχε αρχίσει η σχεδίαση του μοντέλου O.S.I. Εκείνη την εποχή η συγκεκριμένη οικογένεια πρωτοκόλλων, η οποία βασιζόταν σε ελαφρώς διαφορετική διαστρωμάτωση επιπέδων, ήταν ήδη επί πολύ καιρό

σε ευρεία χρήση. Το TCP/IP ήταν θεμελιώδες πρωτόκολλο για το δίκτυο ARPANET και τα άλλα δίκτυα που εξελίχθηκαν στο σημερινό Διαδίκτυο. Ως αποτέλεσμα το μοντέλο O.S.I. παραμερίστηκε και σήμερα μόνο ένα υποσύνολό του χρησιμοποιείται ακόμη. Η επικρατούσα αντίληψη είναι ότι οι περισσότερες προδιαγραφές του είναι περίπλοκες και η πλήρης λειτουργικότητά του θα χρειαζόταν μεγάλο χρόνο κατασκευής, αν και συνεχίζουν να υπάρχουν σθεναροί υποστηρικτές του (Wikipedia\_OSI\_model, n.d.).

Το μοντέλο O.S.I. όπως ήδη αναφέραμε αποτελείται από επτά ιεραρχικά επίπεδα, τα οποία καλύπτουν διάφορες δικτυακές λειτουργίες, εξοπλισμό και πρωτόκολλα. Το χαμηλότερο επίπεδο βρίσκεται πλησιέστερα στο υλικό και το υψηλότερο στην εφαρμογή. Κάθε επίπεδο εκτελεί μικρό υποσύνολο λειτουργιών, που απαιτούνται για την επικοινωνία με άλλο σύστημα. Κάθε επίπεδο επικοινωνεί με τα επίπεδα, που βρίσκονται αμέσως πάνω και κάτω από αυτό και προσφέρει υπηρεσίες στο ανώτερο του επίπεδο. Στηρίζεται στο αμέσως χαμηλότερο επίπεδο, το οποίο εκτελεί τις πρωτογενείς λειτουργίες και παρέχει υπηρεσίες στο αμέσως ανώτερο. Τα επίπεδα θα πρέπει να είναι έτσι προσδιορισμένα, ώστε αλλαγές σε ένα επίπεδο να μην απαιτούν αλλαγές και σε άλλα επίπεδα. Στην Εικόνα 6- 1 φαίνονται τα επίπεδα του μοντέλου O.S.I. (Κωνσταντοπούλου Μ., 2015)



Εικόνα 6- 1: Τα επίπεδα του μοντέλου OSI

Ας δούμε μερικές γνωστές συσκευές σε ποιο επίπεδο του μοντέλου OSI λειτουργούν. Σε ένα δίκτυο υπολογιστών στο φυσικό επίπεδο (1° επίπεδο) λειτουργούν τα καλώδια, οι repeaters (επαναλήπτες) και τα hubs (συγκεντρωτές), στο επίπεδο σύνδεσης δεδομένων (2° επίπεδο) λειτουργούν οι MAC διευθύνσεις των καρτών δικτύου, τα bridges (γέφυρες) και τα switches (μεταγωγείς), στο επίπεδο δικτύου (3° επίπεδο) λειτουργούν οι routers (δρομολογητές). Στο επίπεδο εφαρμογής (7° επίπεδο) λειτουργούν οι διάφορες υπηρεσίες (DNS, FTP, κτλ.).

## 6.2 IP protocol

Πρωτόκολλο (protocol) είναι ένα σύνολο κανόνων που ελέγχουν την αλληλεπίδραση μεταξύ των διαφόρων συσκευών σε ένα δίκτυο ή το διαδίκτυο. Ειδικότερα ως πρωτόκολλο επικοινωνίας ορίζεται ένα σύνολο συμβάσεων - κανόνων συμφωνημένων και από τα δύο επικοινωνούντα μέρη, που εξυπηρετούν την μεταξύ τους ανταλλαγή πληροφοριών. Με άλλα λόγια το πρωτόκολλο επικοινωνίας, είναι μια δέσμη κανόνων στους οποίους στηρίζεται η επικοινωνία των συσκευών (υπολογιστών, ενεργών και παθητικών στοιχείων) σε ένα δίκτυο. Οι κανόνες αυτοί καθορίζουν τη μορφή, το χρόνο και τη σειρά μετάδοσης των πληροφοριών στο δίκτυο. Εκτελούν επίσης, έλεγχο και διόρθωση σφαλμάτων στη διάρκεια μετάδοσης των πληροφοριών. (WikiIPProtocol, n.d.).



Το πρωτόκολλο IP συχνά αναφέρεται ως **TCP/IP** λόγω της απόλυτης συνεργασίας των δύο πρωτοκόλλων και της απόλυτης κυριαρχίας τους στο χώρο των υπολογιστικών δικτύων. Στο TCP/IP τα επίπεδα του πρωτοκόλλου OSI διαφοροποιούνται ελαφρά (γίνονται 4 - συμπύσσονται).

Στο διαδίκτυο και στα δίκτυα τηλεπικοινωνιών γενικότερα γίνεται εκτεταμένη χρήση πρωτοκόλλων. Διαφορετικά πρωτόκολλα χρησιμοποιούνται για να εκτελέσουν διαφορετικές ενέργειες απαραίτητες στην ανταλλαγή πληροφοριών. Πολλά από αυτά τα πρωτοκόλλα είναι ιδιαίτερα απλά ενώ άλλα είναι ιδιαίτερα πολύπλοκα. Η κατανόηση των δικτύων επικοινωνιών είναι ουσιαστικά ισοδύναμη με την κατανόηση αυτών των πρωτοκόλλων.

Είναι φανερό, ότι η σχεδίαση των πρωτοκόλλων επικοινωνίας αποτελεί ένα τμήμα της σχεδίασης ενός δικτύου. Ουσιαστικά μιλάμε για μια συνολικότερη σχεδίαση που αναφέρεται ως μοντέλο.



**Μοντέλο** ονομάζεται μια προδιαγραφή που έχει οριστεί από κάποιο οργανισμό προτυποποίησης ως οδηγός αναφοράς για τη σχεδίαση δικτύων.

Για την υλοποίηση της επικοινωνίας στα δίκτυα υπολογιστών χρησιμοποιούνται διάφορες τεχνικές μεταγωγής και πολύπλεξης, η λειτουργία των οποίων ξεφεύγει από τους σκοπούς του συγκεκριμένου κειμένου. (WikiIPProtocol, n.d.) (Andrew S Tanenbaum, 1996).

Το πρωτόκολλο IP (πολλές φορές αναφέρεται και πρωτόκολλο δικτύου ή και πρωτόκολλο TCP/IP) είναι ένα σύνολο κανόνων που επιτρέπουν στα υπολογιστικά συστήματα να συνδέονται στο δίκτυο και να «μιλούν» (επικοινωνούν) μεταξύ τους και να ανταλλάσσουν δεδομένα και πληροφορίες.

Κάθε υπολογιστής που βρίσκεται στο διαδίκτυο πρέπει να έχει μια μοναδική διεύθυνση, η οποία ονομάζεται διεύθυνση IP. Το μήκος μια διεύθυνσης IP είναι 32 bits στο πρότυπο IPv4. Μια τυπική IP διεύθυνση είναι της μορφής: 195.251.209.44 και λέμε ότι είναι σε δεκαδική μορφή με τελείες (dotted decimal format). Οι υπολογιστές όμως επικοινωνούν μεταξύ τους με δυαδική μορφή. Οπότε όταν πρέπει να επικοινωνήσουν, η διεύθυνση IP μεταφράζεται

στη δυαδική μορφή της. Η προηγούμενη διεύθυνση (195.251.209.44) σε δυαδική μορφή θα είναι:

**11000011. 11111011. 11010001.00101100**

Οι τέσσερις δυαδικοί αριθμοί ονομάζονται οκτάδες (octets) διότι αποτελούνται από οκτώ δυαδικά ψηφία. Αν λάβουμε υπόψη και τις 4 θέσεις έχουμε 32 bits και για αυτό λέμε ότι το μήκος μιας διεύθυνσης IP είναι 32 bit.

Επειδή σε κάθε θέση μπορούμε να έχουμε είτε 1 είτε 0, οι δυνατοί συνδυασμοί για κάθε οκτάδα είναι  $2^8=256$ . Άρα η κάθε οκτάδα μπορεί να έχει τιμή από 0 ως 255. Για 4 οκτάδες έχουμε  $2^{32}$  δυνατές τιμές δηλαδή περίπου 4,3 δισεκατομμύρια διευθύνσεις.

Η διεύθυνση 0.0.0.0 εκπροσωπεί τη διεύθυνση όλων των δικτυακών interface που περιέχει ένα υπολογιστικό σύστημα (όλες οι κάρτες δικτύου που υπάρχουν σε έναν Linux Server). Αλλιώς ονομάζεται η «διεύθυνση γι' αυτό το δίκτυο» (this network address). Μια διεύθυνση της μορφής x.y.z.255 χρησιμοποιείται συνήθως για broadcasts. Broadcast είναι η διεύθυνση στην οποία γίνεται αποστολή ενός μηνύματος όταν δεν ξέρει ο πομπός τον παραλήπτη ή όταν θέλει να πάρουν όλα τα μέρη του δικτύου ένα μήνυμα.

Διευθύνσεις της μορφής 192.168.x.y και 10.x.y.z χρησιμοποιούνται για τοπικά δίκτυα (στα όρια μιας επιχείρησης, ενός οργανισμού η ενός σπιτιού) και δεν δρομολογούνται δια μέσου του διαδικτύου.

Οι οκτάδες χρησιμοποιούνται για να δημιουργήσουν Τάξεις (Classes) για τις IP διευθύνσεις που μπορούν να δοθούν σε έναν οργανισμό (ή εργαστήριο ή υπολογιστική νησίδα) ανάλογα με το μέγεθος και τις ανάγκες της (CISCO\_IP\_Networking).

Οι οκτάδες χωρίζονται σε δύο τμήματα: net και host. Το τμήμα net περιλαμβάνει πάντα την πρώτη οκτάδα. Χρησιμοποιείται για την ταυτότητα του δικτύου στο οποίο ανήκει ένας υπολογιστής. Το Host δείχνει την ταυτότητα του ίδιου του υπολογιστή στο δίκτυο. Το τμήμα Host περιλαμβάνει πάντα την τελευταία οκτάδα.

Υπάρχουν 5 τάξεις IP και κάποιες επιπλέον ειδικές διευθύνσεις (CISCO\_IP\_Networking) (CONTA\_Lab, n.d.).

- **Προεπιλεγμένο δίκτυο (default network):** Η διεύθυνση του είναι 0.0.0.0 και χρησιμοποιείται για το προεπιλεγμένο δίκτυο ή όπως είπαμε «διεύθυνση γι' αυτό το δίκτυο» (default network or this network address).
- **Τάξη A (Class A):** Χρησιμοποιείται από μεγάλα δίκτυα. Στην τάξη αυτή ανήκουν οι διευθύνσεις με την πρώτη οκτάδα από 1-126. Οι υπόλοιπες οκτάδες χρησιμοποιούνται για την ταυτότητα των επιμέρους hosts. Σε αυτή την τάξη ανήκουν παραπάνω από τις μισές διαθέσιμες διευθύνσεις IP. Στα δίκτυα αυτά το πρώτο δυαδικό ψηφίο της πρώτης οκτάδας της IP διεύθυνσης είναι πάντα 0.
- **Διεύθυνση Βρόχου (Loopback):** Αντιστοιχεί στην IP διεύθυνση 127.0.0.0 και χρησιμοποιείται από το σύστημά μας για να στέλνει μηνύματα πίσω στον εαυτό του (ανατροφοδότηση). Γενικότερα χρησιμοποιείται για τον έλεγχο προβλημάτων και έλεγχο του δικτύου. Μπορεί να χρησιμοποιηθεί και μόνο με το λεκτικό loopback

χωρίς τη χρήση του αριθμητικού. Επίσης μπορεί να χρησιμοποιηθεί όταν δεν έχει ακόμα αποδοθεί πραγματική ή τοπική IP στο σύστημά μας.

- **Τάξη B (Class B):** Χρησιμοποιείται από μεσαίου μέγεθος δίκτυα. Η πρώτη οκτάδα στις διευθύνσεις αυτής της τάξης είναι από 128-191. Επίσης στις διευθύνσεις αυτές η δεύτερη οκτάδα περιλαμβάνεται στη ταυτότητα του δικτύου (net identifier). Οι άλλες δύο οκτάδες χρησιμοποιούνται για την ταυτότητα του κάθε host. Σε αυτή την ομάδα ανήκουν περίπου το ένα τέταρτο των διαθέσιμων διευθύνσεων. Το πρώτο bit της πρώτης οκτάδας είναι 1 και το δεύτερο 0.
- **Τάξη C (Class C):** Οι διευθύνσεις αυτές χρησιμοποιούνται από μικρές και μεσαίες επιχειρήσεις με την πρώτη οκτάδα να είναι από 192-223. Εδώ στο Net identifier περιλαμβάνεται η δεύτερη και η τρίτη οκτάδα, ενώ η τελευταία οκτάδα χρησιμοποιείται για την ταυτότητα του host.
- **Τάξη D (Class D):** Χρησιμοποιείται για multicasting. Τα τρία πρώτα bits είναι 1 και το τέταρτο 0 ενώ τα επόμενα 28 bits χρησιμοποιούνται για να δείξουν την ομάδα των υπολογιστών για τους οποίους προορίζεται το multicast.
- **Τάξη E (Class E):** Η τάξη αυτή χρησιμοποιείται για πειραματικούς σκοπούς και είναι ίδια με την προηγούμενη.
- **Διεύθυνση Εκπομπής (Broadcast):** Τα μηνύματα που προορίζονται για όλους τους υπολογιστές που είναι συνδεδεμένοι στο διαδίκτυο στέλνονται ως broadcasts και χρησιμοποιούν πάντα την διεύθυνση IP 255.255.255.255.

Το πρωτόκολλο IP είναι μη προσανατολισμένο προς σύνδεση. Κάθε πακέτο του αποτελεί μια διαφορετική οντότητα.

Το IP δεν ενδιαφέρεται για το αν ένα πακέτο θα φτάσει στον προορισμό του ή αν τα πακέτα θα φτάσουν στη σωστή σειρά ή αν έχουν χαθεί στην πορεία. Οι έλεγχοι αυτοί γίνονται από πρωτόκολλα υψηλότερου επιπέδου, όπως το TCP.

Το πρωτόκολλο IP είναι ένα πρωτόκολλο βέλτιστης προσπάθειας (best effort protocol). Η έκδοση που χρησιμοποιείται τώρα είναι το IPv4 (τέσσερα νούμερα χωρισμένα με τελείες) με περιορισμό διαφορετικών διευθύνσεων που μπορούν να μοιραστούν. Υπάρχει και το πρωτόκολλο IPv6 το οποίο μπορεί να υποστηρίξει πολύ περισσότερες διαφορετικές διευθύνσεις αλλά η αποκλειστική χρήση του έχει μεταφερθεί για το μέλλον. (CONTA\_Lab, n.d.)

### 6.3 Διευθυνσιοδότηση τοπικών δικτύων

Το πρόβλημα που υπάρχει με το διαμοιρασμό των διευθύνσεων IP είναι ότι ένας μεγάλος οργανισμός που έχει για παράδειγμα ένα δίκτυο Class B μπορεί να έχει μέχρι 64.000 hosts οι οποίοι θα είναι πρακτικά αδύνατο να ελεγχθούν από τον διαχειριστή δικτύου. Επίσης, μπορεί να έχει πολλά τοπικά δίκτυα συνδεδεμένα μεταξύ τους με αποτέλεσμα να απαιτούνται παρά πολλές διαφορετικές διευθύνσεις IP.

Για την επίλυση του παραπάνω προβλήματος δημιουργήθηκε η διευθυνσιοδότηση με υποδίκτυα (subnet addressing).

Η ιδέα για τη διευθυνσιοδότηση με υποδίκτυα (subnet addressing ή subnetting) προϋποθέτει την προσθήκη ενός επιπέδου μεταξύ net id και host id το οποίο και

ονομάζεται subnet. Το υποδίκτυο αγνοεί το δίκτυο που βρίσκεται εκτός του οργανισμού. Δηλαδή ένας εξωτερικός host βλέπει μόνο την αρχική δομή της διεύθυνσης με δύο επίπεδα, ενώ στο εσωτερικό ο διαχειριστής του δικτύου μπορεί να επιλέγει οποιονδήποτε συνδυασμό μήκους για τα πεδία subnet id και host id.

Για να γίνει κατανοητή η διευθυνσιοδότηση με υποδίκτυα, ας σκεφτούμε για παράδειγμα έναν οργανισμό ο οποίος έχει για παράδειγμα μια Class C IP διεύθυνση αλλά βρίσκεται σε παραπάνω από ένα φυσικά δίκτυα. Μόνο οι τοπικές πύλες (local gateways) γνωρίζουν ότι υπάρχουν πολλαπλά φυσικά δίκτυα και τον τρόπο με τον οποίο πρέπει να δρομολογηθεί η κυκλοφορία των διαφόρων πακέτων σε αυτά. Οι πύλες αυτές σε άλλα συστήματα δρομολογούν την κίνηση σαν να υπάρχει μόνο ένα φυσικό δίκτυο.

Ένα παράδειγμα διευθυνσιοδότησης αυτού του τύπου φαίνεται παρακάτω.

Ένα υποδίκτυο είναι το

Διεύθυνση δικτύου 192.168.1.0

IP δρομολογητή-gateway 192.168.1.1

IPs σταθμών εργασίας 192.168.1.2 μέχρι 192.168.1.254 (μέχρι 253 σταθμούς μπορεί να περιέχει το 1<sup>ο</sup> τοπικό μας δίκτυο)

Broadcast 192.168.1.255

Μάσκα 255.255.255.0

Ένα δεύτερο υποδίκτυο είναι το

Διεύθυνση δικτύου 192.168.2.0

IP δρομολογητή-gateway 192.168.2.1

IPs σταθμών εργασίας 192.168.2.2 μέχρι 192.168.2.254 (μέχρι 253 σταθμούς μπορεί να περιέχει το 2<sup>ο</sup> τοπικό μας δίκτυο)

Broadcast 192.168.2.255

Μάσκα 255.255.255.0

Μια πληροφορία μπορεί να μεταφερθεί από το ένα υποδίκτυο στο δεύτερο μόνο μεταξύ των δύο δρομολογητών που βρίσκονται στις πύλες (gateways) και μόνο όταν η μία έχει οδηγίες δρομολόγησης προς την άλλη (όπως ονομάζεται στη γλώσσα των πρωτοκόλλων δρομολόγησης, η μια gateway πρέπει να «διαφημίζεται» στην άλλη)

Όλα τα προηγούμενα μπορούν να αναφέρονται για το μεν 1<sup>ο</sup> υποδίκτυο ως 192.168.1.0/24, ενώ για το 2<sup>ο</sup> υποδίκτυο ως 192.168.2.0/24 και πολλές φορές στα αρχεία παραμετροποίησης (configuration files) του Linux Server κάνουμε με αυτόν τον τρόπο αναφορά στο υποδίκτυο που θέλουμε να έχει πρόσβαση σε κάποια υπηρεσία. Για παράδειγμα έτσι κάνουμε και στο αρχείο παραμετροποίησης του Squid (παράγραφος 8.1)

### Ασκήσεις - Δραστηριότητες

- Συζητήστε μεταξύ σας τη διευθυνσιοδότηση του τοπικού δικτύου του εργαστηρίου σας στη συνέχεια συζητήστε διάφορα θέματα σχετικά με τη διευθυνσιοδότηση και προβλήματα που προκύπτουν από την περιορισμένο αριθμό διευθύνσεων του IPv4
- Συζητήστε για τα στοιχεία ενός υποδικτύου (gateway, broadcast, network address, hosts IPs) της μορφής 192.168.1.0/25

### 6.4 Δικτυακές υπηρεσίες - sockets - ports

Για τις δικτυακές υπηρεσίες το Linux (όπως και όλα τα λειτουργικά συστήματα) χρησιμοποιούν τις εικονικές «πόρτες» (ή «θύρες» - τελικά σημεία επικοινωνίας - communication endpoints) στις οποίες μπορεί να «ακούει» μια διεργασία (ένας δαίμονας) που θα εξυπηρετήσει ένα αίτημα. Το αίτημα αυτό μπορεί να είναι του τοπικού χρήστη, ενός απομακρυσμένου χρήστη, ενός άλλου υπολογιστή, μιας τοπικής υπηρεσίας ή και μιας άλλης απομακρυσμένης υπηρεσίας. Η δημιουργία των λογικών αυτών «θυρών» έγινε για να μπορούν τα δικτυακά πρωτόκολλα (TCP/UDP) να αντιστοιχίζουν υπηρεσίες σε αιτήματα.



**Socket** από την πλευρά των δικτυακών υπηρεσιών είναι μια σύνδεση μεταξύ μιας υπηρεσίας του Linux Server μας και ενός απομακρυσμένου συστήματος το οποίο κάνει ένα αίτημα προς το σύστημά μας

Μερικές γνωστές τέτοιες πόρτες φαίνονται στον ακόλουθο Πίνακα (Common\_Ports\_MIT, n.d.)

| Πόρτα | Όνομα       | Περιγραφή   |
|-------|-------------|---|
| 21    | ftp         | File Transfer Protocol (FTP) port. μερικές φορές χρησιμοποιείται από το File Service Protocol (FSP) |
| 22    | ssh         | Η υπηρεσία Secure Shell (SSH)   |
| 23    | telnet      | Η υπηρεσία Telnet   |
| 25    | smtp        | Simple Mail Transfer Protocol (SMTP)  |
| 53    | domain      | domain name services (such as BIND)   |
| 80    | http        | Οι υπηρεσίες HyperText Transfer Protocol (HTTP) for World Wide Web (WWW)                            |
| 88    | kerberos    | Kerberos network authentication system  |
| 109   | pop2        | Post Office Protocol version 2  |
| 110   | pop3        | Post Office Protocol version 3  |
| 115   | sftp        | Secure File Transfer Protocol (SFTP) services   |
| 119   | nntp        | Network News Transfer Protocol (NNTP) for the USENET discussion system                              |
| 123   | ntp         | Network Time Protocol (NTP)   |
| 137   | netbios-ns  | NETBIOS Name Service. Χρησιμοποιείται στο Red Hat Enterprise Linux από τον Samba                    |
| 138   | netbios-dgm | NETBIOS Datagram Service. Χρησιμοποιείται στο Red Hat Enterprise Linux από τον Samba                |
| 139   | netbios-ssn | NETBIOS Session Service. Χρησιμοποιείται στο Red Hat Enterprise Linux από τον Samba                 |
| 143   | imap        | Internet Message Access Protocol (IMAP)   |
| 161   | snmp        | Simple Network Management Protocol (SNMP)   |

|            |                |   |
|------------|----------------|---|
| <b>389</b> | ldap           | Lightweight Directory Access Protocol (LDAP)                            |
| <b>546</b> | dhcpcv6-Client | Dynamic Host Configuration Protocol (DHCP) version 6 Client             |
| <b>547</b> | dhcpcv6-Server | Η υπηρεσία Dynamic Host Configuration Protocol (DHCP) έκδοση 6          |
| <b>565</b> | Whoami         | whoami user ID listing  |
| <b>636</b> | ldaps          | Lightweight Directory Access Protocol over Secure Sockets Layer (LDAPS) |
| <b>765</b> | webster        | Network Dictionary  |
| <b>767</b> | phonebook      | Network Phonebook   |
| <b>992</b> | telnetS        | Telnet over Secure Sockets Layer (TelnetS)                              |
| <b>993</b> | imaps          | Internet Message Access Protocol over Secure Sockets Layer (IMAPS)      |
| <b>994</b> | ircs           | Internet Relay Chat over Secure Sockets Layer (IRCS)                    |
| <b>995</b> | pop3s          | Post Office Protocol version 3 over Secure Sockets Layer (POP3S)        |

Πίνακας 6- 1: Γνωστές Πόρτες Λειτουργικού Συστήματος (Common Ports)

Στο (Common\_Ports\_MIT, n.d.) μπορεί κανείς να βρει μια πλήρη λίστα με τις γνωστές πόρτες των λειτουργικών συστημάτων

Η αντιστοίχιση κάθε πόρτας σε υπηρεσίες δικτύου βρίσκεται στο αρχείο /etc/services. Εάν δημιουργήσουμε κάποια στιγμή έναν δικό μας εξυπηρετητή (Server), τότε πρέπει να προσέξουμε να αντιστοιχίσουμε μια πόρτα στον Linux Server η οποία δε χρησιμοποιείται από άλλη υπηρεσία.

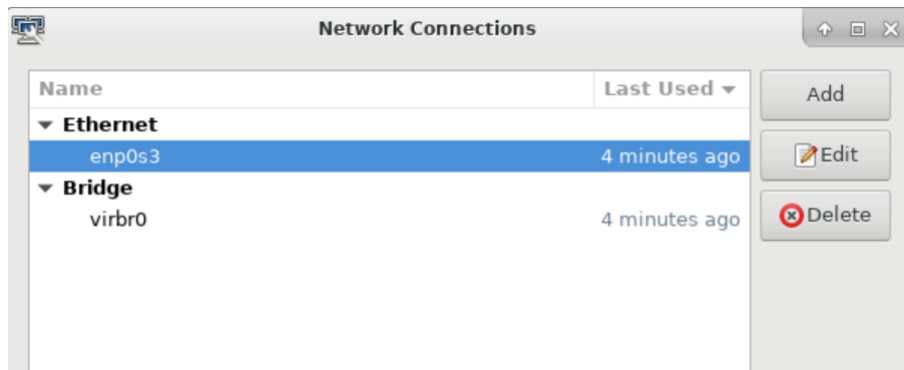
## 6.5 Βασικοί Εξυπηρετητές δικτυακής δραστηριότητας

Οι βασικοί εξυπηρετητές δικτυακής δραστηριότητας είναι αυτοί που εξυπηρετούν αιτήματα ενεργών στοιχείων του δικτύου. Τέτοιοι είναι ο εξυπηρετητής ιστοσελίδων (web Server), ο εξυπηρετητής ηλεκτρονικού ταχυδρομείου (email Server), ο Εξυπηρετητής Ονοματοδοσίας (domain name Server ή system), ο Διακομιστής Μεσολάβησης (Proxy Server), ο Εξυπηρετητής που αποδίδει δυναμικά IPs στους hosts (Dhcp Server), ο PXE-Server, ο Ftp Server κ.α.

## 6.6 Γραφικό εργαλείο διαχείρισης δικτυακών στοιχείων

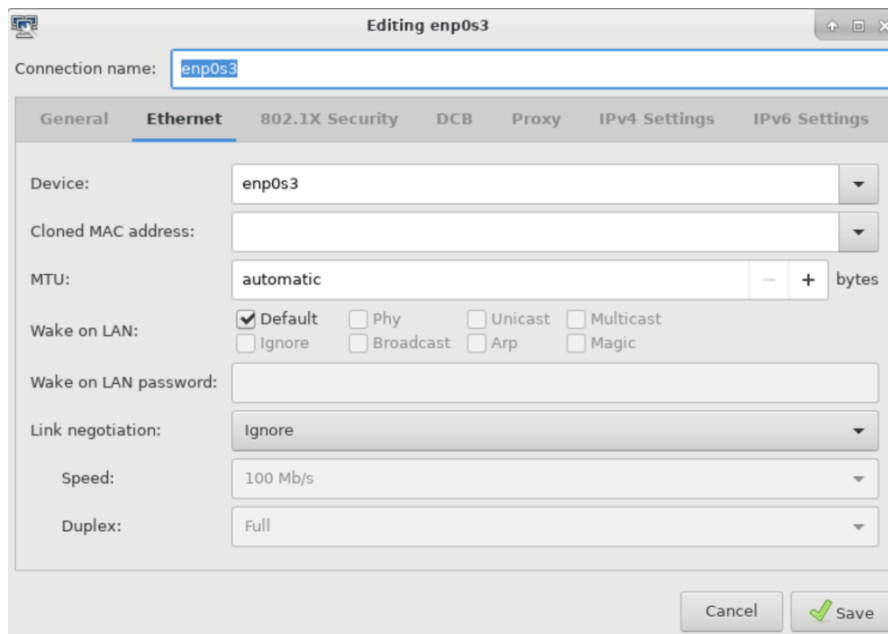
Ένα γραφικό εργαλείο για τη διαχείριση των δικτυακών στοιχείων του Linux Server το οποίο περιλαμβάνει το λειτουργικό σύστημα είναι αυτό που φαίνεται στην Εικόνα 6- 2:



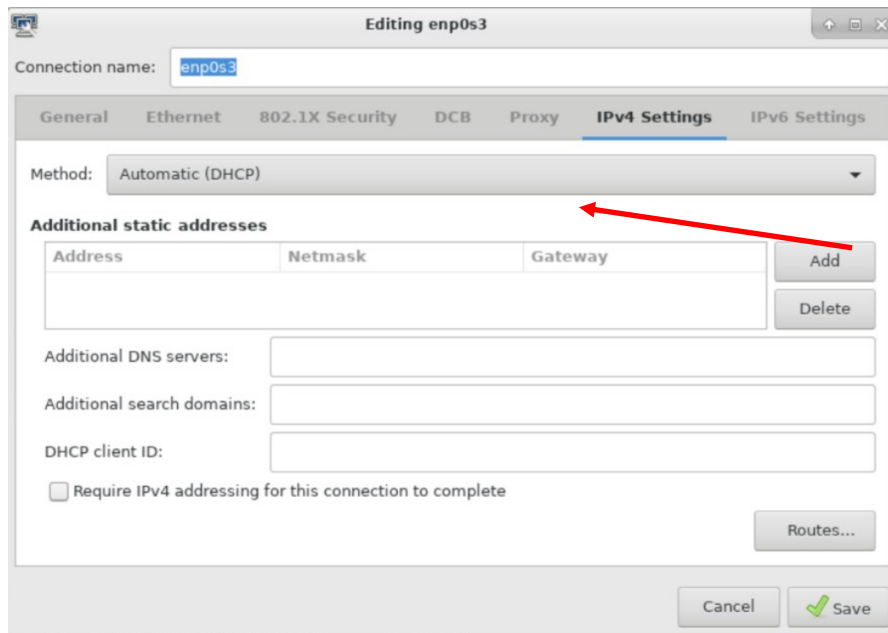


Εικόνα 6- 2: Απεικόνιση των δικτυακών διεπαφών του Server

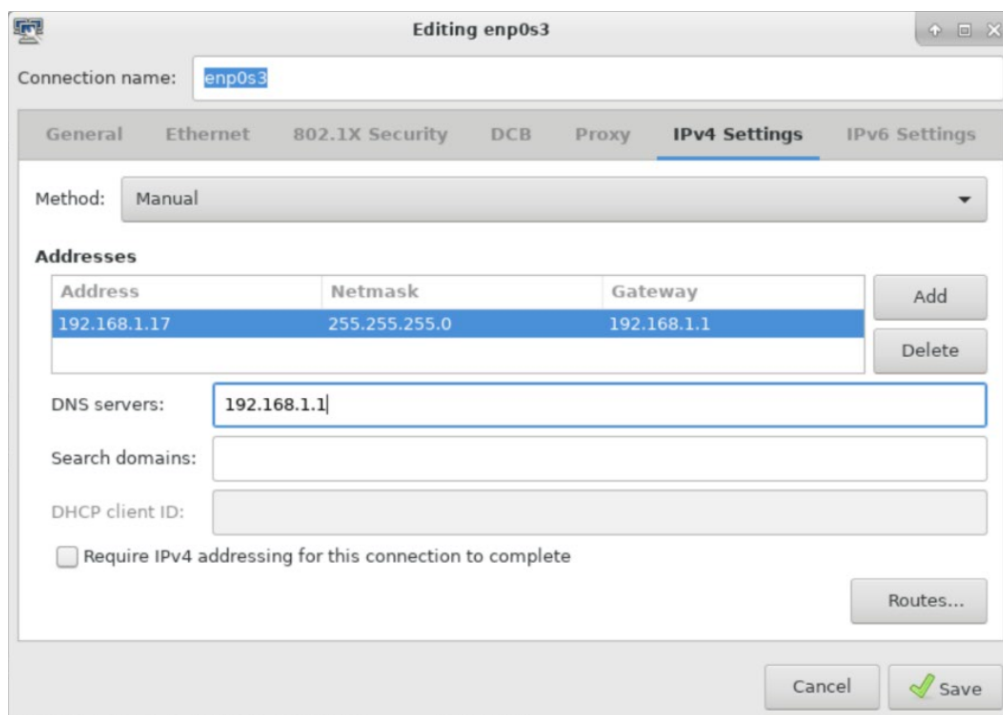
Με την ένδειξη eth0 εννοείται η κάρτα δικτύου με φυσική σύνδεση καλωδίου Ethernet. Αν υπάρχουν δύο τέτοιες κάρτες δικτύου τότε θα βλέπουμε ένα **eth0** και ένα δεύτερο **eth1**. Αν επιπλέον υπάρχει και ασύρματη (WiFi) δικτυακή κάρτα τότε θα βλέπουμε και ένα τρίτο interface με μια ένδειξη για παράδειγμα **enp0s3**. Ο αναγνώστης μπορεί να βρει πληροφορίες για την ονοματοδοσία των δικτυακών διεπαφών στο (FedoraDocsNamingInterface, n.d.)



Εικόνα 6- 3: Διαχείριση δικτυακής διεπαφής Server



Εικόνα 6- 4: Προσδιορισμός απόδοσης IP αυτόματα, μέσω DHCP



Εικόνα 6- 5: Προσδιορισμός απόδοσης IP μη αυτόματα, «με το χέρι»

### Ασκήσεις - Δραστηριότητες

Χρησιμοποιήστε την εφαρμογή σύμφωνα με τις οδηγίες και εκτελέστε τις εντολές της παραγράφου και δείτε τα αποτελέσματα που φαίνονται στις εικόνες

## 6.7 Command line και script Ethernet interface control

Η διαχείριση των δικτυακών διεπαφών του υπολογιστικού συστήματος που φιλοξενεί τον Linux Server μπορεί να γίνει και από τη γραμμή εντολών (command line)

Με την εντολή

```
[root@ekdda ~]# ifconfig
```

Μπορούμε να δούμε τις δικτυακές διεπαφές (network interfaces – κάρτες δικτύου). Ένα παράδειγμα εκτέλεσης της παραπάνω εντολής και μάλιστα μέσα από εικονική μηχανή η οποία βρίσκεται στις προδιαγραφές λειτουργίας του σεμιναρίου Linux Server είναι η ακόλουθη :

```
[root@ekdda ~]# ifconfig

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.8 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::3eab:52eb:192c:78af prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:58:cb:46 txqueuelen 1000 (Ethernet)
    RX packets 904 bytes 99747 (97.4 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 510 bytes 195845 (191.2 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 10 bytes 964 (964.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 10 bytes 964 (964.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

virbr0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    inet 192.168.124.1 netmask 255.255.255.0 broadcast 192.168.124.255
    ether 52:54:00:01:68:ee txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Στην παραπάνω έξοδο παρατηρούμε ότι υπάρχουν τόσο πραγματικές όσο και εικονικές δικτυακές διεπαφές. Στην 1<sup>η</sup> γραμμή εξόδου παρατηρούμε ότι δίνονται πληροφορίες για τη

διεπαφή (interface) eth0 (συμπληρωματικά δείτε και την παράγραφο 6.6). Στη 2<sup>η</sup> γραμμή αναφέρεται και η διεύθυνση IP του interface, η οποία είναι και η IP του συστήματός μας (αν έχουμε ένα δικτυακό interface).

Στο δεύτερο τμήμα αναφέρεται ένα εικονικό interface με το όνομα lo (αυτό σημαίνει loopback). Την διεύθυνση loopback την είδαμε ήδη στην παράγραφο 6.2, όπου αναφέραμε ότι αντιστοιχεί στην IP διεύθυνση 127.0.0.0 και χρησιμοποιείται από το σύστημά μας για να στέλνει μηνύματα πίσω στον εαυτό του (ανατροφοδότηση).

Στο τρίτο τμήμα βλέπουμε μια εικονική διεπαφή και μάλιστα bridge με το όνομα virbr0 (virtual bridge). Η διεπαφή αυτή είναι ο ενδιάμεσος μεταξύ της εικονικής μηχανής που χρησιμοποιούμε για τις ανάγκες του σεμιναρίου και της πραγματικής κάρτας γραφικών, ώστε η εικονική μηχανή να έχει πρόσβαση στο Internet και γενικότερα στο εξωτερικό δίκτυο.

### Ασκήσεις - Δραστηριότητες

- Βρείτε την IP του Linux Server σας
- Ανακαλύψτε πόσες και ποιες δικτυακές διεπαφές υπάρχουν
- Μπορείτε να καταλάβετε εάν ο Linux Server σας είναι εγκατεστημένος σε εικονική μηχανή ;

## 6.8 Δικτυακά πρωτόκολλα

Στην παράγραφο αυτή θα περιγράψουμε κάποια βασικά στοιχεία από τα δυο βασικά πρωτόκολλα που υπάρχουν σήμερα και χρησιμοποιούνται από τις δικτυακές υπηρεσίες σε έναν Linux Server. Το TCP και το UDP

### 6.8.1 Το πρωτόκολλο TCP

Το Πρωτόκολλο Ελέγχου Μετάδοσης/Πρωτόκολλο Διαδικτύου (TCP/IP όπως συνήθως αναφέρεται), είναι μια συλλογή πρωτοκόλλων επικοινωνίας στα οποία βασίζεται το Διαδίκτυο αλλά και μεγάλο ποσοστό των εμπορικών δικτύων. Η ονομασία TCP/IP προέρχεται από τις συντομογραφίες των δυο κυριότερων πρωτοκόλλων της συλλογής: το Πρωτόκολλο Ελέγχου Μετάδοσης TCP (Transfer Control Protocol) και το πρωτόκολλο Διαδικτύου IP (Internet Protocol). (Andrew S Tanenbaum, 1996).

Αυτή η συλλογή πρωτοκόλλων είναι οργανωμένη σε επίπεδα. Το καθένα τους απαντά σε συγκεκριμένα προβλήματα μεταφοράς δεδομένων και παρέχει μια καθορισμένη υπηρεσία στα υψηλότερα επίπεδα. Τα ανώτερα επίπεδα είναι πιο κοντά στη λογική του χρήστη και εξετάζουν πιο αφηρημένα δεδομένα. Στηρίζονται στα πρωτόκολλα των χαμηλότερων επιπέδων για την μετάφραση δεδομένων σε μορφές οι οποίες είναι δυνατόν να διαβιβαστούν με φυσικά μέσα. (WikiIPProtocol, n.d.)

Τα τρία ανώτερα επίπεδα του μοντέλου OSI (Εφαρμογής, Παρουσίασης και Συνεδρίας, δείτε και την παράγραφο 6.1) αποτελούν ένα ενιαίο επίπεδο στο TCP/IP, το επίπεδο Εφαρμογής. Τα χαρακτηριστικά του στρώματος Συνεδρίας αναλαμβάνονται από τις ίδιες εφαρμογές ή απλώς αγνοούνται. Ένα απλουστευμένο σχεδιάγραμμα της στοίβας του μοντέλου TCP/IP είναι το εξής:

| Επίπεδα |           | Περιγραφή Λειτουργίας  |
|---------|-----------|--|
| 1       | Εφαρμογής | π.χ. HTTP, FTP, DNS - (πρωτόκολλα δρομολόγησης, όπως το RIP, τα οποία βασίζονται στο πρωτόκολλο UDP, μπορούν επίσης να ταξινομηθούν στο στρώμα Δικτύου).   |
| 2       | Μεταφοράς | π.χ. TCP, UDP, RTP - (πρωτόκολλα δρομολόγησης, όπως το OSPF, που λειτουργούν πάνω από το IP, μπορούν επίσης να καταχωρηθούν στο στρώμα Δικτύου).   |
| 3       | Δικτύου   | Πρωτόκολλο IP - (Τα πρωτόκολλα ICMP και IGMP, παρ'όλο που βασίζονται πάνω στο IP για την λειτουργία τους, καταχωρούνται στο στρώμα Δικτύου. Το ARP αποτελεί μια από τις ολιγάριθμες εξαιρέσεις, εφ'όσον είναι ανεξάρτητο του IP) |
| 4       | Συνδέσμου | π.χ. Ethernet, Token Ring, κλπ.  |

Πίνακας 6- 2: Μοντέλο Διαστρωμάτωσης (ή Αναφοράς) του Internet

### 6.8.2 Το πρωτόκολλο UDP

Το πρωτόκολλο **User Datagram Protocol (UDP)** είναι ένα από τα βασικά πρωτόκολλα που χρησιμοποιούνται στο Διαδίκτυο. Χρησιμοποιεί ένα απλό μοντέλο επικοινωνίας χωρίς σύνδεση (connection less) και χωρίς πολλές απαιτήσεις ασφάλειας στην επικοινωνία. Διάφορα προγράμματα χρησιμοποιούν το πρωτόκολλο UDP για την αποστολή σύντομων μηνυμάτων (γνωστών και ως datagrams) από τον έναν υπολογιστή στον άλλον μέσα σε ένα δίκτυο υπολογιστών. Ένα από τα κύρια χαρακτηριστικά του UDP είναι ότι δεν εγγυάται αξιόπιστη επικοινωνία. Τα πακέτα UDP που αποστέλλονται από έναν υπολογιστή μπορεί να φτάσουν στον παραλήπτη με λάθος σειρά, διπλά ή να μην φτάσουν καθόλου εάν το δίκτυο έχει μεγάλο φόρτο. Αντίθετα με το πρωτόκολλο TCP το οποίο διαθέτει όλους τους απαραίτητους μηχανισμούς ελέγχου και επιβολής της αξιοπιστίας και συνεπώς μπορεί να εγγυηθεί την αξιόπιστη επικοινωνία μεταξύ των υπολογιστών. Η έλλειψη των μηχανισμών αυτών από το πρωτόκολλο UDP το καθιστά αρκετά πιο γρήγορο και αποτελεσματικό, τουλάχιστον για τις εφαρμογές εκείνες που δεν απαιτούν «και τόσο» αξιόπιστη επικοινωνία. Το UDP ονομάζεται πρωτόκολλο «καλύτερης προσπάθειας» (best effort) στο οποίο ο έλεγχος σφαλμάτων είναι αδύναμος (απλό checksum). Τα πακέτα μπορεί να χαθούν ή να παραδοθούν αλλού. Επίσης δεν έχει διαλόγους χειραψίας (handshaking) και έτσι εκθέτει τα δεδομένα σε κίνδυνο. Για το λόγο αυτό χρησιμοποιείται από υπηρεσίες που είτε τα δεδομένα τους δεν είναι κρίσιμα, είτε μπορούν να διορθωθούν σχετικά εύκολα, είτε είναι αδιάφορα αν κάποια πληροφορία χαθεί.



**handshaking** είναι η διαδικασία κατά την οποία επιβεβαιώνεται ότι η σύνδεση μεταξύ Client και Server είναι ασφαλής και επιβεβαιωμένο ότι δε θα χαθούν δεδομένα στην πορεία. Πριν να προσπαθήσει ένα πρόγραμμα-πελάτης (Client) να συνδεθεί με έναν server, ο server πρέπει πρώτα να δεσμεύσει μια πόρτα και να την ανοίξει ώστε να δέχεται συνδέσεις. Αυτό καλείται *passive open*. Όταν γίνει αυτό, ο Client μπορεί να αρχίσει τη σύνδεση (*active open*). Για να γίνει μια σύνδεση, γίνεται μια "χειραψία" ανάμεσα στα συμμετέχοντα μέρη, το λεγόμενο **three-way handshake (το οποίο χρησιμοποιεί το TCP)**, κατά το οποίο ο Client και ο Server γνωρίζονται και καθορίζουν πώς θα επιβεβαιώνει ο ένας τον άλλο ότι ένα πακέτο στάλθηκε από τη μεριά και παραλείφθηκε από την άλλη ακέραιο.

Οι εφαρμογές audio και video streaming χρησιμοποιούν ως επί το πλείστον πακέτα UDP. Για τις εφαρμογές αυτές είναι πολύ σημαντικό τα πακέτα να παραδοθούν στον παραλήπτη σε σύντομο χρονικό διάστημα ούτως ώστε να μην υπάρχει διακοπή στην ροή του ήχου ή της εικόνας. Κατά συνέπεια προτιμάται το πρωτόκολλο UDP διότι είναι αρκετά γρήγορο, παρόλο που υπάρχει η πιθανότητα μερικά πακέτα UDP να χαθούν. Στην περίπτωση που χαθεί κάποιο πακέτο, οι εφαρμογές αυτές διαθέτουν ειδικούς μηχανισμούς διόρθωσης και παρεμβολής έτσι ώστε ο τελικός χρήστης να μην παρατηρεί καμία αλλοίωση ή διακοπή στην ροή του ήχου και της εικόνας λόγω του χαμένου πακέτου. Σε αντίθεση με το πρωτόκολλο TCP, το UDP υποστηρίζει *broadcasting*, δηλαδή την αποστολή ενός πακέτου σε όλους τους υπολογιστές ενός δικτύου, και *multicasting*, δηλαδή την αποστολή ενός πακέτου σε κάποιους συγκεκριμένους υπολογιστές ενός δικτύου. Η τελευταία δυνατότητα χρησιμοποιείται πολύ συχνά στις εφαρμογές audio και video streaming, έτσι ώστε μία ροή ήχου ή εικόνας να μεταδίδεται ταυτόχρονα σε πολλούς συνδρομητές. Μερικές σημαντικές υπηρεσίες που χρησιμοποιούν πακέτα UDP είναι οι εξής: Domain Name System (DNS), IPTV, Voice over IP (VoIP), Trivial File Transfer Protocol (TFTP) και τα παιχνίδια που παίζονται ζωντανά μέσω του Διαδικτύου. (CISCO\_IP\_Networking) (CISCO\_SNMP, n.d.)

### 6.8.3 Το πρωτόκολλο SNMP

Το **Simple Network Management Protocol (SNMP)** είναι μέρος της σουίτας πρωτοκόλλων διαδικτύου (IP - Internet Protocol), όπως έχει ορισθεί από το Internet Engineering Task Force (IETF). Χρησιμοποιείται στα συστήματα διαχείρισης δικτύων, στη διαχείριση και παρακολούθηση δικτυακών συσκευών που απαιτούν παρέμβαση του διαχειριστή δικτύου. Αποτελείται από μια ομάδα προτύπων για τη διαχείριση δικτύου και περιλαμβάνει ένα πρωτόκολλο επιπέδου εφαρμογών (*application layer*), ένα σχήμα βάσης δεδομένων και μια ομάδα από σύνολα δεδομένων. (IETF\_Internet\_Engineering\_Task\_Force, n.d.) (CISCO\_SNMP, n.d.)

Το SNMP είναι ένα πρωτόκολλο του επιπέδου εφαρμογών (δείτε περισσότερα για τα επίπεδα στις παραγράφους 6.1 και 6.8.1) του οποίο διευκολύνει την ανταλλαγή πληροφοριών διαχείρισης μεταξύ των συσκευών του δικτύου. Είναι μέρος του TCP/IP και

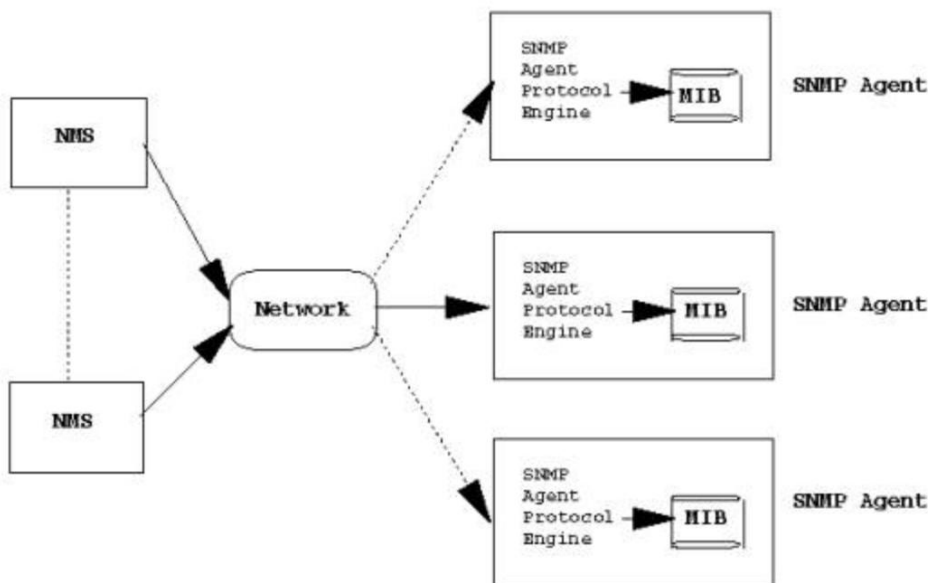
επιτρέπει στους διαχειριστές να παρακολουθούν την απόδοσή του και να επιλύουν τα προβλήματα που εμφανίζονται (CONTA\_Lab, n.d.) (CISCO\_SNMP, n.d.)

Ένα δίκτυο το οποίο διαχειρίζεται με SNMP έχει τρία βασικά στοιχεία: διαχειριζόμενες συσκευές, πράκτορες (agents) και συστήματα διαχείρισης δικτύου (Network Management Systems- NMS).

Μια διαχειριζόμενη συσκευή είναι ένας κόμβος του δικτύου ο οποίος περιέχει ένα SNMP πράκτορα και βρίσκεται μέσα στο διαχειριζόμενο δίκτυο. Οι διαχειριζόμενες συσκευές συλλέγουν και αποθηκεύουν πληροφορίες και τις διαθέτουν στο σύστημα διαχείρισης του δικτύου με χρήση του SNMP. Τέτοιες συσκευές είναι οι δρομολογητές, οι γέφυρες, οι διακόπτες (interrupts). (CONTA\_Lab, n.d.) (CISCO\_SNMP, n.d.)

Ένας πράκτορας έχει γνώση των τοπικών πληροφοριών διαχείρισης και τις μετατρέπει σε μορφή που είναι συμβατή με το SNMP. Στη συνέχεια ένα σύστημα διαχείρισης δικτύου εκτελεί εφαρμογές οι οποίες παρακολουθούν και ελέγχουν τις διαχειριζόμενες συσκευές. Το σύστημα διαχείρισης του δικτύου προσφέρει τον κύριο όγκο των πόρων επεξεργασίας που απαιτούνται για τη διαχείριση. (CONTA\_Lab, n.d.) (CISCO\_SNMP, n.d.)

Η αρχιτεκτονική του SNMP φαίνεται την παρακάτω Εικόνα 6- 6.



Εικόνα 6- 6: Αρχιτεκτονική SNMP - εικόνα από το (CONTA\_Lab, n.d.)

Στη βασική του μορφή, το SNMP προσφέρει στα εργαλεία παρακολούθησης του συστήματος, δεδομένα υπολογιστικών πόρων ως μεταβλητές, όπως πχ. "free memory", "system name", "number of running processes", "default route". Οι μεταβλητές αυτές είναι τα απεικονιζόμενα MIBs στην παραπάνω Εικόνα 6- 6 κλπ. Ταυτόχρονα, επιτρέπει ενέργειες όπως η εφαρμογή νέας ή η αλλαγή της υπάρχουσας παραμετροποίησης της δικτυακής διάταξης.

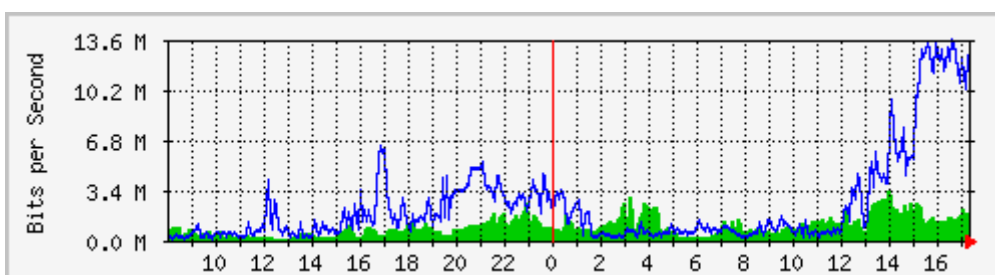
Όπως είπαμε, το SNMP λειτουργεί στο επίπεδο εφαρμογής της σουίτας πρωτοκόλλου Διαδικτύου. Όλα τα μηνύματα SNMP μεταφέρονται μέσω του User Datagram Protocol (UDP). Ο πράκτορας SNMP λαμβάνει αιτήματα στη θύρα UDP 161. Ο διαχειριστής μπορεί να στείλει αιτήματα από οποιαδήποτε διαθέσιμη θύρα προέλευσης στη θύρα 161 στον πράκτορα. Η απάντηση του πράκτορα αποστέλλεται πίσω στη θύρα προέλευσης του διαχειριστή. Ο διαχειριστής λαμβάνει ειδοποιήσεις (Traps and InformRequests) στη θύρα 162. Ο πράκτορας μπορεί να δημιουργήσει ειδοποιήσεις από οποιαδήποτε διαθέσιμη θύρα. Όταν χρησιμοποιείται με το Security Layer Security ή το Datagram Transport Layer Security, λαμβάνονται αιτήματα στη θύρα 10161 και οι ειδοποιήσεις αποστέλλονται στη θύρα 10162 (CISCO\_SNMP, n.d.)



Το **community string** είναι ένα «password» με το οποίο ένας πελάτης (πχ ένα πρόγραμμα απεικόνισης δραστηριότητας (monitoring tool) μπορεί να έχει πρόσβαση στις πληροφορίες του πρωτοκόλλου SMNP

Όπως είδαμε Το SNMP χρησιμοποιεί το πρωτόκολλο UDP για να μεταφέρει την πληροφορία του. Είδαμε στην προηγούμενη παράγραφο ότι το UDP είναι ένα μη ασφαλές πρωτόκολλο. Επειδή λοιπόν πρόκειται για ένα πρωτόκολλο του οποίου το μέλημα δεν είναι να μη χαθεί όλη η πληροφορία που μεταφέρεται, χρησιμοποιείται και σε περιπτώσεις απεικόνισης κάποιων στοιχείων του συστήματος και της κίνησης στα δικτυακά μέρη του. Πράγματι δεν είναι και πολύ σημαντικό αν για κάποια δευτερόλεπτα χαθεί η απεικόνιση της κατάστασης της μνήμης ή της κίνησης που έχει στην κάρτα δικτύου, γιατί έχει σημασία η γενικότερη εικόνα τους.

Με άλλα λόγια σε ένα διάγραμμα της μορφής της **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.** του οποίου η πληροφορία μεταφέρεται από το τα πρωτόκολλα SNMP και UDP δεν έχει σημασία αν λείπει μια μικρή γραμμούλα, γιατί οι υπόλοιπες δείχνουν καθαρά την πραγματική κατάσταση του απεικονιζόμενου στοιχείου.



Εικόνα 6- 7: Απεικόνιση πληροφορίας που μεταφέρεται με το πρωτόκολλο UDP

Το SNMP χρησιμοποιείται σε αρκετά εργαλεία απεικόνισης (monitoring) όπως τα MRTGs, το CACTI, το NAGIOS, το ZABBIX και το MUNIN.

Η εγκατάστασή του γίνεται με την εγκατάσταση των πακέτων **net-snmp net-snmp-utils php-snmp**



```
[root@ekdda ~]# dnf -y install net-snmp net-snmp-utils php-snmp
```

Στη συνέχεια το πρωτόκολλο πρέπει να παραμετροποιηθεί

```
[root@ekdda ~]# gedit /etc/snmp/snmpd.conf
```

στη γραμμή 41: αφαιρούμε το σχόλιο (comment out)

```
com2sec notConfigUser default public
```



Το **uncomment** ή **comment out** είναι η αφαίρεση σχολίου από μια γραμμή. Με τον όρο **αφαίρεση σχολίου** στα αρχεία του linux εννοούμε την αφαίρεση του χαρακτήρα **sharp (#)** από το μπροστινό μέρος της γραμμής. Όταν υπάρχει το σχόλιο (#) στο μπροστινό μέρος της γραμμής, η γραμμή αυτή δε λαμβάνεται υπόψη στο διάβασμα του αρχείου από τις διεργασίες του συστήματος. Με απλά λόγια **οι γραμμές με σχόλιο δεν εκτελούνται.**

στις γραμμές 74 και 75: προσδιορίζουμε το community string στο τοπικό μας δίκτυο. Εδώ το communication string είναι το **ekddalinux**

```
# change "mynetwork" to your own network
# change community name to anyone except public, private (for
security reason)
com2sec local localhost ekddalinux
com2sec mynetwork 192.168.1.0/24 ekddalinux
```

στις γραμμές 78 και 79: αφαιρούμε το σχόλιο και αλλάζουμε ως εξής :

```
group MyRWGroup v2c local
group MyROGroup v2c mynetwork
```

στη γραμμή 85: αφαιρούμε το σχόλιο και αλλάζουμε ως εξής :

```
view all included .1 80
```

στις γραμμές 93 και 94: αφαιρούμε το σχόλιο και αλλάζουμε ως εξής :

```
access MyROGroup "" v2c noauth exact all none none
access MyRWGroup "" v2c noauth exact all all all
```

Στη συνέχεια αποθηκεύουμε και κλείνουμε το αρχείο

Στη συνέχεια εκκινούμε την υπηρεσία snmpd και αν θέλουμε την κάνουμε enable (δείτε στην παράγραφο 4.1 για τη χρήση της παραμέτρου enable)

```
[root@ekdda ~]# systemctl start snmpd
[root@ekdda ~]# systemctl enable snmpd
```

Με την παρακάτω εντολή κάνουμε πιστοποίηση (validation) του community string

```
[root@ekdda ~]# snmpwalk -v2c -c ekddalinux localhost system
SNMPv2-MIB::sysDescr.0 = STRING: Linux dlp.linux.server 3.10.0-
229.4.2.el7.x86_64 #1 SMP Wed May.....
SNMPv2-MIB::sysObjectID.0 = OID: NET-SNMP-
MIB::netSnmAgentOIDs.10
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (91954)
0:15:19.54
SNMPv2-MIB::sysContact.0 = STRING: Root <root@localhost>
(configure /etc/snmp
.....
.....
SNMPv2-MIB::sysORUpTime.9 = Timeticks: (4) 0:00:00.04
SNMPv2-MIB::sysORUpTime.10 = Timeticks: (4) 0:00:00.04
```

Το πρωτόκολλο πλέον θεωρείται έτοιμο να αποστείλει στους κατάλληλους Clients (τις εφαρμογές δηλαδή οπτικοποίησης και παρακολούθησης δραστηριότητας (monitoring tools), τις πληροφορίες που χρειάζονται. Στην παράγραφο 7.2 θα δούμε τέτοια εργαλεία

### *Ασκήσεις - Δραστηριότητες*

Πραγματοποιήστε την εγκατάσταση του πρωτοκόλλου σύμφωνα με τις οδηγίες και εκτελέστε τις εντολές της παραγράφου και δείτε τα αποτελέσματα που φαίνονται στις εικόνες

## 7 Ασφάλεια Δικτύου

Η ασφάλεια του δικτύου κι η προστασία του Linux Server και των χρηστών του από κακόβουλες ενέργειες αποτελεί ίσως την πιο σοβαρή και κρίσιμη δραστηριότητα του διαχειριστή.

Σε καθημερινή βάση ένας Server δέχεται δεκάδες επιθέσεις από διάφορους επίδοξους εισβολείς (intruders). Αυτές οι επιθέσεις μπορούν να είναι απλές (ρουτίνας) ή πολύ σοβαρές. Οι επιθέσεις ρουτίνας γίνονται συνήθως με κάποια αυτοματοποιημένη διαδικασία και αυτό που αναζητούν οι επίδοξοι εισβολείς είναι συστήματα τα οποία έχουν χαλαρή ασφάλεια με διαχειριστές (ή και χωρίς διαχειριστές) οι οποίοι δεν λαμβάνουν σοβαρά μέτρα ασφάλειας. Οι σοβαρές επιθέσεις είναι πιο δύσκολα αντιμετωπίσιμες και συνήθως είναι στοχευμένες με σκοπό να εισέλθουν οι εισβολείς σε ένα σύστημα με σκοπό να κλέψουν ή να καταστρέψουν δεδομένα και πληροφορίες ή να σταματήσουν τις υπηρεσίες που προσφέρουν οι συγκεκριμένοι Servers.

Σε κάθε περίπτωση ο διαχειριστής ανάλογα και με τη σοβαρότητα και τη σπουδαιότητα των δεδομένων και της πληροφορίας που διαχειρίζονται και της υπηρεσίας που προσφέρουν θα πρέπει να είναι πολύ προσεκτικοί, να ενημερώνονται διαρκώς από τα επίσημα forums της διανομής που χρησιμοποιούν και να έχουν (ή και να αναπτύσσουν) τα απαραίτητα εργαλεία προκειμένου να επιβλέπουν των Server που διαχειρίζονται και να μπορούν να αντιμετωπίζουν τις επιθέσεις που δέχονται.

Υπάρχουν τρία πιο συνηθισμένα ήδη επιθέσεων, τα οποία ανάλογα την περίπτωση μπορούν να είναι ρουτίνας ή να εξελιχθούν σε πολύ σοβαρά ήδη επιθέσεων αν ο διαχειριστής δεν είναι σε ετοιμότητα.

α) **Brute Force** (Ωμής Βίας) : Στην περίπτωση αυτού του είδους επίθεσης γίνεται προσπάθεια εισβολής στον Linux Server με στόχο κάποιο πρωτόκολλο μέσω ονόματος χρήστη και κωδικού πρόσβασης (username και password). Στην ουσία οι επίδοξοι εισβολείς προσπαθούν να ανακαλύψουν το συνθηματικό, δοκιμάζοντας πιθανούς συνδυασμούς.

β) **DOS** (Denial of Service Attack – Επίθεση Άρνησης Υπηρεσίας)

γ) **Exploits Attacks** (Επιθέσεις Εκμετάλλευσης)

Υπάρχουν και άλλα ήδη επιθέσεων οι οποίες όμως μπορούμε να πούμε ότι είναι πιο ειδικές και αφορούν πιο εξειδικευμένες περιπτώσεις επίθεσης. Για παράδειγμα

- Wifi attacks (πρόσβαση και έλεγχος κάποιας ασύρματης συσκευής επικοινωνίας)
- Watering Hole Attacks (εκμεταλλεύεται τις ενέργειες μια ομάδας χρηστών ενός εταιρικού website προκειμένου να το «τρυπήσει» με απώτερο σκοπό τον έλεγχο του συστήματος της εταιρείας κτλ

Ο διαχειριστής του Linux Server πρέπει να εκτελεί διαδικασίες ενημέρωσης του λογισμικού συστήματος όποτε αυτές είναι έτοιμες, καθώς και των εφαρμογών και των διαφόρων

υπηρεσιών και εξυπηρετητών αφού ενημερωθεί από το επίσημο site και τα forums της διανομής για πιο λόγο εκδόθηκε το update. Οι περισσότερες σοβαρές επιθέσεις αφορούν τρύπες του συστήματος (system holes) και μιλώντας γενικότερα οι περισσότερες επιτυχημένες διεισδύσεις βασίζονται σε αυτές και όχι στην «έμπνευση» των hackers.

## 7.1 Ασφάλεια Δικτυακών Υπηρεσιών

Στις επόμενες παραγράφους περιγράφουμε διάφορα εργαλεία και εφαρμογές που μας βοηθούν στην ασφάλεια των δικτυακών μας υπηρεσιών από επίδοξους εισβολείς.

### 7.1.1 SELinux

Το Security Enhanced Linux είναι ένα σύνολο αρθρωμάτων (modules) τα οποία προσθέτουν ακόμη ένα επίπεδο ασφάλειας στον Linux Server, λογικά τοποθετημένο, μπροστά από το firewall και ως πρώτη ασπίδα ασφαλείας ανάμεσα στον Linux Server και το εξωτερικό δίκτυο. Το SELinux εκτός από τις διανομές RedHat (Fedora, Centos, RH enterprise edition κτλ) υπάρχει και για άλλες διανομές Linux.



Το **runtime session** που αναφέρεται πολλές φορές σε όλο το κείμενο είναι η τρέχουσα περίοδος λειτουργίας του συστήματός μας. Δηλαδή από την εκκίνηση, μέχρι τον τερματισμό ή την επανεκκίνηση

Το SELinux προσφέρει εμπλουτισμένη ασφάλεια στον Linux Server μας. Περιέχει ένα σύνολο από περίπου 500 διακόπτες οι οποίοι επιτρέπουν ή όχι τη χρήση κάποιων υπηρεσιών ή ενεργειών.

Μπορούμε να βρούμε μια πλήρη λίστα όλων των διακοπών (SELinux boolean list) στο (SelinuxBooleans, n.d.)

Για παράδειγμα υπάρχει διακόπτης που επιτρέπει ή όχι τη χρήση του /home directory ή τη χρήση του /public\_html κάτω από κάθε home directory για τον web Server

```
setsebool httpd_enable_homedirs true
```

ακόμη ένα παράδειγμα, ο διακόπτης για το αν μπορεί το σύστημά μας να συνδεθεί στο δίκτυο

```
setsebool httpd_can_network_connect=on
```

με την ακόλουθη εντολή βλέπουμε όλους τους διακόπτες καθώς και την κατάσταση στην οποία βρίσκονται (on ή off)

```
[root@ekdda ~]# getsebool -a | less
```

Η έξοδος της προηγούμενης εντολής φαίνεται στην Εικόνα 7- 1

```

abrt_anon_write --> off
abrt_handle_event --> off
abrt_upload_watch_anon_write --> on
antivirus_can_scan_system --> off
antivirus_use_jit --> off
auditadm_exec_content --> on
authlogin_nsswitch_use_ldap --> off
authlogin_radius --> off
authlogin_yubikey --> off
awstats_purge_apache_log_files --> off
boinc_execmem --> on
cdrecord_read_content --> off
cluster_can_network_connect --> off
cluster_manage_all_files --> off
cluster_use_execmem --> off
cobbler_anon_write --> off
cobbler_can_network_connect --> off
cobbler_use_cifs --> off
cobbler_use_nfs --> off
collectd_tcp_network_connect --> off
condor_tcp_network_connect --> off
conman_can_network --> off
:

```

Εικόνα 7- 1: Λίστα διακοπών SELinux

με την ακόλουθη εντολή βλέπουμε όλους τους διακόπτες που περιέχουν τη λέξη - υπηρεσία «httpd» και αντίστοιχα αν είναι on ή off

```
[root@ekdda ~]# getsebool -a | grep httpd
```

Η έξοδος της προηγούμενης εντολής φαίνεται στην Εικόνα 7- 2

```

httpd_execmem --> off
httpd_graceful_shutdown --> on
httpd_manage_ipa --> off
httpd_mod_auth_ntlm_winbind --> off
httpd_mod_auth_pam --> off
httpd_read_user_content --> off
httpd_run_ipa --> off
httpd_run_preupgrade --> off
httpd_run_stickshift --> off
httpd_serve_cobbler_files --> off
httpd_setrlimit --> off
httpd_ssi_exec --> off
httpd_sys_script_anon_write --> off
httpd_tmp_exec --> off
httpd_tty_comm --> off
httpd_unified --> off
httpd_use_cifs --> off
httpd_use_fusefs --> off
httpd_use_gpg --> off
httpd_use_nfs --> off
httpd_use_openstack --> off
httpd_use_sasl --> off
httpd_verify_dns --> off
[root@mail ~]#

```

Εικόνα 7- 2: Διακόπτες που περιέχουν την υπηρεσία httpd

με τις ακόλουθες εντολές μπορούμε να ανοίξουμε έναν διακόπτη

```
[root@ekdda ~]# setsebool -P httpd_can_network_connect on
```

ή

```
[root@ekdda ~]# # setsebool -P httpd_can_network_connect 1
```

και με τις ακόλουθες να τον κλείσουμε

```
[root@ekdda ~]# setsebool -P httpd_can_network_connect off
```

ή

```
[root@ekdda ~]# # setsebool -P httpd_can_network_connect 0
```

Μπορούμε να διαπιστώσουμε την κατάσταση στην οποία βρίσκεται το SELinux εκτελώντας την εντολή

```
[root@ekdda ~]# sestatus

SELinux status:                enabled
SELinuxfs mount:              /sys/fs/selinux
SELinux root directory:       /etc/selinux
Loaded policy name:            targeted
Current mode:                  permissive
Mode from config file:         error (Success)
Policy MLS status:             enabled
Policy deny_unknown status:    allowed
Max kernel policy version:     31
```

Το SELinux μπορεί να απενεργοποιηθεί αν ανοίξουμε το αρχείο παραμετροποίησης

```
[root@ekdda ~]# gedit /etc/selinux/config
```

ή

```
[root@ekdda ~]# gedit /etc/sysconfig/selinux
```

Και στη γραμμή 7 αλλάζουμε το enforcing σε disabled (Εικόνα 7- 3)

```

1
2 # This file controls the state of SELinux
3 # SELINUX= can take one of three settings:
4 #     enforcing - SELinux security is enforced.
5 #     permissive - SELinux prints warnings instead of enforcing.
6 #     disabled - No SELinux policy is loaded.
7 SELINUX=enforcing
8 # SELINUX=disabled
9 # SELINUXTYPE= can take one of three settings:
10 #     targeted - Targeted processes only have SELinux security enforcing.
11 #     minimum - Modification of targeted policy. Only selected processes are
12 #     mls - Multi Level Security.
13 SELINUXTYPE=targeted
14

```

Εικόνα 7- 3: Ενεργοποίηση και απενεργοποίηση του SELinux

Μετά την αλλαγή χρειάζεται επανεκκίνηση του συστήματος

Αν το SELinux είναι απενεργοποιημένο και ενεργοποιηθεί, στην επανεκκίνηση δημιουργεί μια διαδικασία που λέγεται relabeling (Εικόνα 7- 4)

```

[ OK ] Reached target System Initialization.
Starting Relabel all filesystems...

*** Warning -- SELinux targeted policy relabel is required.
*** Relabeling could take a very long time, depending on file
*** system size and speed of hard drives.
Warning: Skipping the following R/O filesystems:
/sys/fs/cgroup
Relabeling / /boot /dev /dev/hugepages /dev/mqueue /dev/pts /dev
ernel/debug /tmp
87.8%

```

Εικόνα 7- 4: Διαδικασία Relabeling μετά την ενεργοποίηση του SELinux

Μπορούμε να απενεργοποιήσουμε το SELinux προσωρινά, για το runtime session εκτελώντας την εντολή

```
[root@ekdda ~]# setenforce 0
```

Μπορούμε να δούμε τις διαφορετικές καταστάσεις στις οποίες βρίσκεται το SELinux εκτελώντας τις εντολές

```
[root@ekdda ~]# setenforce --help
usage: setenforce [ Enforcing | Permissive | 1 | 0 ]
```

```
[root@ekdda ~]# setenforce 0
[root@mail ~]# sestatus
SELinux status:                enabled
SELinuxfs mount:              /sys/fs/selinux
SELinux root directory:       /etc/selinux
Loaded policy name:            targeted
Current mode:                  permissive
Mode from config file:         error (Success)
Policy MLS status:             enabled
Policy deny_unknown status:    allowed
Max kernel policy version:     31
```

και

```
[root@ekdda ~]# setenforce 1
[root@mail ~]# sestatus
SELinux status:                enabled
SELinuxfs mount:              /sys/fs/selinux
SELinux root directory:       /etc/selinux
Loaded policy name:            targeted
Current mode:                  enforcing
Mode from config file:         error (Success)
Policy MLS status:             enabled
Policy deny_unknown status:    allowed
Max kernel policy version:     31
```



Μεταβάλλοντας την κατάσταση του SELinux από **enforcing** (enabled) σε **permissive** (disabled) το SELinux δεν απενεργοποιείται πραγματικά όπως γίνεται με τη διαδικασία που περιγράψαμε προηγουμένως. Απενεργοποιείται μόνο για το runtime session και επιπλέον συνεχίζει να καταγράφει αναφορές προσπάθειας παραβίασης (**violations reports**).

### Ασκήσεις - Δραστηριότητες

Εκτελέστε τις εντολές της παραγράφου και δείτε τα αποτελέσματα που φαίνονται στις εικόνες

#### 7.1.2 Firewall

Το τείχος προστασίας (firewall) που φιλοξενεί η έκδοση fedora που χρησιμοποιούμε, αποτελεί μια εξελιγμένη μορφή υπηρεσίας (την **firewalld**). Εκτός του γεγονότος ότι είναι μια υπηρεσία η οποία πρέπει να εκκινείται, να είναι enable ή disable, μπορούμε να τη



σταματήσουμε κτλ. Μπορούμε να εκτελέσουμε σε αυτήν δηλαδή όλες τις λειτουργίες μιας υπηρεσίας (δείτε και παράγραφο 4.1)

```
[root@ekdda ~]# systemctl start firewalld
[root@ekdda ~]# systemctl enable firewalld
[root@ekdda ~]# systemctl stop firewalld
[root@ekdda ~]# systemctl disable firewalld
```

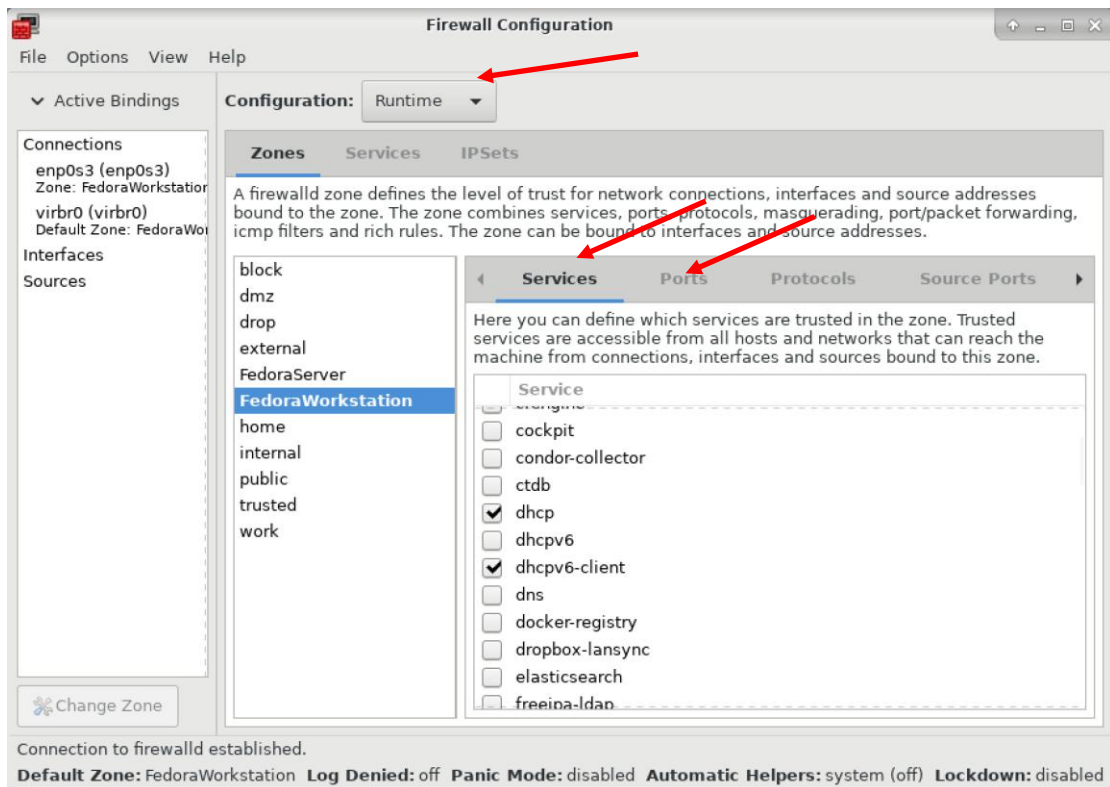
Πρέπει να τονίσουμε ότι δεν πρόκειται για το παλιό firewall το οποίο υλοποιούνταν με τα ipchains, αλλά πρόκειται για ένα εντελώς καινούριο εργαλείο με φιλοσοφία υπηρεσίας



Το **runtime session** που αναφέρεται πολλές φορές σε όλο το κείμενο είναι η τρέχουσα περίοδος λειτουργίας του συστήματός μας. Δηλαδή από την εκκίνηση, μέχρι τον τερματισμό ή την επανεκκίνηση

Έχει τη δυνατότητα τόσο γραφικής διαχείρισης, όσο και διαχείρισης command line. Επιπλέον έχει πολλές ακόμη δυνατότητες, μια από αυτές είναι η δυνατότητα masquerading και port forwarding που παρέχει ακόμη ένα επίπεδο ασφάλειας στο διαχειριστή. Η διαδικασία port forwarding περιγράφεται αναλυτικά στην επόμενη παράγραφο.

Το γραφικό του περιβάλλον διαχείρισης φαίνεται στην ακόλουθη Εικόνα 7- 5

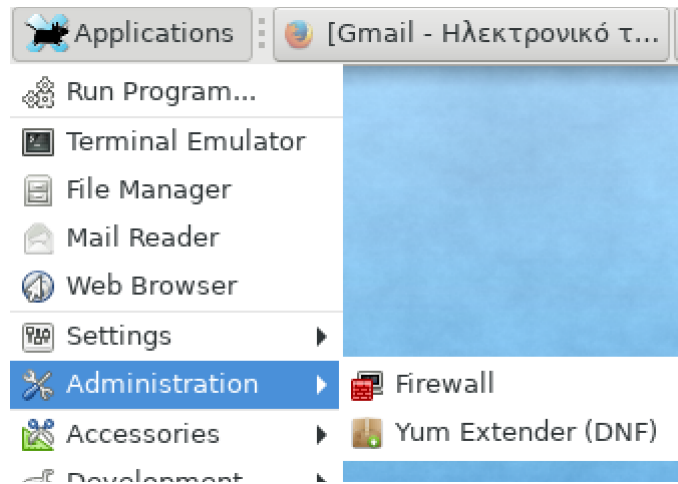


Εικόνα 7- 5: Γραφικό περιβάλλον διαχείρισης του εξελιγμένου Firewall

Και μπορούμε να το εκκινήσουμε από τη γραμμή εντολών του τερματικού με την εντολή

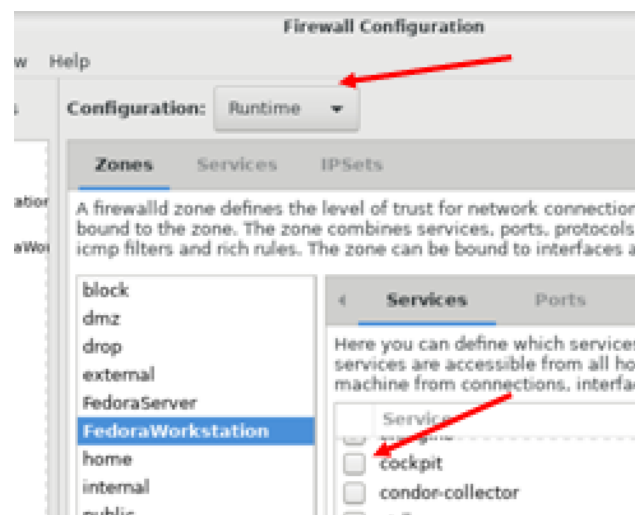
```
[root@ekdda ~]# firewall-config
```

ή να το βρούμε στο γραφικό μενού του X Window Manager. Για παράδειγμα στον xfce X Window Manager βρίσκεται στη διαδρομή Applications -> Administration -> Firewall (Εικόνα 7- 6)



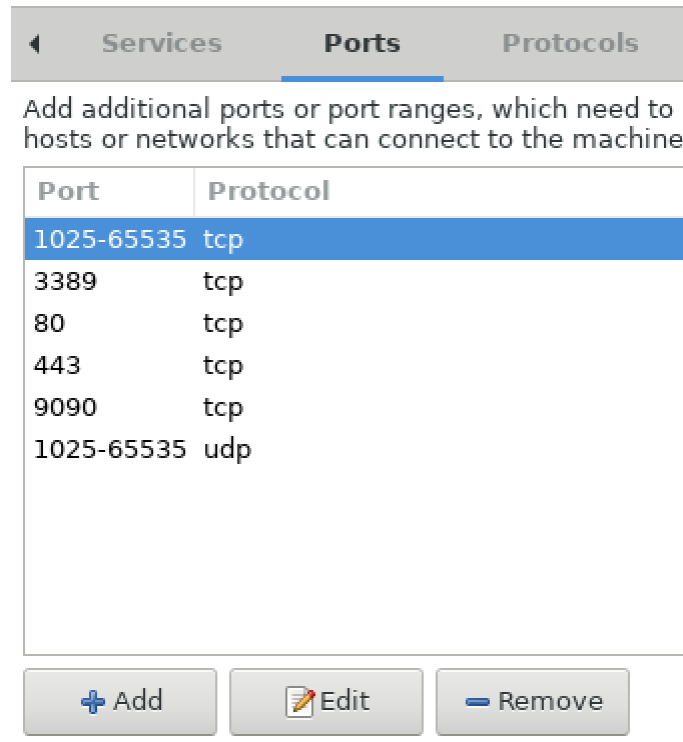
Εικόνα 7- 6: Διαδρομή για το περιβάλλον διαχείρισης του Firewall στον xfce

Μέσω του γραφικού περιβάλλοντος μπορούμε με checkboxes να ανοίξουμε ή να κλείσουμε υπηρεσίες και επίσης για το runtime session ή μόνιμα (permanent) (Εικόνα 7- 7). Αυτά τα δυο τελευταία είναι αυτά που με το systemctl έχουμε χαρακτηρίσει enable ή disable (δείτε και παράγραφο 4.1)



Εικόνα 7- 7: Ενεργοποίηση υπηρεσίας μόνιμα ή για το runtime session

Στην καρτέλα Ports (Εικόνα 7- 8) μπορούμε πάλι με checkboxes να ανοίξουμε ή να κλείσουμε υπηρεσίες και επίσης για το runtime session ή μόνιμα (permanent) (Εικόνα 7- 7).



Εικόνα 7- 8: Ενεργοποίηση πόρτας μόνιμα ή για το runtime session

Για τη διαχείριση του Firewall από τη γραμμή εντολών (command line) χρειαζόμαστε τις εντολές **firewall-cmd** (από το firewall command) που περιγράφονται αναλυτικά πιο κάτω.

Η ακόλουθη εντολή προσθέτει στις επιτρεπόμενες από το firewall πόρτες την πόρτα με το όνομα <port-service> στο πρωτόκολλο tcp και επίσης αν θέλουμε βάζουμε την παράμετρο – permanent για μόνιμη εγγραφή στο firewall

```
[root@ekdda ~]# firewall-cmd --add-port=<port service>/tcp --permanent
```



Η παράμετρος **--permanent** σημαίνει μόνιμη εγγραφή στο firewall. Δηλαδή δε χρειάζεται στην επόμενη επανεκκίνηση του συστήματος να το ξανανοίξουμε την πόρτα ή την υπηρεσία. Με την παράμετρο αυτή θα είναι μόνιμα ανοικτές. Αν θέλουμε το άνοιγμα της πόρτας ή της υπηρεσίας να ισχύει μόνο για το runtime session, δε βάζουμε καμιά παράμετρο.

Η ακόλουθη εντολή προσθέτει μόνιμα στις επιτρεπόμενες από το firewall υπηρεσίες την υπηρεσία με το όνομα <service>

```
[root@ekdda ~]# firewall-cmd --add-service=<service> --permanent
```



Πρέπει να τονίσουμε ότι δεν ανοίγουν όλες οι υπηρεσίες με τον ίδιο τρόπο. Άλλες υπηρεσίες, αρκεί να ανοίξουμε το service, σε άλλες υπηρεσίες αρκεί να ανοίξουμε την πόρτα και σε άλλες υπηρεσίες πρέπει να ανοίξουν και τα δύο.

Η ακόλουθη εντολή αφαιρεί μόνιμα από τις επιτρεπόμενες από το firewall πόρτες την πόρτα με το όνομα <port-service> στο πρωτόκολλο tcp .

```
[root@ekdda ~]# firewall-cmd --remove-port=<port service>/tcp -  
-permanent
```

Η ακόλουθη εντολή αφαιρεί μόνιμα από τις επιτρεπόμενες από το firewall υπηρεσίες την υπηρεσία με το όνομα <service> .

```
[root@ekdda ~]# firewall-cmd --remove-service=<service> --  
permanent
```

Η ακόλουθη εντολή εμφανίζει μια λίστα με τις επιτρεπόμενες από το firewall υπηρεσίες.

```
[root@ekdda ~]# firewall-cmd --list-services
```

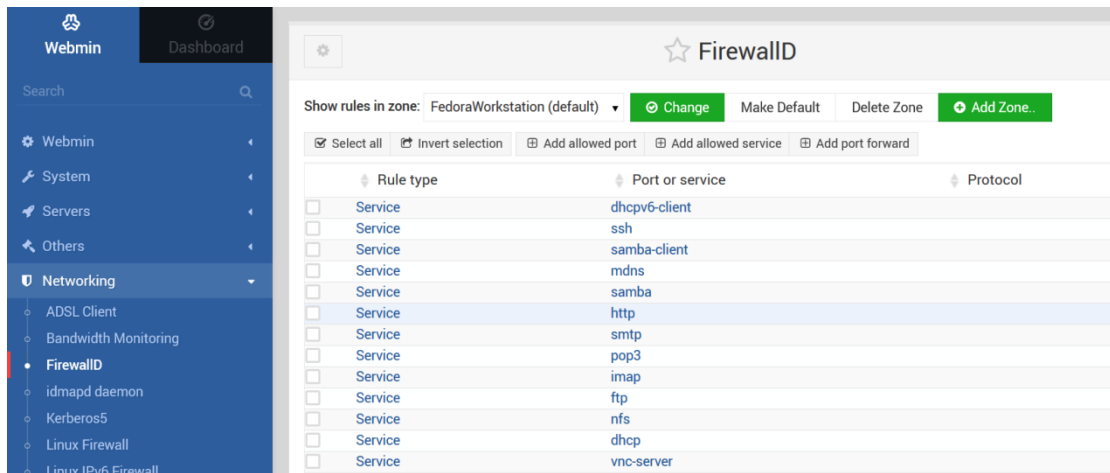
Η ακόλουθη εντολή εμφανίζει μια λίστα με τις επιτρεπόμενες από το firewall πόρτες.

```
[root@ekdda ~]# firewall-cmd --list-ports
```

Η ακόλουθη εντολή κάνει επαναφόρτωση των κανόνων του firewall και πρέπει να εκτελείται κάθε φορά που έχουμε οποιαδήποτε αλλαγή στο firewall.

```
[root@ekdda ~]# firewall-cmd -reload
```

Τέλος το firewall είναι διαχειρίσιμο και από το διαδικτυακό εργαλείο διαχείρισης webmin που περιγράψαμε στο κεφάλαιο 5. Στο webmin τη συγκεκριμένη υπηρεσία, μπορούμε να τη βρούμε στη διαδρομή Networking -> FirewallD



Εικόνα 7- 9: Διαχείριση του Firewall από το διαχειριστικό περιβάλλον Webmin

### Ασκήσεις - Δραστηριότητες

Εκτελέστε τις εντολές της παραγράφου και δείτε τα αποτελέσματα που φαίνονται στις εικόνες

#### 7.1.3 Port Forwarding

Το Port Forwarding είναι μια τεχνική με την οποία ανακατευθύνεται ένα αίτημα επικοινωνίας από έναν συνδυασμό διεύθυνσης IP και αριθμού θύρας σε έναν άλλο συνδυασμό, ενώ τα πακέτα διασχίζουν μια πύλη δικτύου, όπως ένας δρομολογητής ή ένα τείχος προστασίας. Αυτή η τεχνική χρησιμοποιείται πιο συχνά για να είναι διαθέσιμες οι υπηρεσίες από έναν κεντρικό υπολογιστή που βρίσκεται σε ένα προστατευμένο ή masquerading (μεταμφιεσμένο - καμουφλαρισμένο) εσωτερικό δίκτυο, σε Clients (υπολογιστές ή χρήστες που αιτούνται μιας υπηρεσίας) στην αντίθετη πλευρά της πύλης στο εξωτερικό δίκτυο, προωθώντας τα αιτήματα μεταξύ μιας κρυμμένης IP και μιας κρυμμένης θύρας που είναι ορατές μόνο στο εσωτερικό δίκτυο και μιας φανεράς διεύθυνσης IP και μιας φανεράς θύρας που είναι ορατές στο εξωτερικό δίκτυο.



Η τεχνική του port forwarding μπορεί να χρησιμοποιηθεί γενικότερα με διάφορους τρόπους προκειμένου να ασφαλίσουμε πόρτες και πρωτόκολλα. Στα επόμενα θα περιγράψουμε μια από αυτές τις χρήσεις.

Την τεχνική του port forwarding μπορούμε να την εφαρμόσουμε ως διαχειριστές προκειμένου να κρύψουμε (καμουφλάρουμε) μια πόρτα από τον εξωτερικό δίκτυο. Την τεχνική αυτή μας την προσφέρει το Firewall που περιγράψαμε στην προηγούμενη παράγραφο.

Για την εφαρμογή αυτής της τεχνικής πρέπει να υπάρχουν δύο κάρτες δικτύου (δικτυακές διεπαφές – network interfaces). Η μία από αυτές θα βλέπει το εξωτερικό δίκτυο έχοντας

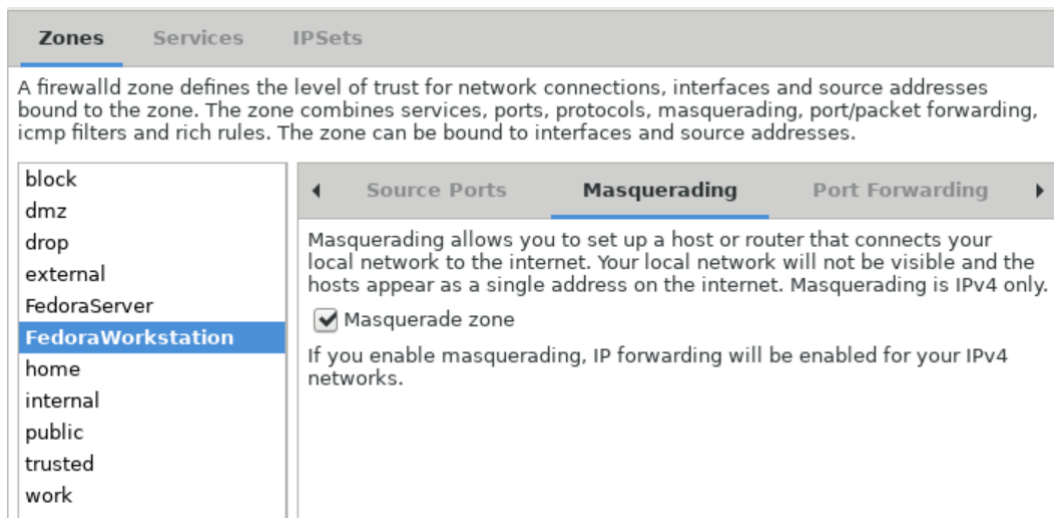
πραγματική IP (real IP) και η άλλη θα έχει τοπική διεύθυνση IP (local IP) και θα βλέπει το εσωτερικό δίκτυο.

Αυτό που μπορούμε να πετύχουμε με αυτή τη διαδικασία είναι να δρομολογούμε την κίνηση από μια πόρτα τους συστήματος που βλέπει το εξωτερικό δίκτυο σε μια πόρτα του συστήματος που βλέπει μόνο το εσωτερικό μας δίκτυο.

Για παράδειγμα αν θέλουμε να έχουμε έναν sshd (υπηρεσία ασφαλούς κελύφους - secure shell) στην πόρτα 22 και δε θέλουμε να είναι ορατή στο εξωτερικό δίκτυο για να αποφεύγουμε τις επιθέσεις, αλλά παράλληλα θέλουμε να έχουμε πρόσβαση απ' έξω στη συγκεκριμένη πόρτα, μπορούμε να τη διασυνδέσουμε εικονικά από μια τυχαία πόρτα για παράδειγμα την 6500 της πραγματικής δημόσιας IP (που «βλέπει» όλος κόσμος) στη γνωστή πόρτα 22 της τοπικής IP που «βλέπουν» μόνο όσοι είναι συνδεδεμένοι στο τοπικό ιδιωτικό μας δίκτυο και φυσικά και ο Linux Server μας. Με αυτό τον τρόπο αποφεύγουμε απ' ευθείας επιθέσεις στην πόρτα 22 (η οποία είναι μια πόρτα που δέχεται και τις περισσότερες επιθέσεις), αφού στο δημόσιο δίκτυο θα φαίνεται κλειστή.

Για να πραγματοποιηθεί αυτό το port forwarding, ανοίγουμε το firewall που είδαμε προηγουμένως.


Η διαδικασία ισχύει μόνο για IPv4 δίκτυα και πριν από όλα πρέπει να τσεκάρουμε το Masquerading Zone όπως φαίνεται στην ακόλουθη Εικόνα 7- 10

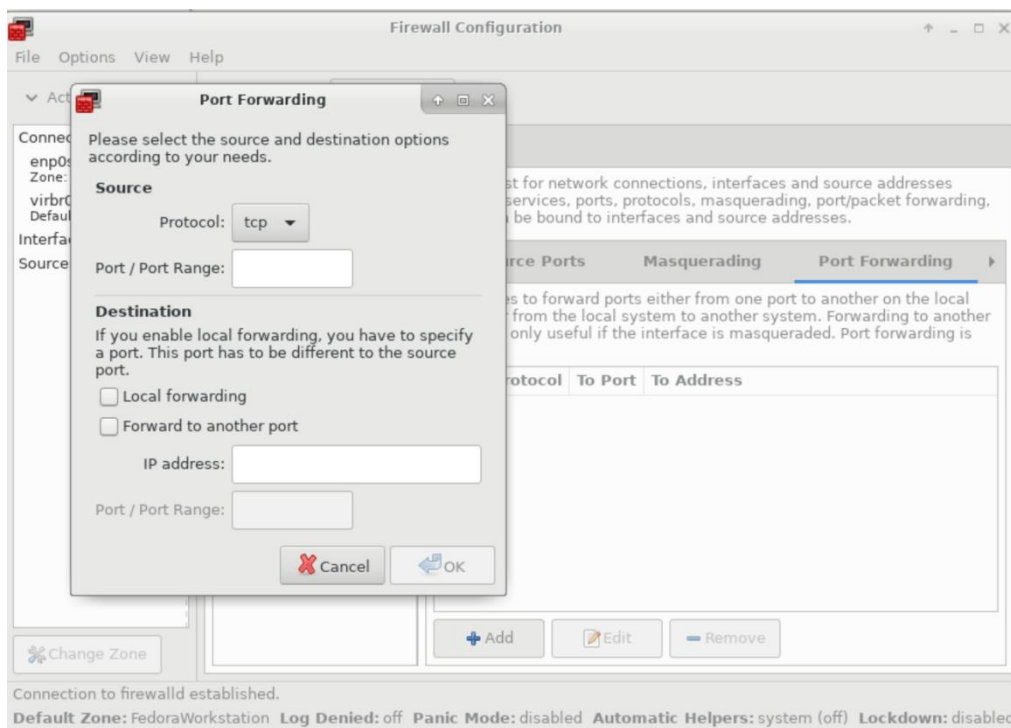


Εικόνα 7- 10: Διαδικασία Masquerading

με το masquerading δημιουργούμε ένα εικονικό κανάλι το οποίο διασυνδέει το τοπικό ιδιωτικό μας δίκτυο με το δημόσιο δίκτυο που διασυνδέει όλο τον κόσμο. Με άλλα λόγια δημιουργεί ένα κανάλι ανάμεσα στις δύο κάρτες δικτύου

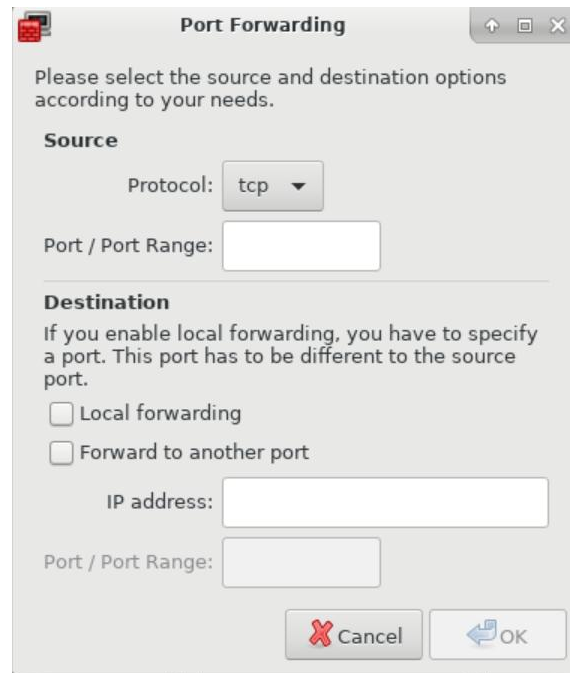
Στη συνέχεια επιλέγουμε το **Port Forwarding** στα δεξιά (Εικόνα 7- 11)

 Το **masquerading** (μεταμφίεση - καμουφλάρισμα) είναι μια τεχνική που κρύβει έναν ολόκληρο χώρο διευθύνσεων IP, που συνήθως αποτελείται από ιδιωτικές διευθύνσεις IP ή μια μεμονωμένη τοπική ιδιωτική διεύθυνση, πίσω από μια μεμονωμένη πραγματική δημόσια διεύθυνση IP. Οι κρυφές διευθύνσεις μετατρέπονται σε μία δημόσια διεύθυνση IP ως διεύθυνση προέλευσης των εξερχόμενων πακέτων IP, έτσι ώστε να εμφανίζονται ότι δεν προέρχονται από τον κρυφό κεντρικό υπολογιστή αλλά από την συσκευή που έχει τη δημόσια διεύθυνση. Λόγω της δημοτικότητας αυτής της τεχνικής ο όρος NAT (Network Address Translation) έχει γίνει σχεδόν συνώνυμος με τη μεταμφίεση IP.



Εικόνα 7- 11: Port Forwarding

Πατώντας το add ανοίγει το επόμενο παράθυρο (Εικόνα 7- 12)



Εικόνα 7- 12: Port Forwarding παράθυρο διαλόγου add

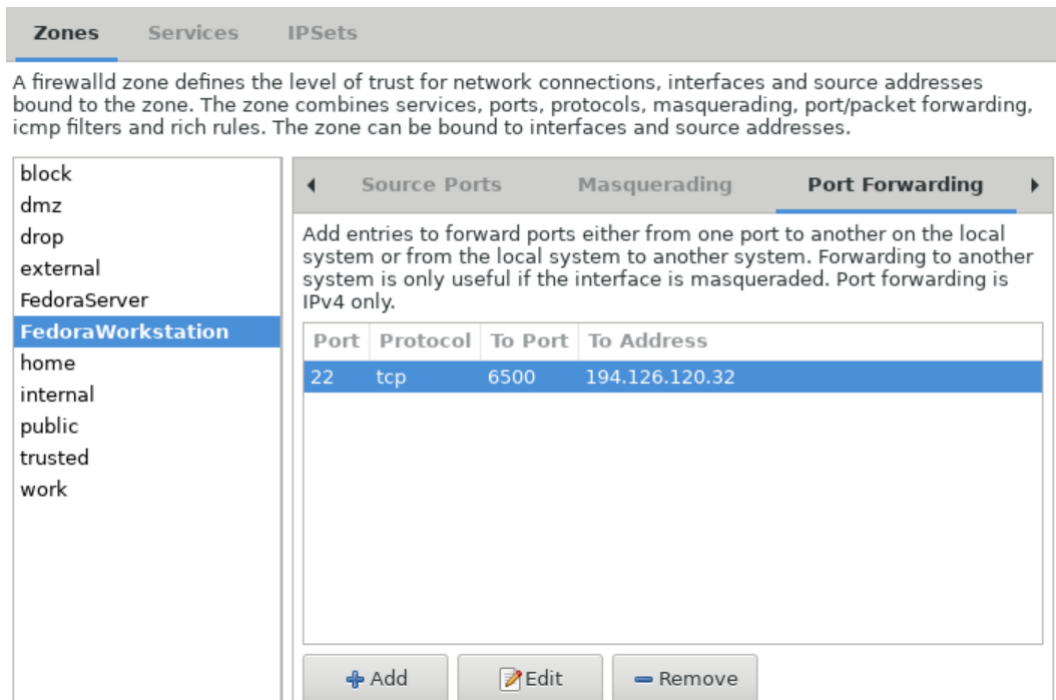
στο οποίο πρέπει να συμπληρώσουμε την τοπική διεύθυνση IP με την τοπική πόρτα (και το πρωτόκολλο επικοινωνίας) και την πραγματική πόρτα με την πόρτα που θα είναι ορατή στο εξωτερικό δίκτυο. Όταν συμπληρωθούν αυτά θα έχουμε μια εικόνα όπως η Εικόνα 7- 13



Εικόνα 7- 13: Port Forwarding παράθυρο διαλόγου add συμπληρωμένο

Η πραγματική δημόσια IP που φαίνεται στην Εικόνα 7- 14 είναι υποθετική. Σε αυτό το σημείο θα πρέπει να βάλουμε την IP που έχουμε αγοράσει από τον πάροχό μας και έχουμε αποδώσει στη δεύτερη κάρτα δικτύου.





Εικόνα 7- 14: Port Forwarding με την αντίστοιχη εγγραφή

Τέλος στην Εικόνα 7- 14 βλέπουμε την εγγραφή στο Port Forwarding

### Ασκήσεις - Δραστηριότητες

Σε περίπτωση που υπάρχει διπλή κάρτα δικτύου πραγματοποιήστε τις διαδικασίες που περιγράφονται στην παρούσα παράγραφο.

#### 7.1.4 Fail2Ban

Το Fail2Ban είναι εξαιρετικό εργαλείο το οποίο επιτρέπει στο διαχειριστή του Linux Server να αποκλείει αυτόματα επιθέσεις, την ώρα της επίθεσης. Ήρθε να λύσει ένα πρόβλημα που παλιότερα χρειαζόταν ο διαχειριστής να γράψει κώδικα (κάποιο bash script) και να συνδυάσει εργαλεία και αρχεία προκειμένου να επιτύχει το ίδιο αποτέλεσμα.

Ένα βασικό πρόβλημα που υπήρχε πάντα στην ασφάλεια των Server και γενικότερα των υπολογιστικών συστημάτων που φιλοξενούν υπηρεσίες και είναι ανοικτά στο εξωτερικό δίκτυο είναι ότι πολλές επιθέσεις γίνονται από την απέναντι πλευρά της γης σε ώρες που στην περιοχή του Server είναι νύχτα και στην πλευρά των επιτιθέμενων είναι μέρα. Παλαιότερα ο διαχειριστής διαπίστωνε ότι όλο το βράδυ δεχόταν επίθεση και στην καλύτερη περίπτωση οι υπηρεσίες του ή ολόκληρο το σύστημα ήταν «κάτω» (ανενεργό) και στη χειρότερη ότι τελικά οι εισβολείς είχαν εισέλθει και είχαν πλέον κάνει ανεπανόρθωτη ζημιά.

Το Fail2Ban έρχεται να λύσει αυτό ακριβώς το πρόβλημα. Μπορεί να αποκλείσει επιτιθέμενες IPs αυτόματα χωρίς την παρέμβαση του διαχειριστή ανάλογα με την παραμετροποίηση που θα έχει. Για παράδειγμα μπορεί να αποκλείσει μια συγκεκριμένη IP

αν κάνει 5 διαδοχικές προσπάθειες σύνδεσης (login) στην πόρτα 22, όπου βρίσκεται η υπηρεσία sshd (secure shell - ασφαλούς κελύφους)

Η εγκατάσταση του Fail2Ban γίνεται με την ακόλουθη εντολή

```
[root@ekdda ~]# dnf -y install fail2ban
```

Στη συνέχεια αντιγράφουμε δυο αρχεία από την πρωτότυπη μορφή τους σε μια τοπική στις οποίες θα κάνουμε όλες τις παραμετροποιήσεις. Αντιγράφουμε το **fail2ban.conf** στο **fail2ban.local** και το **jail.conf** στο **jail.local**

```
[root@ekdda ~]# cp /etc/fail2ban/fail2ban.conf
/etc/fail2ban/fail2ban.local
[root@ekdda ~]# cp /etc/fail2ban/jail.conf
/etc/fail2ban/jail.local
```

ανοίγουμε το αρχείο παραμετροποίησης **jail.local**

```
[root@ekdda ~]# gedit /etc/fail2ban/jail.local
```

Στη γραμμή 50 προσθέτουμε την ακόλουθη εντολή η οποία ζητά από το Fail2Ban να αγνοήσει τη διεύθυνση του localhost

```
ignoreip = 127.0.0.1
```

Στη γραμμή 59, 63, 66 ορίζουμε κάποιους γενικούς κανόνες για το χρόνο απαγόρευσης (bantime) καθώς και τις επιτρεπόμενες προσπάθειες (επαναλήψεις προσπάθειας σύνδεσης) από το απομακρυσμένο σύστημα (maxretry), αλλά στα παρακάτω θα δούμε ότι μπορούμε να ορίσουμε διαφορετικές τέτοιες παραμέτρους για το κάθε πρωτόκολλο το οποίο θα προστατεύει το fail2Ban. Αυτά που θα ορίσουμε στα διάφορα τμήματα (sections) για τα διάφορα πρωτόκολλα και υπηρεσίες έχουν μεγαλύτερη ισχύ από αυτό που ορίζεται εδώ στο γενικό τμήμα (section)

Στη γραμμή 59 προσθέτουμε (χρόνος αποκλεισμού σε s)

```
bantime = 86400
```

Στη γραμμή 63 προσθέτουμε

```
findtime = 600
```

Στη γραμμή 66 προσθέτουμε

```
maxretry = 5
```

Οι τρεις τελευταίες αλλαγές είναι πολύ σημαντικές και σημαίνουν : απαγόρευσε για 86400 sec κάθε IP η οποία κάνει brute forcing 6 φορές κάθε 10 λεπτά (600 sec).

Στη γραμμή 87 αλλάζουμε

```
backend = systemd
```

Στη γραμμή 113 αλλάζουμε την κατάσταση του Fail2Ban

```
enable = true
```

Στη γραμμή 130 - 133 αλλάζουμε τον τρόπο με το οποίο θα μας ειδοποιεί το Fail2Ban

```
enable = true
```

Στη γραμμή 139 προσδιορίζουμε τον SMTP του συστήματός μας (sendmail ή postfix) αν θέλουμε ως διαχειριστές να λαμβάνουμε email με την παράνομη δραστηριότητα που ανακαλύπτει το Fail2Ban

```
mta = sendmail
```

Από τη γραμμή 221 ξεκινάει το τμήμα στο οποίο προσδιορίζουμε ποιες υπηρεσίες προστατεύει το Fail2Ban. Για κάθε υπηρεσία μπορούν να επανακαθοριστούν οι ρυθμίσεις λειτουργίας ξεχωριστά.

Αποθηκεύουμε και βγαίνουμε

Ξεκινούμε την υπηρεσία

```
[root@ekdda ~]# systemctl enable fail2ban  
[root@ekdda ~]# systemctl start fail2ban
```

Με την ακόλουθη εντολή μπορούμε να δούμε τις καταγραφές (logs) του Fail2Ban

```
[root@ekdda ~]# more /var/log/fail2ban.log
```

Οι επόμενες τρεις εντολές είναι πολύ σημαντικές αφού έχει τελειώσει η ρύθμιση

Με την επόμενη εντολή βλέπουμε ποιες υπηρεσίες προστατεύονται

```
[root@ekdda ~]# Fail2ban-Client status
```

Αν προστατεύεται για παράδειγμα η υπηρεσία sshd, με την επόμενη εντολή βλέπουμε λεπτομέρειες δράσης του Fail2Ban σε αυτή την υπηρεσία

```
[root@ekdda ~]# Fail2ban-Client -d sshd
```

Και με την επόμενη εντολή βλέπουμε αν υπάρχουν προσπάθειες για login στον sshd, πόσες είναι αυτές και αν έχουν αποκλειστεί κάποιες IPs σύμφωνα με τους κανόνες

```
[root@ekdda ~]# Fail2ban-Client status sshd
```

### Ασκήσεις - Δραστηριότητες

Πραγματοποιήστε την εγκατάσταση της εφαρμογής σύμφωνα με τις οδηγίες και εκτελέστε τις εντολές της παραγράφου. Δημιουργήστε μια brute Force από το Kali (παράγραφος 7.1.5) και δείτε στα logs του Fail2Ban την αποκλεισμένη IP. Επίσης από το Cockpit δείτε πώς φαίνεται η δράση του Fail2Ban

### 7.1.5 Kali

Το Kali είναι μια διανομή Linux που χρησιμοποιείται από διαχειριστές σε όλο τον κόσμο για να εξετάζουν θέματα ασφάλειας των υπολογιστικών συστημάτων τους. Διαθέτει όλα εκείνα τα εργαλεία τα οποία μπορεί να χρησιμοποιήσει ένας διαχειριστής για να διαπιστώσει κατά πόσο ο Linux Server του είναι ασφαλής και αντέχει σε διαφόρων ειδών επιθέσεις.



Εικόνα 7- 15: Η διανομή Kali

Εργαλεία που συμπεριλαμβάνονται στο Kali για Brute Force Attack (Επίθεση Ωμής Βίας - δείτε στην εισαγωγή του κεφαλαίου)

- hydra (xhydra – GUI)
- medusa
- Ncrack
- Air-crack
- L0phtCrack – OphCrack

Χρησιμοποιώντας ένα από τα παραπάνω εργαλεία μπορούμε να δημιουργήσουμε ως διαχειριστές μια επίθεση ωμής βίας στο σύστημά μας ώστε να δούμε εάν τα εργαλεία που διαθέτουμε μπορούν να ανταποκριθούν και να μην επιτρέψουν στους εισβολείς να δημιουργήσουν πρόβλημα στο σύστημά μας.

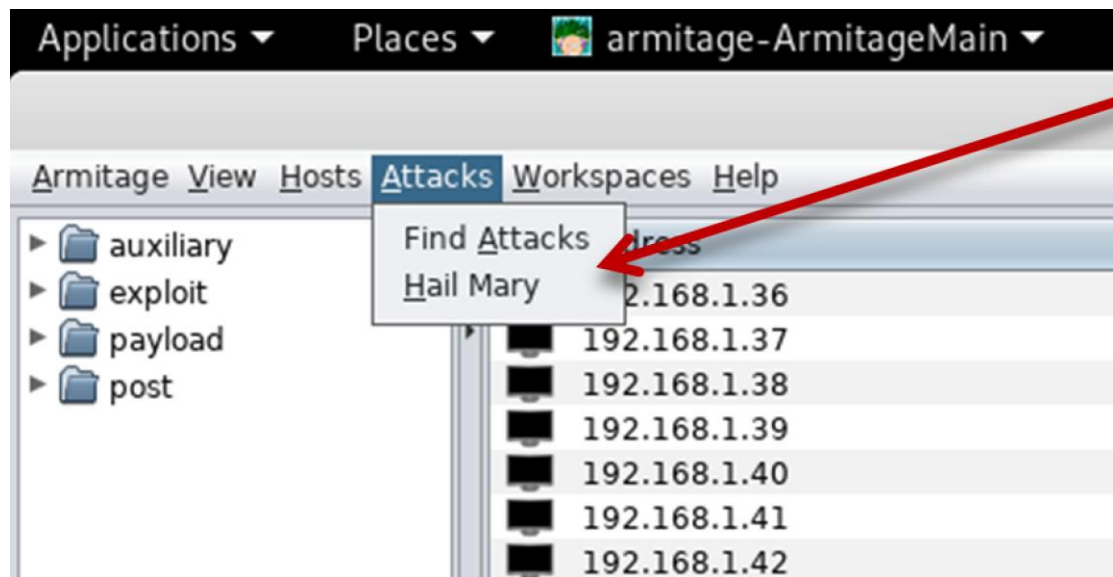
#### **7.1.5.1 System Vulnerabilities – Exploits attack**

Με τον όρο System Vulnerabilities εννοούμε τα τρωτά σημεία του συστήματός μας. Αυτά τα σημεία αφορούν πρωτόκολλα, υπηρεσίες, λογισμικά, τον πυρήνα, αλλά και το ίδιο το λειτουργικό τα οποία εξαιτίας αστοχιών των προγραμματιστών δημιουργούν συνθήκες τις οποίες οι εισβολείς μπορούν να εκμεταλλευτούν ώστε να μπουν στο σύστημά μας. Οι διαδικασίες με τις οποίες αποφύγουμε γνωστές τρωτότητες είναι οι ενημερώσεις όλων των προηγούμενων και φυσικά η παρακολούθηση της δραστηριότητας (monitoring). Μόλις βρεθεί η τρωτότητα, εφαρμόζεται μια Exploit Attack (Επίθεση Εκμετάλλευσης) με κάποιο γνωστό script που εφαρμόζεται με συγκεκριμένο τρόπο σε συγκεκριμένο πρωτόκολλο και σε συγκεκριμένη πόρτα.

#### **7.1.5.2 Armitage και Metasploit**

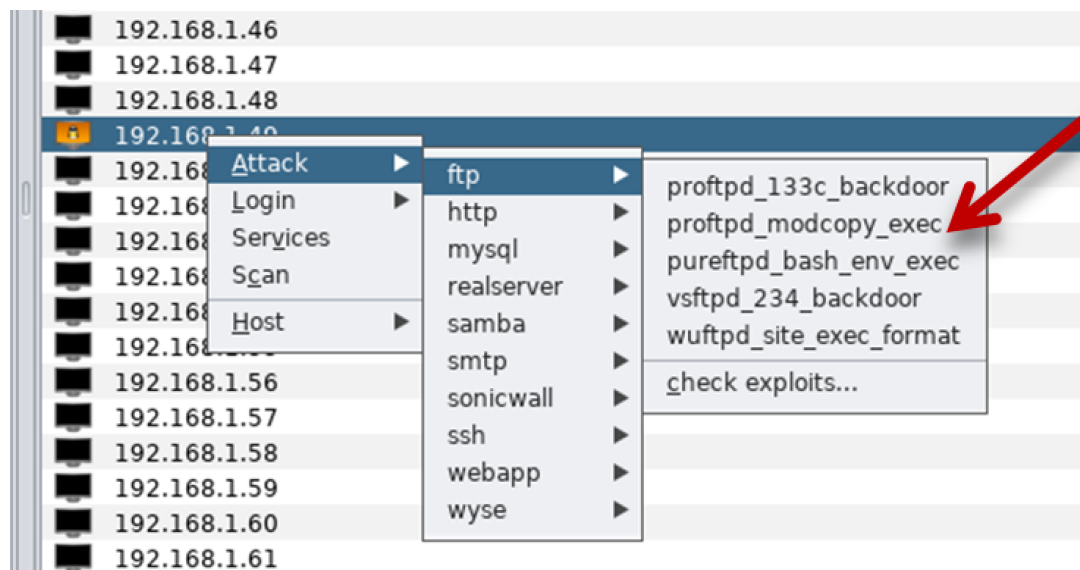
Το Kali διαθέτει εργαλεία τα οποία εκμεταλλεύονται τις τρωτότητες (vulnerabilities) του συστήματός μας. Τα εργαλεία αυτά είναι το σύνολο εργαλείων Metasploit Project (Framework edition) (WikiMetasploit, n.d.) και το γραφικό Armitage το οποίο κάνει πιο εύκολη τη λειτουργία του Metasploit.

Στις Εικόνα 7- 16 και Εικόνα 7- 17, βλέπουμε ότι έχει ήδη εκτελεστεί μια χαρτογράφηση του δικτύου (βλέπουμε τις IPs οι οποίες έχουν καταγραφεί. Με πορτοκαλί είναι οι ενεργές IPs). Από το μενού επιλέγοντας «Find Attacks» βλέπουμε ότι μπορούμε να αναζητήσουμε κατάλληλες επιθέσεις στην IP που έχουμε επιλέξει. Η επιλογή «Hail Mary» προκαλεί μια «τρομερή» επίθεση σε όλα τα πρωτόκολλα και με όλους τους τρόπους που διαθέτει το Metasploit.



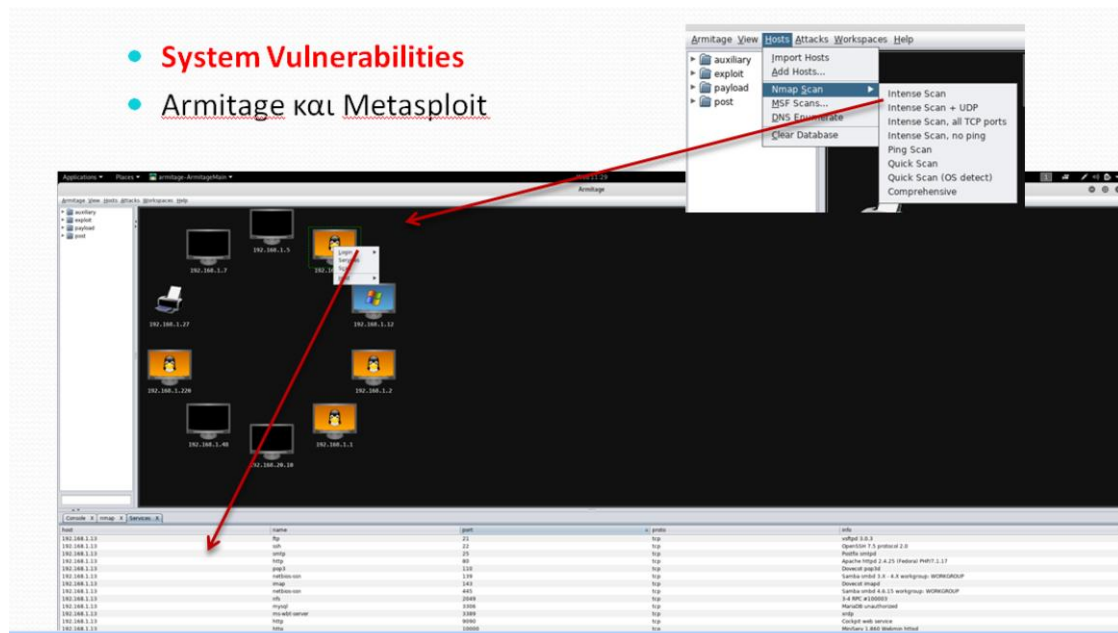
Εικόνα 7- 16: Εφαρμογή επίθεσης σε τοπικό δίκτυο

το γραφικό Armitage προτείνει συγκεκριμένες επιθέσεις σε συγκεκριμένα πρωτόκολλα για κάθε υπολογιστικό σύστημα που έχει χαρτογραφήσει.



Εικόνα 7- 17: Εφαρμογή επίθεσης σε πρωτόκολλο σε συγκεκριμένη IP

Στην επόμενη Εικόνα 7- 18 βλέπουμε μια πιο γραφική απεικόνιση του τοπικού δικτύου όπου παρουσιάζονται τα συστήματα με τα λειτουργικά τους, οι πιθανές επιθέσεις καθώς και οι μολύνσεις με exploits (εκμεταλλεύσεις) που μπορούν να εφαρμοσθούν.



Εικόνα 7- 18: Υπολογιστικά συστήματα πιθανές επιθέσεις σε πρωτόκολλα και πιθανά exploits

Αν όλα πάνε καλά για τον εισβολέα, τότε παίρνει τον έλεγχο του απομακρυσμένου συστήματος αποκτώντας συνήθως πρόσβαση σε ένα τερματικό με δικαιώματα διαχειριστή (root)

Είναι προφανές ότι ως διαχειριστές, με τη χρήση των σχετικών εργαλείων του Kali μπορούμε να διαπιστώσουμε πόσο τρωτός είναι ο Linux Server μας και να κάνουμε όλες τις σχετικές ενέργειες προκειμένου να αποκλείσουμε τα τρωτά σημεία.

### 7.1.5.3 DOS Attack

Όπως είδαμε στα προηγούμενα DOS attack είναι μια επίθεση με σκοπό την καταστολή της λειτουργίας του Server (Denial of Service Attack – Επίθεση Άρνησης Υπηρεσίας). Αυτού του είδους οι επιθέσεις πραγματοποιούνται όταν ο επιτιθέμενος δημιουργεί ένα μεγάλο όγκο από αιτήματα εξυπηρέτησης από τον Server. Αυτό έχει σαν αποτέλεσμα ο Server να μην μπορεί να ανταπεξέλθει και στο τέλος είτε απλά να μην μπορεί να εξυπηρετήσει άλλα «νόμιμα αιτήματα» είτε να κολλήσει εντελώς και να χρειάζεται επανεκκίνηση.

Και για αυτού του είδους τις επιθέσεις υπάρχουν από το Kali κατάλληλα εργαλεία, όπως τα auxiliary synflood και ping of death.

```

msf > use auxiliary/dos/tcp/synflood
msf auxiliary(dos/tcp/synflood) > show options

Module options (auxiliary/dos/tcp/synflood):

  Name      Current Setting  Required  Description
  ----      -
  INTERFACE          no         The name of the interface
  NUM                no         Number of SYNs to send (else unlimited)
  RHOST             yes         The target address
  RPORT            80         The target port
  SHOST             no         The spoofable source address (else randomizes)
  SNAPLEN          65535      yes         The number of bytes to capture
  SPORT            no         The source port (else randomizes)
  TIMEOUT          500        yes         The number of seconds to wait for new data

msf auxiliary(dos/tcp/synflood) > set RHOST 192.168.1.49
RHOST => 192.168.1.49
msf auxiliary(dos/tcp/synflood) > exploit
[*] SYN flooding 192.168.1.49:80..

```

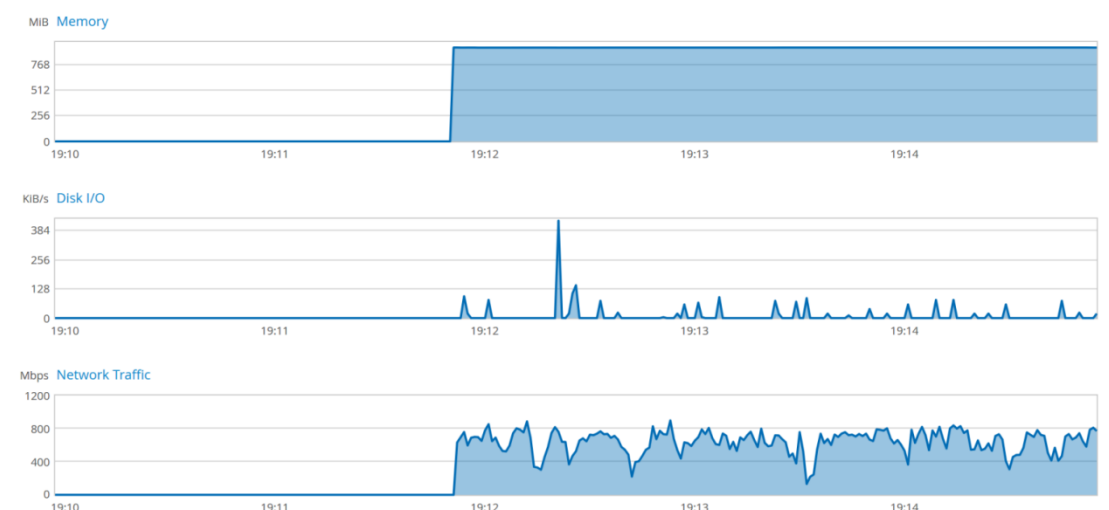
Εικόνα 7- 19: Εργαλεία DOS attack από το Kali

Το Syncc είναι ένας τύπος DoS attack ο οποίος στέλνει ένα μεγάλο αριθμό από σήματα Syncc με σκοπό να καταναλωθούν (απασχοληθούν εξαντλητικά) όλοι οι πόροι του συστήματος στόχος. Με αυτόν τον τρόπο το σύστημα ουσιαστικά τίθεται «εκτός λειτουργίας».

Άλλο είδος DoS attack είναι το Ping of Death

Και αυτά τα εργαλεία χρησιμοποιούνται και από τους διαχειριστές των συστημάτων για να διαπιστώνουν την κατάσταση και την ετοιμότητα του συστήματός τους (κάτι παρόμοιο γίνεται με τα stress tests των τραπεζών με οικονομικούς όρους φυσικά και με παρόμοια εργαλεία)

Στην παρακάτω Εικόνα 7- 20 μπορούμε να δούμε πώς φαίνεται από το cockpirt μια τέτοια DoS attack



Εικόνα 7- 20: Παρακολούθηση μιας επίθεσης DOS από το Cockpirt



## Ασκήσεις - Δραστηριότητες

Χρησιμοποιήστε μια εικονική μηχανή με Kali και χρησιμοποιώντας τα εργαλεία σύμφωνα με τις οδηγίες της παραγράφου, εκτελέστε επιθέσεις στους τοπικούς χρήστες-διαχειριστές με σκοπό μέσα από εργαλεία παρακολούθησης που έχουν εγκαταστήσει, να δουν πώς φαίνονται οι επιθέσεις στα συστήματά τους.

### 7.1.6 Ενημερώσεις (Updates)

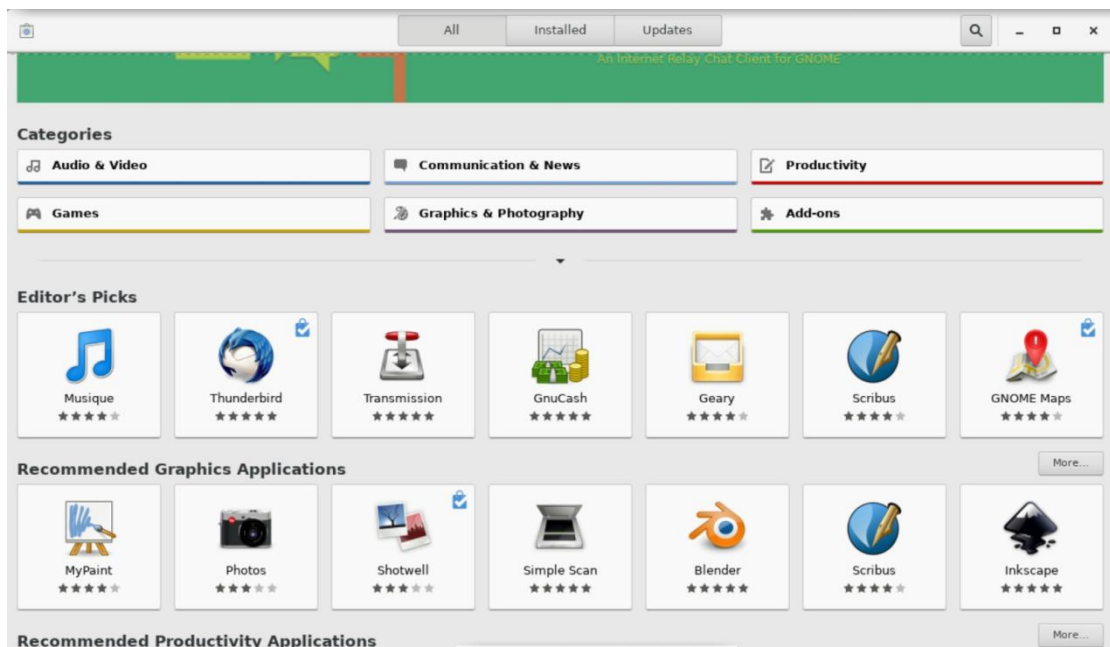
Από τα προηγούμενα είναι σαφές ότι ένα ενημερωμένο (updated) σύστημα με τις τελευταίες εκδόσεις τόσο του λειτουργικού όσο και των άλλων λογισμικών και υπηρεσιών είναι ασφαλέστερο.

Σε καθημερινή βάση ως διαχειριστές πρέπει να συμβουλευόμαστε τα κατάλληλα εργαλεία με τα οποία πρέπει να είμαστε ενήμεροι για τις διαθέσιμες ενημερώσεις τις οποίες πρέπει να εφαρμόσουμε ώστε το σύστημά μας να είναι ασφαλέστερο

Ένα τέτοιο εργαλείο είναι το gnome software, το οποίο εκτελείται με την εντολή

```
[root@ekdda ~]# gnome-software
```

Στην επόμενη Εικόνα 7- 21 μπορούμε να δούμε την οθόνη της εφαρμογής. Η εφαρμογή αυτή διαθέτει μια πολύ δυνατή και λειτουργική διεπαφή χρήστη, με την οποία ο διαχειριστής μπορεί πολύ εύκολα να δει το εγκατεστημένο λογισμικό, το διαθέσιμο, καθώς και τις ενημερώσεις. Επιπλέον μπορεί να κάνει γραφικά κάποια εγκατάσταση.



Εικόνα 7- 21: Η Εφαρμογή gnome-software

Πολλά περισσότερα για τις ενημερώσεις περιγράφονται στην παράγραφο .....

## Ασκήσεις - Δραστηριότητες

Ανοίξτε την εφαρμογή και περιεργαστείτε την

### 7.2 Εργαλεία παρακολούθησης δικτυακής δραστηριότητας

Στις επόμενες παραγράφους παρουσιάζουμε ένα σύνολο εργαλείων με τα οποία μπορούμε να έχουμε μια πολύ καλή παρακολούθηση τόσο της δικτυακής δραστηριότητας του Linux Server όσο και την παρακολούθηση διαφόρων υπολογιστικών πόρων, ανάλογα με το εργαλείο.

Πρέπει να τονίσουμε ότι υπάρχει μια ομάδα εργαλείων παρακολούθησης τα οποία χρησιμοποιούν τα πρωτόκολλα UDP και SNMP, όπως τα MRTGs, το CACTI, το NAGIOS, το ZABBIX και το MUNIN.

Από αυτά τα εργαλεία περιγράφουμε πολύ αναλυτικά την εγκατάσταση και αρκετά κομμάτια της παραμετροποίησης του CACTI, γιατί το θεωρήσαμε πολύ σημαντικό και πολύ αντιπροσωπευτικό. Η λειτουργία του θεωρούμε ότι είναι πολύ ικανοποιητική, η εγκατάσταση σχετικά εύκολη και εκπαιδευτικά το θεωρούμε το καταλληλότερο

Τα προαναφερθέντα λογισμικά είναι σημαντικά και συμπεριλαμβάνονται και στις διανομές **CENTOS** αλλά και στην εμπορική **RedHat Enterprise Edition**. Επιπλέον κάποια από τα παραπάνω λογισμικά έχουν και εμπορικές εκδόσεις

#### 7.2.1 netstat

Η **netstat** είναι εντολή του Linux η οποία παρέχει βασικά στατιστικά στοιχεία για όλες τις δικτυακές δραστηριότητες και ενημερώνει τους χρήστες σχετικά με τις θύρες και τις διευθύνσεις που εκτελούν τις αντίστοιχες συνδέσεις (TCP, UDP). Επίσης παρέχει πληροφορίες για τις πόρτες που είναι ανοικτές και τις διεργασίες που βρίσκονται σε αυτές και είναι διαθέσιμες να εξυπηρετήσουν. Η λέξη netstat προέρχεται από τις λέξεις net και stat (δηλαδή network and Statistics, δίκτυο και στατιστικά στοιχεία). Ιστορικά η netstat εμφανίστηκε πρώτη φορά στην Unix διανομή BSD (Berkley Software Distribution). Η BSD έκδοση 4.2 υποστήριξε την πρώτη οικογένεια πρωτοκόλλων διαδικτύου, TCP/IP. Η netstat έχει ενσωματωθεί στο Linux από την πρώτη έκδοσή του το 1991 και είναι επίσης εντολή και του DOS στα Windows από την εμφάνιση της έκδοσης 3.11 (1993), η οποία μπορούσε επίσης να επικοινωνήσει μέσω TCP/IP με τη βοήθεια επεκτάσεων.

Οι παράμετροι των εντολών της εντολής netstat (καθώς και των εξόδων τους) διαφέρουν από σύστημα σε σύστημα. Όμως η λειτουργία της, στις διάφορες υλοποιήσεις με τις οποίες εμφανίζεται είναι παρόμοια. Η netstat είναι ένα πρόγραμμα γραμμής εντολών και για αυτόν τον λόγο δεν διαθέτει γραφικό περιβάλλον εργασίας χρήστη. Κατά καιρούς έχουν εμφανιστεί κάποια γραφικά περιβάλλοντα για τη συγκεκριμένη εντολή όπως το TCPView (Microsoft), τα οποία καθιστούν δυνατή την εμφάνιση γραφικών στατιστικών.

Εκτελώντας

```
[root@ekdda ~]# netstat --help
```

Μπορούμε να δούμε το σύνολο των παραμέτρων της εντολής καθώς και κάποια παραδείγματα

Ενδεικτικά να αναφέρουμε κάποιες χρήσεις της εντολής μαζί με την έξοδο που έχουν για να δούμε τη χρησιμότητά της

Εκτελώντας

```
[root@ekdda ~]# netstat -apn
```

Έχουμε την ακόλουθη έξοδο (Εικόνα 7- 22)

```
[root@ekdda ~]# netstat -apn | more
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:45301          0.0.0.0:*                LISTEN      -
tcp        0      0 192.168.124.1:53      0.0.0.0:*                LISTEN      1344/dnsmasq
tcp        0      0 0.0.0.0:21           0.0.0.0:*                LISTEN      843/vsftpd
tcp        0      0 127.0.0.1:3350       0.0.0.0:*                LISTEN      1398/xrdp-sesman
tcp        0      0 0.0.0.0:22           0.0.0.0:*                LISTEN      845/sshd
tcp        0      0 127.0.0.1:631        0.0.0.0:*                LISTEN      1420/cupsd
tcp        0      0 0.0.0.0:25           0.0.0.0:*                LISTEN      1173/master
tcp        0      0 0.0.0.0:3389         0.0.0.0:*                LISTEN      1399/xrdp
tcp        0      0 0.0.0.0:445          0.0.0.0:*                LISTEN      1354/smbd
tcp        0      0 0.0.0.0:2049         0.0.0.0:*                LISTEN      -
tcp        0      0 127.0.0.1:199        0.0.0.0:*                LISTEN      836/snmpd
tcp        0      0 0.0.0.0:139          0.0.0.0:*                LISTEN      1354/smbd
tcp        0      0 0.0.0.0:110          0.0.0.0:*                LISTEN      1466/dovecot
tcp        0      0 0.0.0.0:143          0.0.0.0:*                LISTEN      1466/dovecot
tcp        0      0 0.0.0.0:50895        0.0.0.0:*                LISTEN      1387/rpc.statd
tcp        0      0 0.0.0.0:111          0.0.0.0:*                LISTEN      1/systemd
tcp        0      0 0.0.0.0:10000        0.0.0.0:*                LISTEN      1543/perl
tcp        0      0 0.0.0.0:20048        0.0.0.0:*                LISTEN      1447/rpc.mountd
tcp6       0      0 :::22                :::*                    LISTEN      845/sshd
```

Εικόνα 7- 22: Έξοδος της εντολής netstat με παράμετρο -apn

Στην οποία φαίνεται η το πρωτόκολλο, οι ενεργές (active) πόρτες (παράμετρος -a) και σε ποια IP είναι ενεργή (η IP 0.0.0.0 σημαίνει όλες οι κάρτες δικτύου του Linux Server. Επίσης φαίνεται το PID καθώς και το όνομα της υπηρεσίας (παράμετρος -p)

### Ασκήσεις - Δραστηριότητες

Εκτελέστε την εντολή netstat με τις διαφορετικές παραμέτρους (μπορείτε να δείτε μια λίστα των παραμέτρων εκτελώντας την εντολή με την παράμετρο help όπως περιγράφεται πιο πάνω) και δείτε τα αποτελέσματα στο τερματικό

#### 7.2.2 nmap

Το nmap είναι ένα δωρεάν, ανοιχτού κώδικα λογισμικό το οποίο χρειάζεται εγκατάσταση και το οποίο χρησιμεύει στην απεικόνιση των πρωτοκόλλων του συστήματός μας που είναι διαθέσιμα να προσφέρουν δικτυακή υπηρεσία.

Η εγκατάστασή του γίνεται με την εντολή

```
[root@ekdda ~]# dnf -y install nmap
```

Το nmap γενικά χρησιμεύει στη χαρτογράφηση του δικτύου και στην ασφάλεια. Η λέξη nmap προέρχεται από το (Network Mapper δηλαδή χαρτογράφος δικτύου). Πολλά συστήματα και διαχειριστές δικτύου το βρίσκουν επίσης χρήσιμο για εργασίες όπως απογραφή δικτύου, διαχείριση προγραμμάτων αναβάθμισης υπηρεσίας και παρακολούθηση του χρόνου λειτουργίας του κεντρικού υπολογιστή ή της υπηρεσίας.

Το Nmap χρησιμοποιεί πρωτογενή πακέτα IP (ICMP packets) για να προσδιορίσει ποιους κεντρικοί υπολογιστές είναι διαθέσιμοι στο δίκτυο, ποιες υπηρεσίες (όνομα εφαρμογής και έκδοση) προσφέρουν αυτοί οι κεντρικοί υπολογιστές, ποια λειτουργικά συστήματα (και εκδόσεις λειτουργικού συστήματος) εκτελούν, τι είδους φίλτρα πακέτων / τείχη προστασίας χρησιμοποιούνται, και δεκάδες άλλα χαρακτηριστικά. Έχει σχεδιαστεί για γρήγορη σάρωση μεγάλων δικτύων, αλλά λειτουργεί καλά εναντίον μεμονωμένων κεντρικών υπολογιστών.



Το nmap είναι πολύ χρήσιμο πρόγραμμα για τον διαχειριστή του Linux Server. Επειδή όμως μπορεί να χαρτογραφήσει τις υπηρεσίες και τις πόρτες που είναι ανοικτές στα διάφορα συστήματα, γενικότερα «κόβεται» από τους δρομολογητές όταν προσπαθεί να έχει κάποιος πρόσβαση από το δημόσιο δίκτυο και προσπαθεί να χαρτογραφήσει τις δικτυακές υπηρεσίες του Server μας. Έτσι η χρήση του γενικά περιορίζεται πιο πολύ στη χαρτογράφηση του Server που διαχειριζόμαστε και γενικότερα στη χαρτογράφηση του εσωτερικού μας δικτύου.

Το Nmap εκτελείται σε όλα τα μεγάλα λειτουργικά συστήματα υπολογιστών και διατίθενται επίσημα πακέτα για Linux, Windows και Mac OS X. Εκτός από την κλασική εκτελέσιμη γραμμή εντολών nmap, η σουίτα Nmap περιλαμβάνει και έκδοση με γραφικό περιβάλλον (Zenmap - περιγράφεται στην επόμενη παράγραφο), ένα ευέλικτο εργαλείο μεταφοράς δεδομένων, ανακατεύθυνσης και εντοπισμού σφαλμάτων (Ncat), ένα βοηθητικό πρόγραμμα για τη σύγκριση αποτελεσμάτων σάρωσης (Ndiff) και ένα εργαλείο δημιουργίας πακέτων και ανάλυσης απόκρισης (Nping).

Εκτελώντας

```
[root@ekdda ~]# nmap --help
```

Μπορούμε να δούμε το σύνολο των παραμέτρων της εντολής καθώς και κάποια παραδείγματα

Εκτελώντας

```
[root@ekdda ~]# nmap localhost
```

Έχουμε την ακόλουθη έξοδο (Εικόνα 7- 23)

```
[root@ekdda ~]# nmap localhost

Starting Nmap 7.40 ( https://nmap.org ) at 2020-12-26 10:55 EET
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0000090s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 984 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
25/tcp    open  smtp
80/tcp    open  http
110/tcp   open  pop3
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
143/tcp   open  imap
199/tcp   open  smux
445/tcp   open  microsoft-ds
631/tcp   open  ipp
2049/tcp  open  nfs
3306/tcp  open  mysql
3389/tcp  open  ms-wbt-server
9090/tcp  open  zeus-admin
10000/tcp open  snet-sensor-mgmt

Nmap done: 1 IP address (1 host up) scanned in 0.19 seconds
```

Εικόνα 7- 23: Έξοδος της εντολής nmap

### Ασκήσεις - Δραστηριότητες

Εκτελέστε την εντολή nmap με τις διαφορετικές παραμέτρους (μπορείτε να δείτε μια λίστα των παραμέτρων εκτελώντας την εντολή με την παράμετρο help όπως περιγράφεται πιο πάνω) και δείτε τα αποτελέσματα στο τερματικό



**localhost** είναι το τοπικό υπολογιστικό μας σύστημα. Επίσης είναι το default hostname του Linux Server, ώσπου να το αλλάξουμε ως διαχειριστές

### 7.2.3 zenmap

Το zenmap είναι το γραφικό nmap που περιγράψαμε στην προηγούμενη παράγραφο.

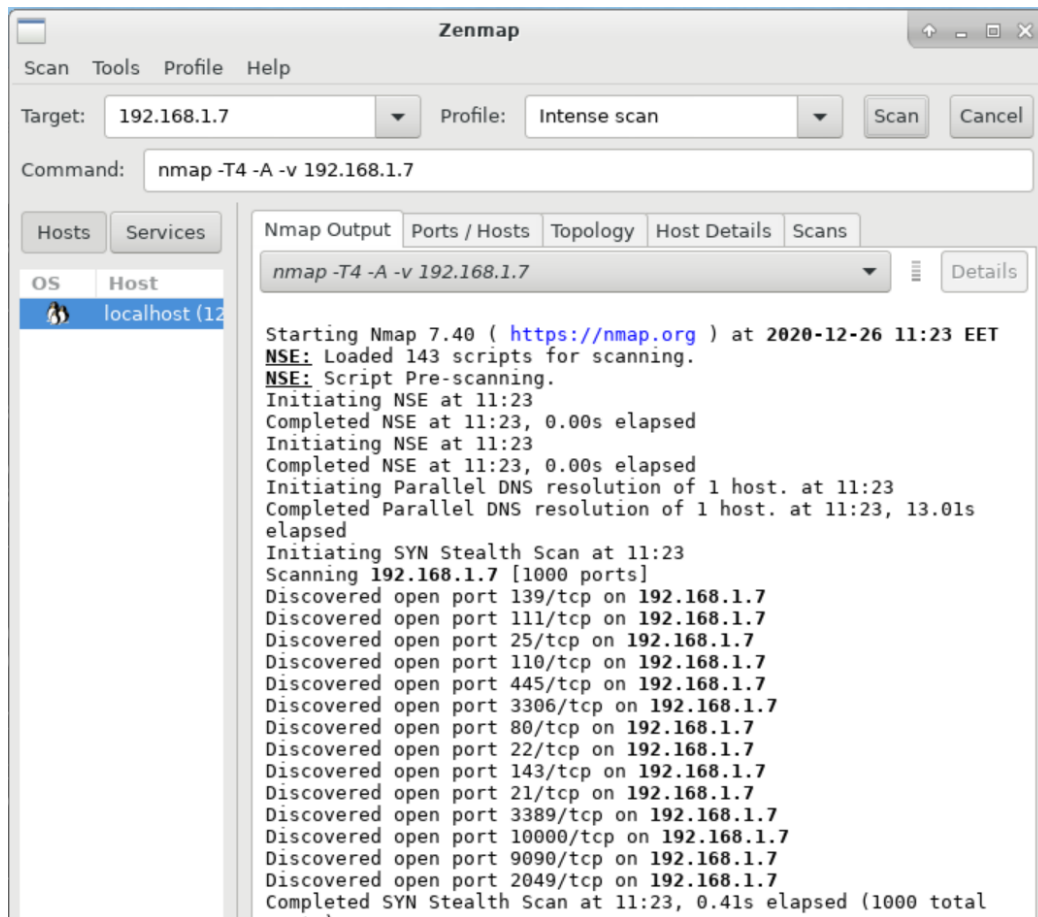
Η εγκατάστασή του γίνεται με την εντολή

```
[root@ekdda ~]# dnf -y install nmap-frontend
```

Και η εκτέλεσή του με την εντολή

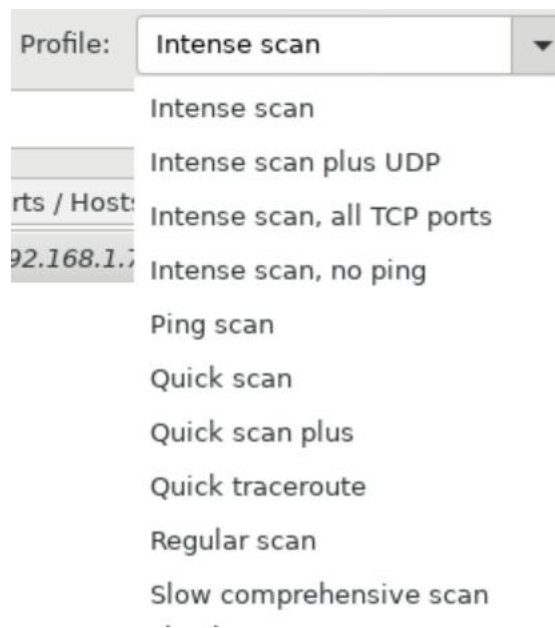
```
[root@ekdda ~]# zenmap
```

Το γραφικό του περιβάλλον είναι αυτό που φαίνεται στην Εικόνα 7- 24



Εικόνα 7- 24: Δικτυακή απεικόνιση του Server με το Zenmap

Στην προηγούμενη Εικόνα 7- 24 φαίνεται η χαρτογράφηση του Linux Server με IP 192.168.1.7. Το zenmap παρέχει διάφορα έτοιμα Profiles δικτυακής χαρτογράφησης του στόχου



Εικόνα 7- 25: Profiles εκτέλεσης του Zenmap

Η απεικόνιση που παρέχει το zenmap είναι αναλυτική (πύρτες και υπηρεσίες - Εικόνα 7- 26)

| Nmap Output |       | Ports / Hosts | Topology | Host Details  | Scans                   |
|-------------|-------|---------------|----------|---------------|-------------------------|
|             | Port  | Protocol      | State    | Service       | Version                 |
| ●           | 21    | tcp           | open     | ftp           | vsftpd 3.0.3            |
| ●           | 22    | tcp           | open     | ssh           | OpenSSH 7.5 (protocol : |
| ●           | 25    | tcp           | open     | smtp          | Postfix smtpd           |
| ●           | 80    | tcp           | open     | http          | Apache httpd 2.4.25 ((F |
| ●           | 110   | tcp           | open     | pop3          | Dovecot pop3d           |
| ●           | 111   | tcp           | open     | rpcbind       | 2-4 (RPC #100000)       |
| ●           | 139   | tcp           | open     | netbios-ssn   | Samba smbd 3.X - 4.X (  |
| ●           | 143   | tcp           | open     | imap          | Dovecot imapd           |
| ●           | 199   | tcp           | open     | smux          | Linux SNMP multiplexe   |
| ●           | 445   | tcp           | open     | netbios-ssn   | Samba smbd 4.6.15 (w    |
| ●           | 631   | tcp           | open     | ipp           | CUPS 2.2                |
| ●           | 2049  | tcp           | open     | nfs_acl       | 3 (RPC #100227)         |
| ●           | 3306  | tcp           | open     | mysql         | MySQL 5.5.5-10.1.33-M   |
| ●           | 3389  | tcp           | open     | ms-wbt-server | Microsoft Terminal Serv |
| ●           | 9090  | tcp           | open     | http          | Cockpit web service     |
| ●           | 10000 | tcp           | open     | http          | MiniServ 1.953 (Webmi   |

Εικόνα 7- 26: Πύρτες και υπηρεσίες με χαρτογράφηση από το Zenmap

μπορεί να απεικονίσει την τοπολογία του δικτύου (Εικόνα 7- 27)



Εικόνα 7- 27: Τοπολογία δικτύου με χαρτογράφηση από το Zenmap

καθώς και να δώσει πολλές άλλες πληροφορίες.

### Ασκήσεις - Δραστηριότητες

Εκτελέστε την εφαρμογή Zenmap με τα διαφορετικά profiles (από το πτυσσόμενο μενού) και χαρτογραφήστε τον Server σας. Δείτε τα αποτελέσματα της χαρτογράφησης στις διάφορες καρτέλες της εφαρμογής



### 7.2.4 Εργαλείο παρακολούθησης fedora

Η Fedora διαθέτει ένα πολύ καλό εργαλείο αποτύπωσης τόσο της δικτυακής δραστηριότητας, όσο και της παρακολούθησης πόρων του συστήματος. Το βρίσκουμε στη διαδρομή (για τον xfce X Windows Manager)

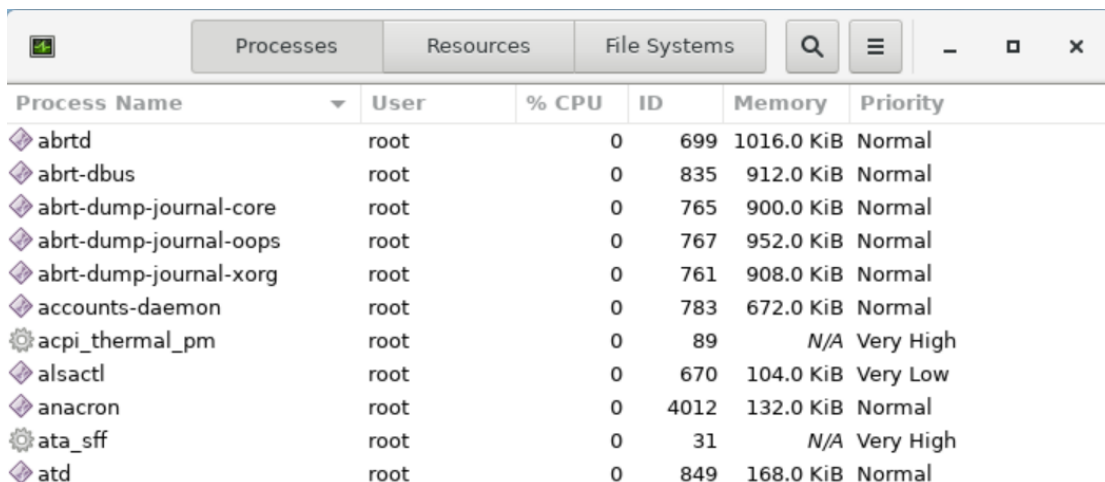
**Applications -> System -> System Monitor**

ή εκτελώντας στη γραμμή εντολών

```
[root@ekdda ~]# gnome-system-monitor
```

Στις επόμενες εικόνες φαίνεται η πληροφορία που μπορούμε να πάρουμε από την εφαρμογή.

Μπορούμε να έχουμε απεικόνιση των διεργασιών του συστήματος

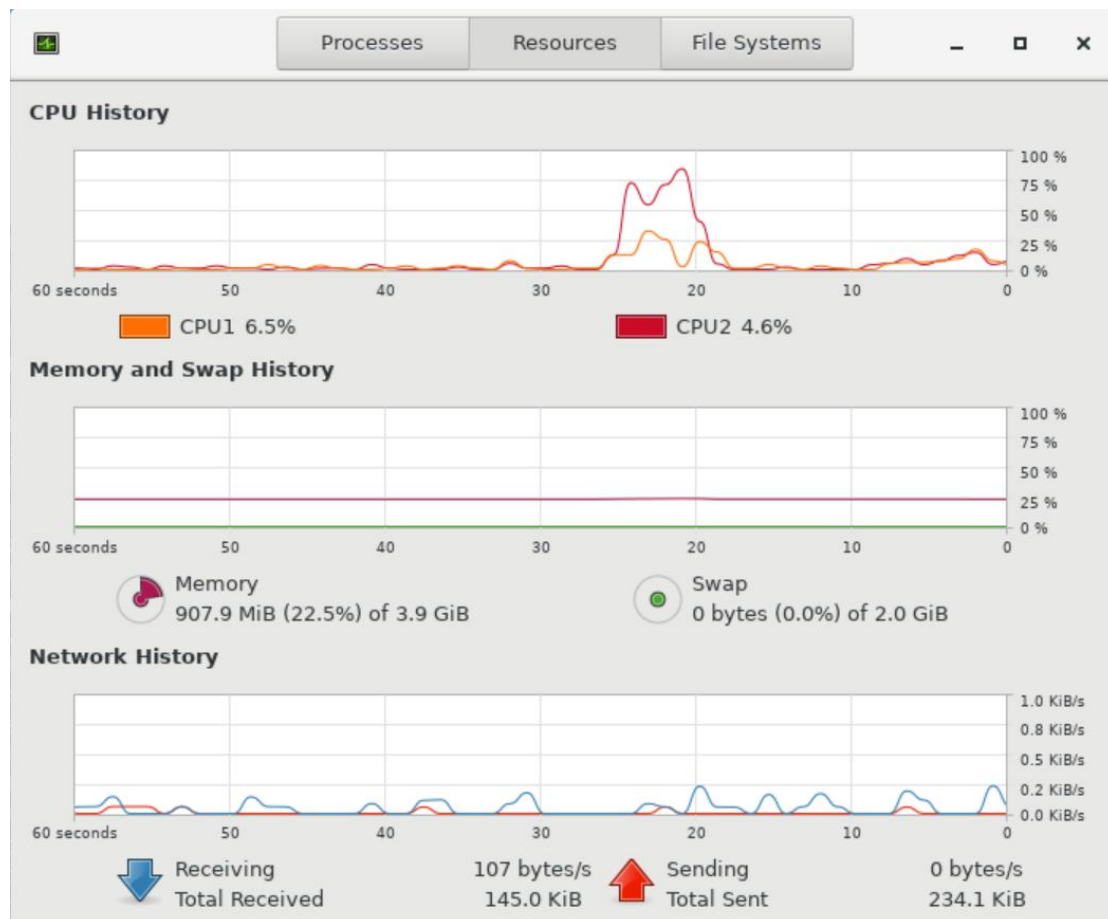


| Process Name           | User | % CPU | ID   | Memory     | Priority  |
|------------------------|------|-------|------|------------|-----------|
| abrtcd                 | root | 0     | 699  | 1016.0 KiB | Normal    |
| abrt-dbus              | root | 0     | 835  | 912.0 KiB  | Normal    |
| abrt-dump-journal-core | root | 0     | 765  | 900.0 KiB  | Normal    |
| abrt-dump-journal-oops | root | 0     | 767  | 952.0 KiB  | Normal    |
| abrt-dump-journal-xorg | root | 0     | 761  | 908.0 KiB  | Normal    |
| accounts-daemon        | root | 0     | 783  | 672.0 KiB  | Normal    |
| acpi_thermal_pm        | root | 0     | 89   | N/A        | Very High |
| alsactl                | root | 0     | 670  | 104.0 KiB  | Very Low  |
| anacron                | root | 0     | 4012 | 132.0 KiB  | Normal    |
| ata_sff                | root | 0     | 31   | N/A        | Very High |
| atd                    | root | 0     | 849  | 168.0 KiB  | Normal    |

Εικόνα 7- 28: Εικόνα των διεργασιών από το gnome-system-monitor

παρακολούθηση της χρήσης επεξεργαστή και μνήμης, καθώς και παρακολούθηση της κίνησης στις δικτυακές διεπαφές (κάρτες δικτύου)





Εικόνα 7- 29: Εικόνα διαφόρων πόρων συστήματος από το gnome-system-monitor

Επίσης μπορούμε να έχουμε απεικόνιση του συστήματος αρχείων

The screenshot shows the 'File Systems' tab of the gnome-system-monitor application, displaying a table of disk usage for various devices.

| Device    | Directory | Type | Total   | Available | Used     | Usage |
|-----------|-----------|------|---------|-----------|----------|-------|
| /dev/mapp | /         | ext4 | 17.8 GB | 4.1 GB    | 12.8 GB  | 75%   |
| /dev/sda1 | /boot     | ext4 | 1.0 GB  | 719.1 MB  | 233.8 MB | 24%   |

Εικόνα 7- 30: Εικόνα του συστήματος αρχείων από το gnome-system-monitor

### Ασκήσεις - Δραστηριότητες

Ανοίξτε την εφαρμογή που περιγράφεται στην παράγραφο και δείτε την πληροφορία που περιγράφεται στις παραπάνω εικόνες

### 7.2.5 Αρχεία Καταγραφής (log files)

Τα αρχεία καταγραφής είναι τα αρχεία στα οποία αποθηκεύει το Linux καταγραφές με τη δραστηριότητα που αφορά είτε το ίδιο το λειτουργικό με ό,τι συμβαίνει κατά τη διάρκεια της λειτουργίας του, είτε με ό,τι συμβαίνει κατά τη διάρκεια της λειτουργίας υπηρεσιών και εφαρμογών. Τα αρχεία αυτά είναι καταγραφές δραστηριότητας και λαθών που προκύπτουν από τη λειτουργία του πυρήνα, των εφαρμογών συστήματος, των εφαρμογών λογισμικού, των υπηρεσιών κτλ.

Τα αρχεία καταγραφής είναι ένα πολύ σημαντικό εργαλείο για τους διαχειριστές, για την παρακολούθηση σημαντικών συμβάντων σχετικά με τον Linux Server. Πρόκειται στην πραγματικότητα για μια ακτινογραφία της λειτουργίας του συστήματος. Πολλές φορές οι διαχειριστές δημιουργούν προσωπικά bash scripts προκειμένου να εξάγουν την πληροφορία που χρειάζονται. Επίσης για την καλύτερη λειτουργία τους εισάγουν εγγραφές και στον χρονοπρογραμματιστή του Linux, τον Cron.

Τα αρχεία καταγραφής βρίσκονται στη διαδρομή

**`/var/log/`**

και μπορούμε να τα διαβάσουμε με τις εντολές `more`, `cat` ή `less`. Για παράδειγμα

```
[root@ekdda ~]# more /var/log/audit/audit.log

type=USER_AUTH msg=audit(1542222078.974:45642): pid=7733 uid=0
auid=4294967295 s
es=4294967295 msg='op=password acct="bstefan" exe="/usr/sbin/ssh" hostname=? ad
dr=192.168.1.51 terminal=ssh res=failed'
type=USER_AUTH msg=audit(1542222078.991:45643): pid=7736 uid=0
auid=4294967295 s
es=4294967295 msg='op=password acct="bstefan" exe="/usr/sbin/ssh" hostname=? ad
dr=192.168.1.51 terminal=ssh res=failed'
type=USER_AUTH msg=audit(1542222079.018:45644): pid=7735 uid=0
auid=4294967295 s
es=4294967295 msg='op=password acct="bstefan" exe="/usr/sbin/ssh" hostname=? ad
dr=192.168.1.51 terminal=ssh res=failed'
```

Τυπικά μπορούμε να κατηγοριοποιήσουμε τα αρχεία καταγραφής στις παρακάτω 4 κατηγορίες :

- Εφαρμογών (Application Logs)
- Συμβάντων (Event Logs)
- Υπηρεσιών (Service Logs)
- Συστήματος (System Logs)

Μερικά σημαντικά τέτοια αρχεία καταγραφής περιγράφονται πιο κάτω.

Στα κλασικά συστήματα Linux υπάρχει το αρχείο καταγραφής

### **`/var/log/messages`**

το οποίο είναι πολύ σημαντικό γιατί περιέχει την καταγραφή όλου του συστήματος κατά την εκκίνηση του συστήματος, αλλά και ότι συμβαίνει κατά τη διάρκεια της λειτουργίας. Περιέχει πληροφορίες για σφάλματα του συστήματος, των εφαρμογών αλλά και με ότι έχει σχέση με υλικό. Για παράδειγμα ένα πρόβλημα στην κάρτα δικτύου

Στα σημερινά συστήματα Fedora (από τη χρήση της `system` και μετά) η καταγραφή στο αρχείο αυτό έχει αντικατασταθεί με την εντολή **`journalctl`**. Για παράδειγμα τα συμβάντα της εκκίνησης (boot) του συστήματος μπορούμε να τα δούμε με την εντολή

```
[root@ekdda ~]# journalctl -b
```

Επίσης αν κάποια υπηρεσία αποτύχει να ξεκινήσει τότε μπορούμε με την εντολή

```
[root@ekdda ~]# journalctl -xe
```

Να ανακαλύψουμε ποιο είναι το πρόβλημα και να το διορθώσουμε. Ας δούμε ένα παράδειγμα. Στο παρακάτω παράδειγμα αποτυγχάνει να εκκινηθεί η υπηρεσία του Apache Web Server, επειδή προσθέσαμε μια λάθος εντολή στο αρχείο παραμετροποίησης.

```
[root@ekdda ~]# systemctl restart httpd
Job for httpd.service failed because the control process exited
with error code.
See "systemctl status httpd.service" and "journalctl -xe" for
details.
[root@mail ~]# journalctl -xe
Dec 31 15:24:18 mail.ekdda02.linux systemd[1]: Starting The
Apache HTTP Server..
-- Subject: Unit httpd.service has begun start-up
-- Defined-By: systemd
-- Unit httpd.service has begun starting up.
Dec 31 15:24:18 mail.ekdda02.linux httpd[20137]: AH00526: Syntax
error on line 1
Dec 31 15:24:18 mail.ekdda02.linux httpd[20137]: Invalid command
'sdfsdfsdfsdf',
Dec 31 15:24:18 mail.ekdda02.linux systemd[1]: httpd.service:
Main process exite
Dec 31 15:24:18 mail.ekdda02.linux audit[1]: SERVICE_START pid=1
uid=0 auid=4294
Dec 31 15:24:18 mail.ekdda02.linux systemd[1]: Failed to start
The Apache HTTP S
-- Subject: Unit httpd.service has failed
```

Από τη μελέτη του αρχείου καταγραφής το οποίο εμφανίζεται με την εντολή `journalctl` και παράμετρο το `xe` που εκτελέσαμε ενδιάμεσα, ανακαλύπτουμε ότι υπάρχει λανθασμένη εντολή στη γραμμή 1 στο αρχείο `httpd.conf`

Η λανθασμένη εντολή είναι «`sdfsdfsdfsdf`»



Πρέπει να καταγράψουμε εδώ, ότι η παραπάνω μέθοδος δίνει πάντα πληροφορίες για τα προβλήματα που προέκυψαν, αλλά δεν είναι πάντα σαφείς, με αποτέλεσμα να μην είναι πάντα εύκολο να ανακαλύψουμε ποιο είναι το πρόβλημα.

Άλλο σημαντικό αρχείο καταγραφής είναι το

### **`/var/log/audit/audit.log`**

Στο οποίο μπορούμε ως διαχειριστές να βρούμε τις προσπάθειες σύνδεσης στο σύστημά μας. Επιτυχημένες και μη επιτυχημένες. Επίσης επιθέσεις τύπου brute force (ωμής βίας - δείτε και την εισαγωγή του 7<sup>ου</sup> κεφαλαίου) εντοπίζονται στο ίδιο αρχείο.

Πολύ σημαντικά επίσης είναι και τα αρχεία καταγραφής του Apache Web Server (αναλυτικά στην παράγραφο 8.2) τα οποία χρησιμοποιούνται και για θέματα λειτουργίας, αλλά και για θέματα ασφάλειας του Server τα οποία βρίσκονται στον κατάλογο

### **`/var/log/httpd/`**

```
[root@ekdda ~]# ls -l /var/log/httpd/
total 640
-rw-r--r-- 1 root root 2194 Dec 31 14:54 access_log
-rw-r--r-- 1 root root 31151 Nov 5 2018 access_log-20181105
-rw-r--r-- 1 root root 2760 Dec 31 15:39 error_log
-rw-r--r-- 1 root root 3780 Nov 5 2018 error_log-20181105
```

Στο παρακάτω φαίνεται το περιεχόμενο ενός από τα `access_logs`

```
[root@ekdda ~]# more /var/log/httpd/access_log
192.168.1.49 - - [29/Dec/2020:14:08:05 +0200] "GET
/example/index.php
9 "-" "Mozilla/5.0 (X11; Fedora; Linux i686; rv:54.0)
Gecko/20100101 Firefox/54.
0"
```

Από μια τέτοια καταγραφή μπορούμε να δούμε ότι στις 29 Δεκεμβρίου στις 14:08:05 η IP 192.168.1.49 ζήτησε και είδε την ιστοσελίδα `index.php` που βρίσκεται κάτω από τον κατάλογο `example` του Server μας. Ο browser του χρήστη ήταν Mozilla έκδοση 54 και το λειτουργικό σύστημα που χρησιμοποιούσε ήταν Linux Fedora.

Τέλος να πούμε ότι τα αρχεία καταγραφής είναι ορατά και από το περιβάλλον του Cockpit (παράγραφος 5.4.1) αλλά και από το περιβάλλον του Webmin (παράγραφος 5.4.2)

### Ασκήσεις - Δραστηριότητες

- Εκτελέστε τις εντολές της παραγράφου και δείτε τα αποτελέσματα που φαίνονται στις εικόνες
- Ανοίξτε το `httpd.conf` κάντε μια λανθασμένη καταχώρηση στη γραμμή 1, επανεκκινήστε την υπηρεσία `httpd`, και δείτε όσα αναφέρονται στην παράγραφο
- Κάντε επιτυχημένες και αποτυχημένες προσπάθειες εισόδου από `ssh` και δείτε τις καταγραφές στο αρχείο καταγραφής `/var/log/audit/audit.log`

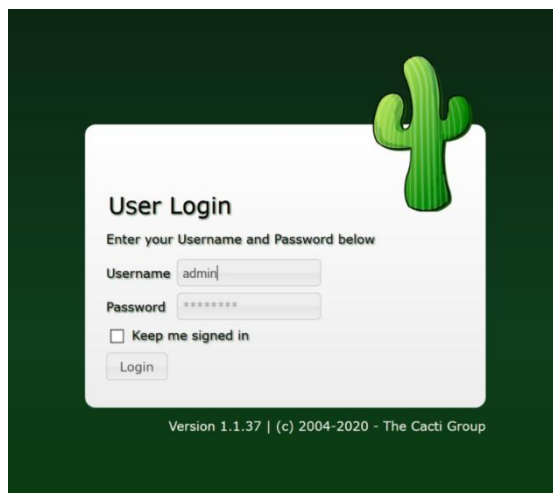
### 7.2.6 Cacti

Όπως είπαμε θεωρούμε το CACTI, πολύ σημαντικό εκπρόσωπο των εργαλείων παρακολούθησης που χρησιμοποιούν τα πρωτόκολλα UDP και SNMP. Η λειτουργία του πολύ ικανοποιητική, η εγκατάσταση σχετικά εύκολη και εκπαιδευτικά αξίζει να προσπαθήσει κάποιος να το εγκαταστήσει και να το λειτουργήσει.



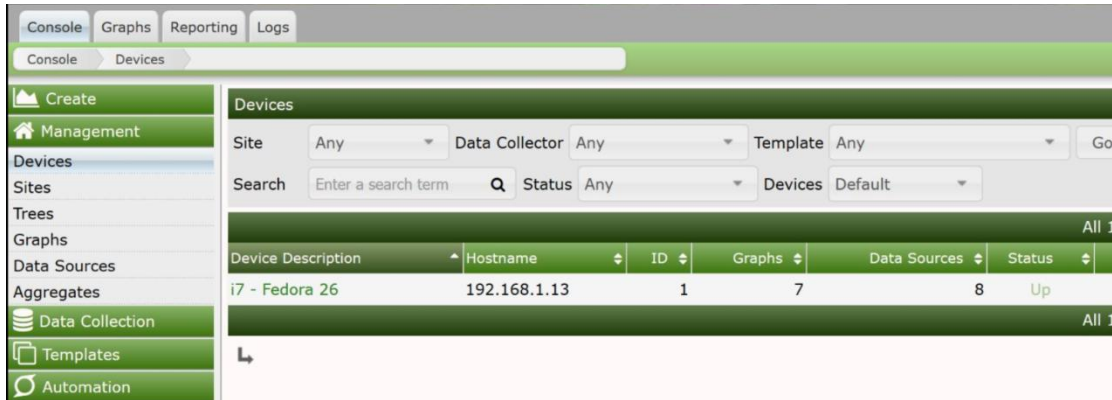
Μεταξύ της εκκίνησης του SNMP και της απεικόνισης των γραφημάτων που περιγράφουμε στη συγκεκριμένη ενότητα χρειάζεται να περάσει αρκετή ώρα και για το λόγο αυτό χρειάζεται υπομονή εκ μέρους του χρήστη που θέλει να δει την απεικόνιση

Η εισαγωγική οθόνη του Cacti είναι η ακόλουθη



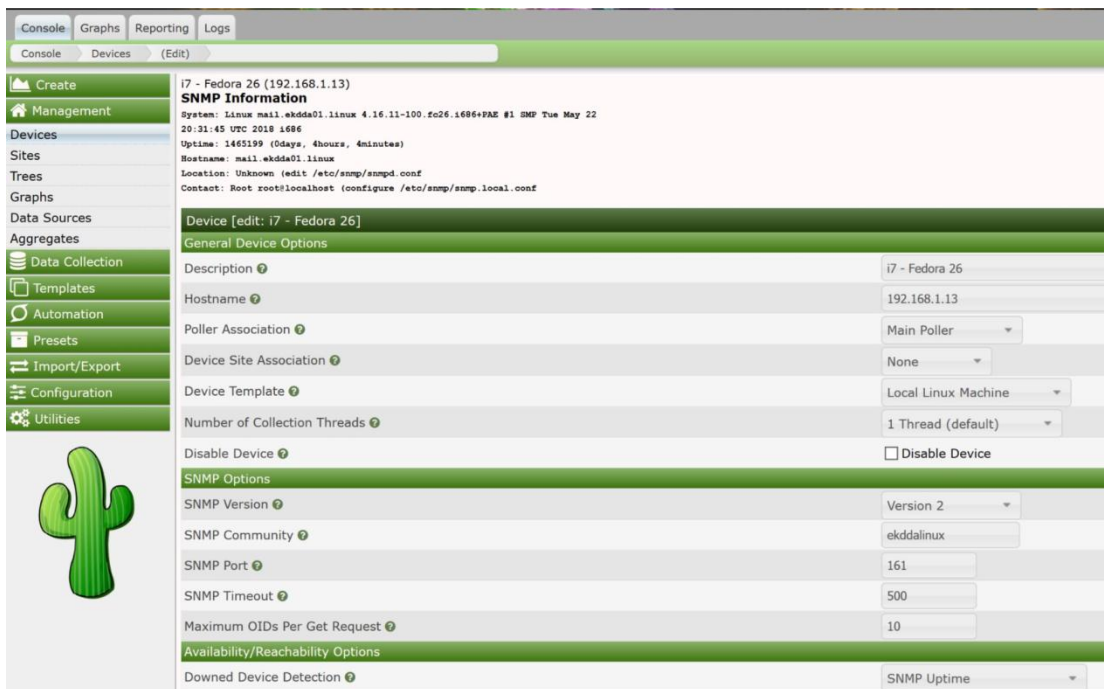
Εικόνα 7- 31: Οθόνη σύνδεσης Cacti

Αρχικά θα πρέπει ο διαχειριστής να ορίσει μια συσκευή (Device) όπως φαίνεται στην παρακάτω Εικόνα 7- 32. Ουσιαστικά ο διαχειριστής θα πρέπει να προσδιορίσει τα χαρακτηριστικά του Linux Server τον οποίο θέλει να παρακολουθεί.



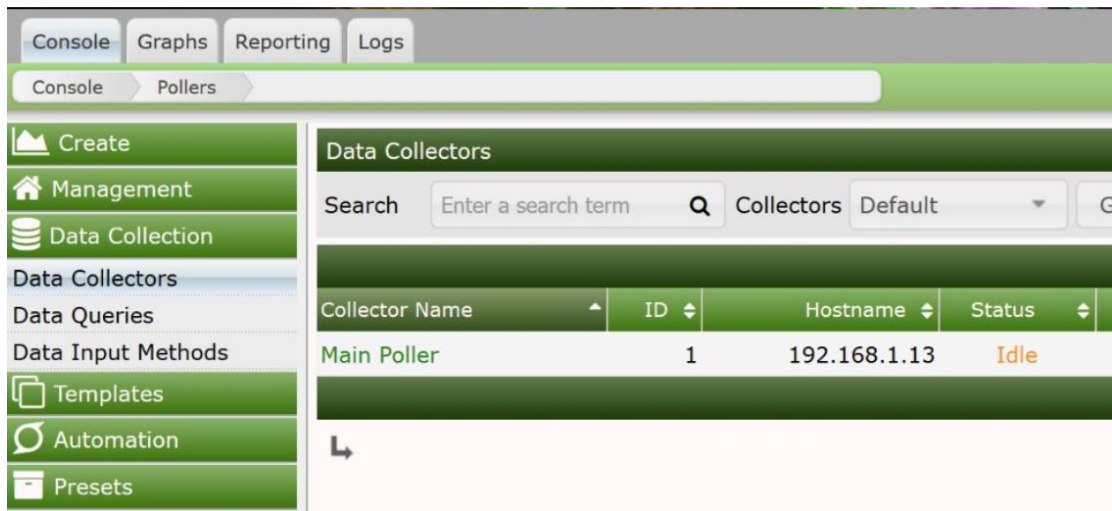
Εικόνα 7- 32: Απεικόνιση του device που θέλει να παρακολουθεί ο διαχειριστής στο Cacti

Κατά τον ορισμό της συσκευής (device) θα πρέπει ο διαχειριστής να προσδιορίσει το λειτουργικό σύστημα, την IP του Server καθώς και το community string που χρησιμοποιεί το SNMP (Εικόνα 7- 33).



Εικόνα 7- 33: Παραμετροποίηση του device που θέλει να παρακολουθεί ο διαχειριστής στο Cacti

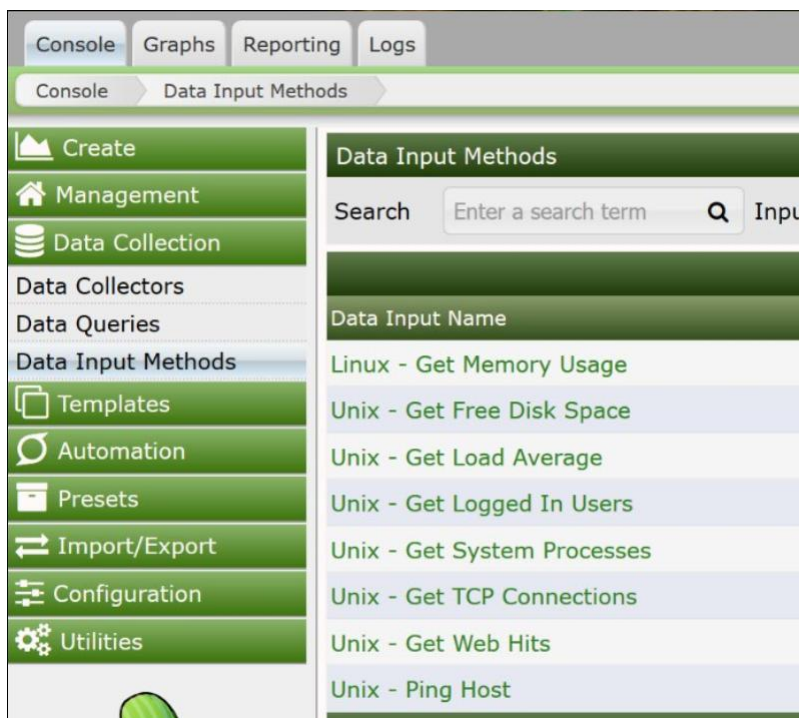
Μόλις τελειώσει ο ορισμός της συσκευής (device) μετά από κάποιο χρόνο αναμονής για την επικοινωνία της εφαρμογής με το SNMP θα δούμε τη συσκευή να γίνεται **up** όπως φαίνεται στο πεδίο Status, στην Εικόνα 7- 32



Εικόνα 7- 34: Προσδιορισμός του Data Collector στο Cacti

Στην οθόνη της Εικόνα 7- 34 προσδιορίζουμε τη συσκευή που θα είναι ο Data Collector (ο συλλέκτης των δεδομένων). Στην περίπτωση μας είναι ο ίδιος Linux Server μας.

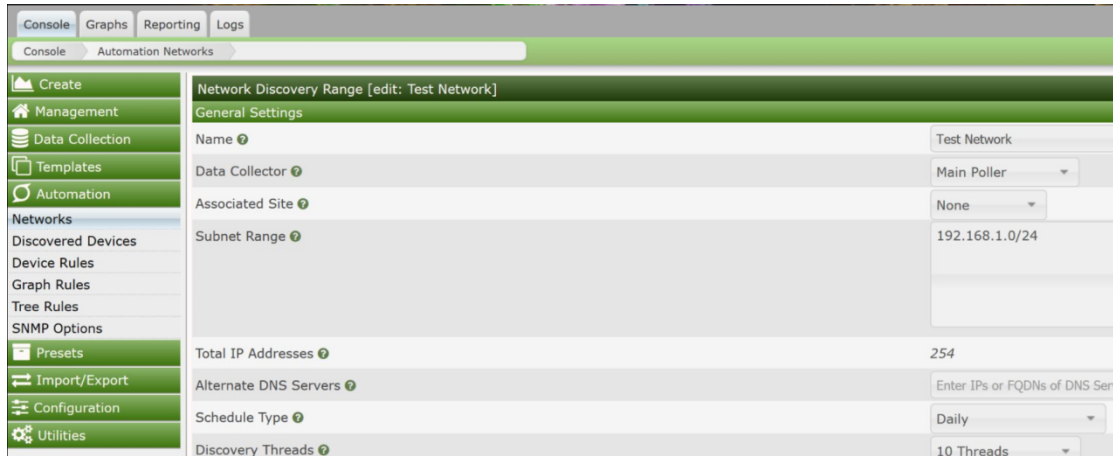
Στη συνέχεια πρέπει να ορίσουμε ποιες πληροφορίες θα συλλέγει, στην επιλογή Data Input Methods, Εικόνα 7- 35



Εικόνα 7- 35: Προσδιορισμός Μεθόδων Εισαγωγής Δεδομένων στο Cacti

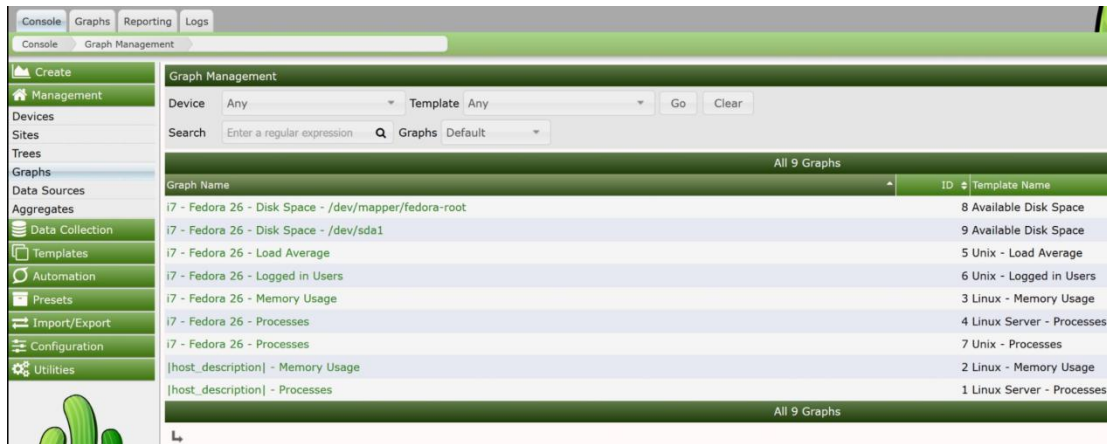
Στη συνέχεια πρέπει να καθοριστεί το υποδίκτυο στο οποίο δουλεύει ο Server μας Εικόνα 7- 36





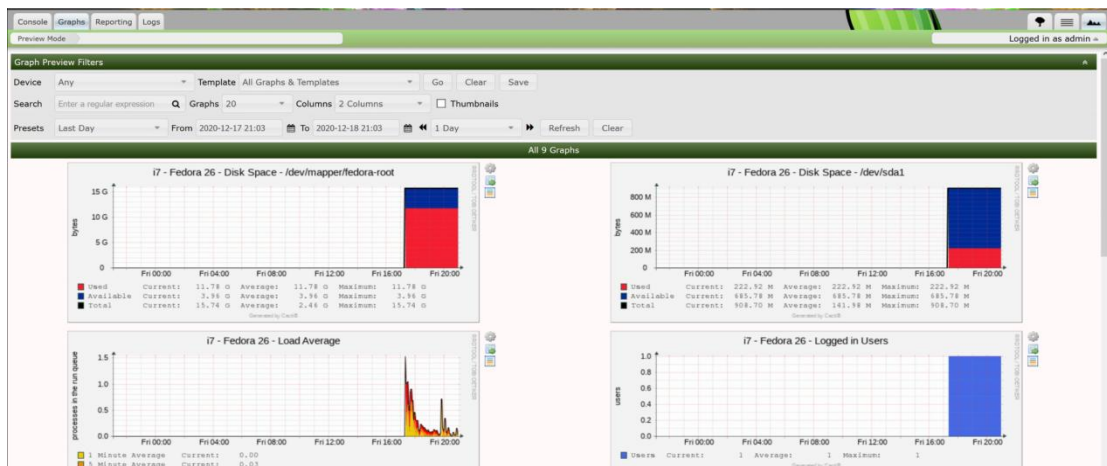
Εικόνα 7- 36: Στοιχεία για το networking στο Cacti

Και τέλος πρέπει να καθορίσουμε ποια γραφήματα θα είναι διαθέσιμα, Εικόνα 7- 37



Εικόνα 7- 37: Προσδιορισμός Διαθέσιμων Γραφημάτων στο Cacti

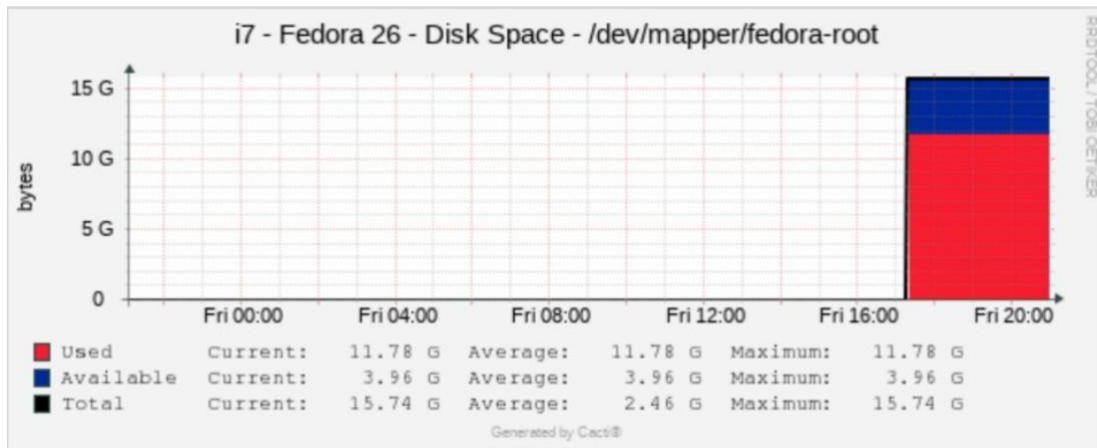
Όταν τελειώσουμε με τις παραμετροποιήσεις και αφού αφήσουμε να περάσει αρκετή ώρα προκειμένου να προλάβει το SNMP να στείλει πληροφορίες, στην επιλογή **Create** του μενού, μπορούμε ως διαχειριστές να δημιουργήσουμε τα διαγράμματα παρακολούθησης όπως φαίνεται στην επόμενη Εικόνα 7- 38



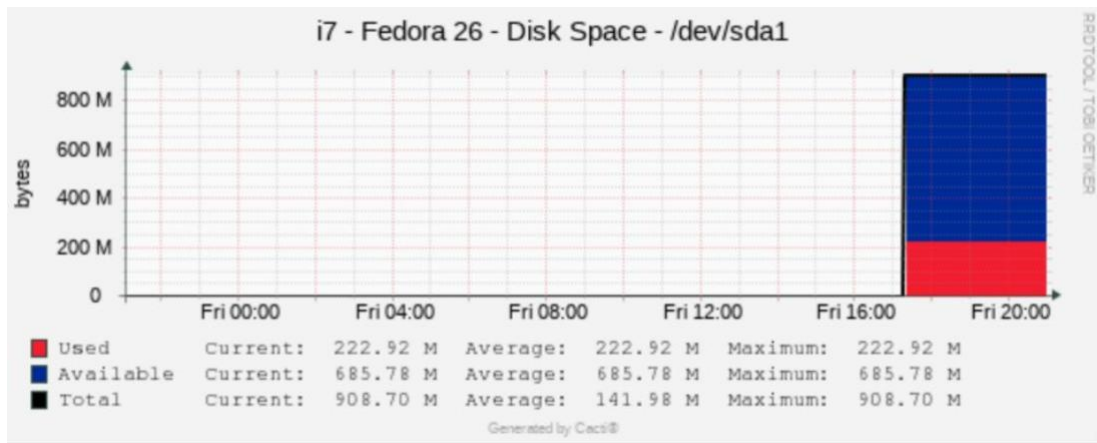
Εικόνα 7- 38: Επιλογή δημιουργίας διαγραμμάτων στο Cacti



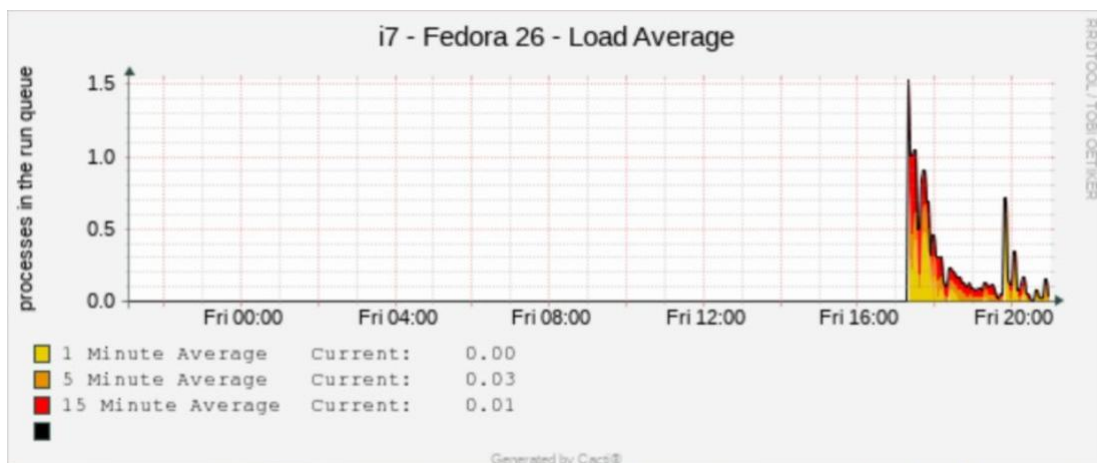
Στις επόμενες εικόνες (Εικόνα 7- 39, Εικόνα 7- 40, Εικόνα 7- 41 και Εικόνα 7- 42) μπορούμε να δούμε μερικά από τα στοιχεία που μπορεί ο διαχειριστής να παρακολουθήσει από τα διαγράμματα του Cacti



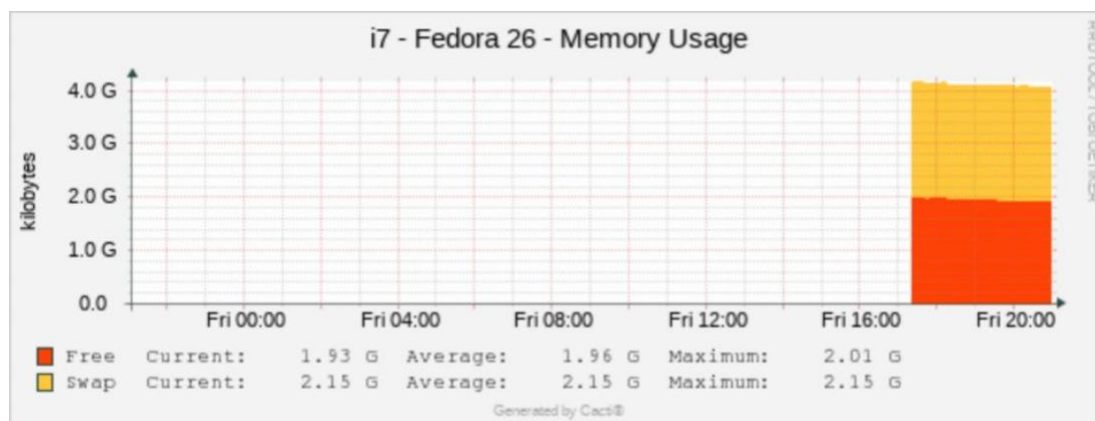
Εικόνα 7- 39: Παρακολούθηση του Disk Space του Linux Server στο Cacti



Εικόνα 7- 40: Παρακολούθηση του Disk Space του Linux Server στο Cacti



Εικόνα 7- 41: Παρακολούθηση του Load Average του Linux Server στο Cacti



Εικόνα 7- 42: Παρακολούθηση της Χρήσης Μνήμης του Linux Server στο Cacti

### Ασκήσεις - Δραστηριότητες

Πραγματοποιήστε την εγκατάσταση της εφαρμογής σύμφωνα με τις οδηγίες και εκτελέστε τις εντολές της παραγράφου και δείτε τα αποτελέσματα που φαίνονται στις εικόνες

#### 7.2.7 Cockpit

Το είδαμε αναλυτικά στο 5<sup>ο</sup> κεφάλαιο στην παράγραφο 5.4.1 πρόκειται για εργαλείο παρακολούθησης αλλά και για εργαλείο διαδικτυακής διαχείρισης του Linux Server.

#### 7.2.8 Webmin

Το είδαμε αναλυτικά στο 5<sup>ο</sup> κεφάλαιο στην παράγραφο 5.4.2 πρόκειται για εργαλείο παρακολούθησης αλλά και για εργαλείο διαδικτυακής διαχείρισης του Linux Server.

## 8 Εγκατάσταση - Διαχείριση Εξυπηρετητών διαδικτυακών υπηρεσιών

### 8.1 Εγκατάσταση παραμετροποίηση και διαχείριση Proxy Server

Proxy Server ή Διακομιστής Μεσολάβησης είναι μια υπηρεσία η οποία επιτρέπει στον διαχειριστή ενός Linux Server αφενός να εξυπηρετεί πιο γρήγορα τους χρήστες που ζητούν διαδικτυακό περιεχόμενο αφετέρου να κάνει διαχείριση του περιεχομένου αυτού.

Ο Proxy Server που θα χρησιμοποιήσουμε σε αυτή την ενότητα είναι ο **Squid**



Εικόνα 8- 1: Εμπορικό σήμα του squid

Ο Squid είναι ένας Proxy Server (διακομιστής μεσολάβησης) για το πρωτόκολλο HTTP. Χρησιμοποιείται κυρίως για τη λειτουργία της προσωρινής αποθήκευσης και προώθησης ιστοσελίδων καθώς και για τη διαχείριση περιεχομένου. Έχει μια ευρεία ποικιλία χρήσεων, συμπεριλαμβανομένης της επιτάχυνσης ενός διακομιστή ιστού με την αποθήκευση επαναλαμβανόμενων αιτημάτων, την προσωρινή αποθήκευση ιστού, το DNS και άλλες αναζητήσεις δικτύου υπολογιστών για μια ομάδα ατόμων που μοιράζονται πόρους δικτύου και βοηθούν την ασφάλεια φιλτράροντας την κίνηση.

Αν και χρησιμοποιείται κυρίως για HTTP και FTP, το Squid περιλαμβάνει περιορισμένη υποστήριξη για πολλά άλλα πρωτόκολλα, όπως το Internet Gopher, SSL, TLS και HTTPS. Το Squid σχεδιάστηκε αρχικά για να λειτουργεί ως υπηρεσία διαχείρισης περιεχομένου σε συστήματα τύπου Unix. Το Squid είναι ελεύθερο λογισμικό και κυκλοφορεί κάτω από την άδεια GNU/GPL (General Public License) (GNU\_GPL, n.d.).

Η εγκατάσταση του squid γίνεται με την εντολή

```
[root@ekdda ~]# dnf -y install squid
```

Το αρχείο παραμετροποίησης είναι το

```
[root@ekdda ~]# vi /etc/squid/squid.conf
```

Στη γραμμή 29 προσθέτουμε τα παρακάτω, με τα οποία ορίζουμε μια access list (acl) η οποία συσχετίζεται με το υποδίκτυο 192.168.1.0/24.

```
acl lan src 192.168.1.0/24
http_access allow localhost
```

Αν το υποδίκτυό μας ήταν της μορφής 10.x.y.z τότε η αντίστοιχη εγγραφή θα ήταν

```
acl lan src 10.0.0.0/24
http_access allow localhost
```

Στη συνέχεια στη γραμμή 57 προσθέτουμε μια άδεια στην προηγούμενη access list να λειτουργεί επάνω από το http

```
http_access allow lan
```

και στην τελευταία γραμμή του αρχείου προσθέτουμε τα παρακάτω

```
request_header_access Referer deny all
request_header_access X-Forwarded-For deny all
request_header_access Via deny all
request_header_access Cache-Control deny all
forwarded_for off
```

Στη συνέχεια αφού αποθηκεύσουμε και κλείσουμε το αρχείο παραμετροποίησης εκκινούμε την υπηρεσία και την κάνουμε enable

```
[root@ekdda ~]# systemctl start squid
[root@ekdda ~]# systemctl enable squid
```

Τέλος κάνουμε τις απαραίτητες εγγραφές στο firewall

```
[root@ekdda ~]# firewall-cmd --add-service=squid --permanent
[root@ekdda ~]# firewall-cmd --reload
```

### 8.1.1 Διαχείριση Περιεχομένου

Μετά την εγκατάσταση του Proxy Server Squid, τα προγράμματα περιήγησης ιστού (browsers) μπορούν να ρυθμιστούν ώστε να το χρησιμοποιούν ως Proxy Server για το HTTP, επιτρέποντας στο Squid να διατηρεί αντίγραφα των επιστρεφόμενων σελίδων, τα οποία, μπορούν να χρησιμοποιηθούν σε επαναλαμβανόμενα αιτήματα για τα ίδιες σελίδες. Αυτή η διαδικασία μειώνει το χρόνο πρόσβασης καθώς και το εύρος ζώνης που καταναλώνεται. Όλη αυτή η διαδικασία είναι ιδιαίτερα χρήσιμη ειδικά σε δίκτυα με μικρό εύρος ζώνης.

Την τεχνική της χρήσης Proxy Server χρησιμοποιούν πολύ συχνά οι πάροχοι υπηρεσιών διαδικτύου ώστε να αυξάνουν την ταχύτητα πρόσβασης στους πελάτες τους και τα διάφορα τοπικά δίκτυα που μοιράζονται μια σύνδεση στο διαδίκτυο.

Ένα πρόγραμμα πελάτη διαδικτύου (για παράδειγμα ένας browser) μπορεί να μη χρησιμοποιεί καθόλου διακομιστή μεσολάβησης. Με αυτόν τον τρόπο δεν έχει κανέναν περιορισμό στο τι μπορεί να δει στο διαδίκτυο ο χρήστης. Πράγμα ιδιαίτερα επικίνδυνο, ειδικά όταν ο χρήστης είναι ανήλικος.

Όταν όμως πρέπει να χρησιμοποιηθεί διαχείριση του περιεχομένου που θα μπορεί να βλέπει ο χρήστης, θα πρέπει να γίνουν ορισμένες ρυθμίσεις στον browser του χρήστη (δείτε πιο κάτω για λεπτομέρειες).

Για Proxy Server είτε μπορεί να χρησιμοποιηθεί αυτός που προτείνει ο πάροχος διαδικτύου είτε μπορεί να χρησιμοποιηθεί ο Squid που εγκαταστήσαμε στον Linux Server μας.

Αν θέλουμε να κάνουμε αποκλεισμό κάποιων δικτυακών τόπων ώστε ο χρήστης να έχει περιορισμούς στις σελίδες στις οποίες μπορεί να έχει πρόσβαση, τότε ανοίγουμε το αρχείο παραμετροποίησης

```
[root@ekdda ~]# vi /etc/squid/squid.conf
```

Στη γραμμή 29 προσθέτουμε το παρακάτω, το οποίο ορίζει μια access list, διασυνδέοντάς την με το αρχείο στο οποίο πραγματοποιείται η διαχείριση περιεχομένου. Στην ουσία θέτει περιορισμό στην πρόσβαση σε συγκεκριμένα domains

```
acl bad_url dstdomain "/etc/squid/bad-sites.squid"
```

Στη γραμμή 45 προσθέτουμε το παρακάτω,

```
http_access deny bad_url
```

Το οποίο ενεργοποιεί την access list που δημιουργήσαμε, σε κάθε πρόσβαση http

Αποθηκεύουμε και κλείνουμε το αρχείο

Στη συνέχεια δημιουργούμε το αρχείο **bad-sites.squid** με το οποίο διαχειριζόμαστε το περιεχόμενο που μπορεί να βλέπουν ο χρήστες. Οποιοδήποτε domain εγγραφεί στο συγκεκριμένο αρχείο έχει πλέον φίλτρο στην πρόσβαση από τους χρήστες

```
[root@ekdda ~]# gedit /etc/squid/bad-sites.squid
```

Και προσθέτουμε κάποιες εγγραφές από δικτυακούς τόπους για τους οποίους θέλουμε να φιλτράρουμε την πρόσβαση στους χρήστες

```
www.<domain1>.gr  
.<domain2>.gr
```

Στη συνέχεια αποθηκεύουμε και κλείνουμε το αρχείο

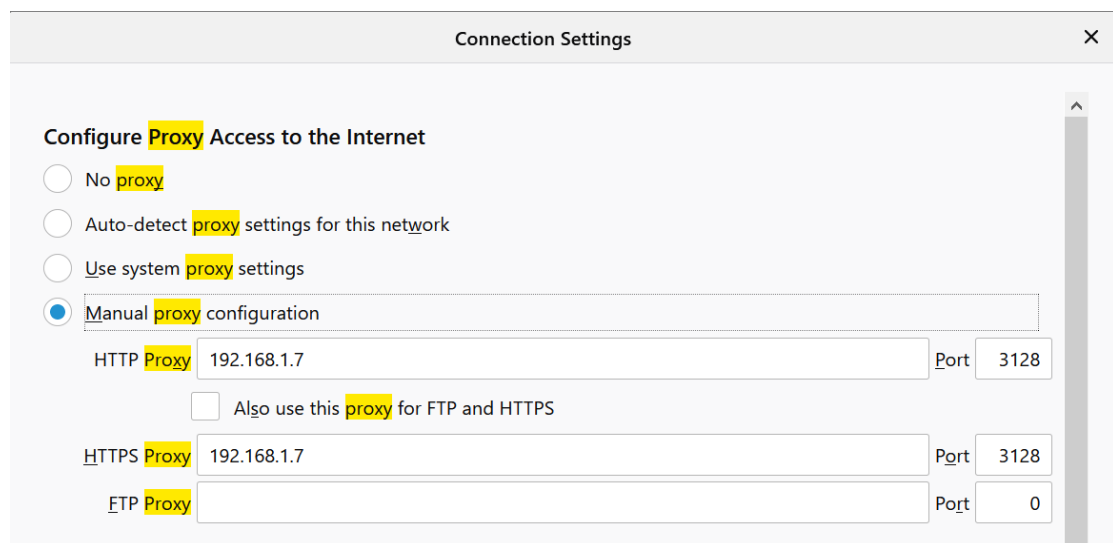
Μετά από οποιαδήποτε αλλαγή στο αρχείο **bad-sites.squid** χρειάζεται επανεκκίνηση ο squid

```
[root@ekdda ~]# systemctl restart squid
```

Προκειμένου να είναι δυνατή η διαχείριση του περιεχομένου που μπορεί να δει ο χρήστης θα πρέπει στις ρυθμίσεις του browser που χρησιμοποιεί ο χρήστης να γίνει ορισμός του Proxy Server όπως φαίνεται στην παρακάτω Εικόνα 8- 2.

Στην παρακάτω Εικόνα 8- 2 υποθέτουμε ότι ο Squid Proxy Server φιλοξενείται στον Linux Server μας ο οποίος έχει IP 192.168.1.7

Η εξ ορισμού (default) πόρτα του Proxy Server μας είναι η 3128



Εικόνα 8- 2: Ορισμός Proxy στον Browser του χρήστη

Η εικόνα που παίρνει ο χρήστης όταν προσπαθεί να έχει πρόσβαση σε ένα domain που έχει αποκλειστεί από τον squid proxy Server είναι η ακόλουθη (Εικόνα 8- 3)



Εικόνα 8- 3: Μήνυμα αποκλεισμού διεύθυνσης URL από τον squid proxy Server

### Ασκήσεις - Δραστηριότητες

Πραγματοποιήστε την εγκατάσταση της εφαρμογής σύμφωνα με τις οδηγίες και εκτελέστε τις εντολές της παραγράφου. Ορίστε ως proxy σε μια συσκευή Windows στον browser την IP του Linux Server σας και κάντε διαχείριση περιεχομένου όπως περιγράφεται στην ενότητα. Δείτε το μήνυμα αποκλεισμού στον browser του χρήστη των Windows.

## 8.2 Εγκατάσταση παραμετροποίηση και διαχείριση Apache Web Server

Web Server είναι μια υπηρεσία που εξυπηρετεί αιτήματα εμφάνισης αρχείων με συστατικά στοιχεία ιστοσελίδας (έγγραφα HTML, εικόνες, φύλλα στυλ CSS, JavaScript κτλ). Πολλές φορές ως Web Server αναφέρεται και ο υπολογιστής που φιλοξενεί την υπηρεσία. Με πολύ απλά λόγια Web Server είναι η υπηρεσία που εμφανίζει το διαδικτυακό περιεχόμενο που φιλοξενεί, σε όποιον (τοπικό ή απομακρυσμένο χρήστη-πελάτη) το ζητήσει.

Apache είναι ένας πολύ δημοφιλής Web Server. Άλλοι Web Servers είναι ο Nginx, ο Lighthpd, ο IIS (Microsoft) κτλ.

Η εγκατάσταση του Apache Web Server γίνεται με την (εμβληματική) εντολή

```
[root@ekdda ~]# dnf -y install httpd
```

Ο apache είναι πάντα προεγκατεστημένος με τη Fedora. Ο βασικός κατάλογος στον οποίο περιέχονται τα αρχεία παραμετροποίησής του είναι ο

```
/etc/httpd/
```

Στον κατάλογο αυτόν εκτός από το βασικό αρχείο παραμετροποίησης του Apache θα βρούμε και άλλα αρχεία παραμετροποίησης τα οποία ανήκουν σε εφαρμογές που έχουν διαδικτυακή διεπαφή (web interface). Τέτοιες εφαρμογές είναι το Cockpit, το Webmin, το PhpMyAdmin κτλ.

Το βασικό αρχείο παραμετροποίησης του Apache είναι το

```
[root@ekdda ~]# gedit /etc/httpd/conf/httpd.conf
```

Γενικά ο Apache έχει πάρα πολλά στοιχεία παραμετροποίησης τα οποία θα μπορούσαν να αποτελέσουν και αντικείμενο ενός ολόκληρου σεμιναρίου. Μια πολύ αναλυτική περιγραφή της παραμετροποίησης του Apache μπορούμε να βρούμε εδώ (Apache\_org, n.d.)

Μερικά πολύ βασικά στοιχεία παραμετροποίησης το Apache είναι τα ακόλουθα

στη γραμμή 41 ορίζουμε την IP καθώς και την πόρτα στην οποία «ακούει» ο Apache. Η IP είναι του Linux Server και η πόρτα είναι η προεπιλεγμένη (default). Μπορούμε αν θέλουμε να αλλάξουμε την προεπιλεγμένη πόρτα σε αυτό το σημείο.

```
# Listen: Allows you to bind Apache to specific IP addresses
# and/or ports, instead of the default.
#
# Change this to Listen on specific IP addresses as shown below
# to prevent Apache from glomming onto all bound IP addresses.
#
Listen 192.168.1.48:80
```

Αν δε θέλουμε να αντιστοιχίσουμε τον web Server σε κάποια IP αλλά θέλουμε να λειτουργεί μόνο ως localhost, γράφουμε μόνο

```
Listen 80
```

συνήθως επιλέγουμε να λειτουργεί μόνο ως localhost όταν ο Web Server είναι στη φάση προετοιμασίας, παραμετροποίησης και δοκιμαστικής λειτουργίας του. Στην κατάσταση αυτή (localhost mode) τον Web Server μπορούν να τον χρησιμοποιήσουν μόνο οι τοπικοί χρήστες του Linux Server.

Στη γραμμή 86 ορίζουμε το email του διαχειριστή του Apache Web Server.

```
# ServerAdmin: Your address, where problems with the server
# should be e-mailed. This address appears on some server-
# generated pages, such as error documents. e.g. admin@your-
# domain.com
#
# ServerAdmin root@localhost
ServerAdmin root@ekdda.gr
```



Στη γραμμή 95 ορίζουμε το όνομα του Server (domain name) το οποίο πρέπει να έχουμε κατοχυρώσει από κάποιον καταχωρητή. Έστω ότι έχουμε κατοχυρώσει το όνομα `www.my_domain.gr`

```
# ServerName gives the name and port that the server uses to
# identify itself. This can often be determined automatically,
# but we recommend you specify # it explicitly to prevent
# problems during startup.
# If your host doesn't have a registered DNS name, enter its IP
# address here.
ServerName www.my_domain.gr:80
```

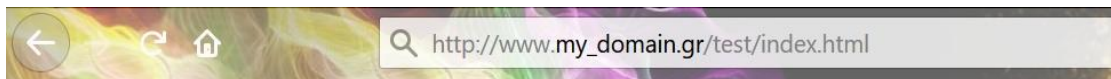
Στη γραμμή 164 ορίζουμε τα ονόματα αρχείων τα οποία θα υπάρχει δυνατότητα πρόσβασης (αιτήματος προβολής εκ μέρους των χρηστών) με το όνομα του καταλόγου

```
# ServerName gives the name and port that the server uses to
# DirectoryIndex: sets the file that Apache will serve if a
# directory
# is requested.
#
<IfModule dir_module>
    DirectoryIndex index.html index.cgi index.php
</IfModule>
```

Η σειρά είναι σημαντική και υποδηλώνει τη σειρά με την οποία αναζητούνται τα αρχεία από το σύστημα

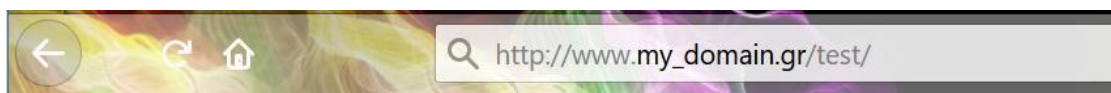
Για παράδειγμα αν έχουμε κάτω από τον κατάλογο `test` το αρχείο με όνομα `index.html`

Τότε δε χρειάζεται η αίτηση για την ιστοσελίδα να γίνει με την πλήρη διαδρομή (Εικόνα 8-4).



Εικόνα 8- 4: Αναζήτηση ιστοσελίδας με πλήρη διαδρομή

αλλά μόνο με το όνομα του directory (Εικόνα 8- 5).



Εικόνα 8- 5: Αναζήτηση ιστοσελίδας μόνο με κατάλογο

Αν στον ίδιο υποκατάλογο υπάρχουν τρία αρχεία, τα `index.html`, `index.cgi`, `index.php`, τότε θα εμφανιστεί μόνο το περιεχόμενο του πρώτου

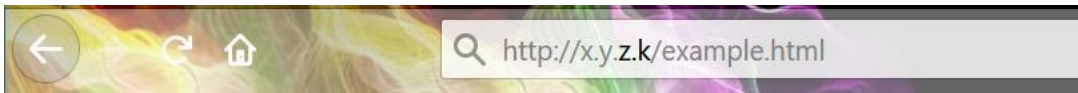
Αν στον ίδιο υποκατάλογο υπάρχει μόνο το αρχείο **index.php**, τότε θα εμφανιστεί μόνο αυτό παρά το γεγονός ότι δεν υπάρχει κάποιο από τα άλλα που υπάρχουν στην παραπάνω εγγραφή στο `http.conf` (γραμμή 164)

### 8.2.1 Κατάλογος-διαδρομή διαδικτυακού περιεχομένου

Ο κατάλογος στον οποίο τοποθετούνται τα αρχεία με διαδικτυακό περιεχόμενο (ιστοσελίδες κτλ) στα οποία θα έχουν πρόσβαση οι χρήστες (τοπικοί και απομακρυσμένοι) είναι ο

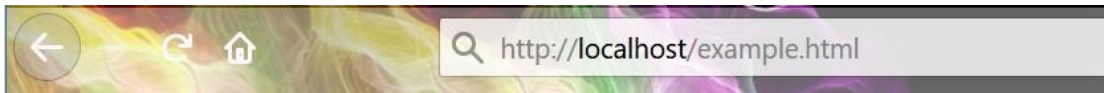
```
/var/www/html/
```

Αν για παράδειγμα ο Server μας έχει IP `x.y.z.k` και έχουμε τοποθετήσει την ιστοσελίδα `example.html` μέσα στον προηγούμενο κατάλογο, τότε μπορούμε από έναν φυλλομετρητή (browser πχ `firefox`, `chrome`, `safari` κτλ) να δούμε το περιεχόμενο της ιστοσελίδας αν πληκτρολογήσουμε τη διεύθυνση URL (Εικόνα 8- 6).



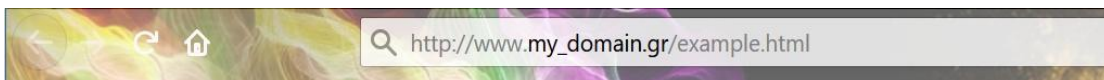
Εικόνα 8- 6: Διαδικτυακή διεύθυνση στην οποία βρίσκεται σελίδα που φιλοξενεί ο Apache

Επιπλέον αν ο χρήστης είναι τοπικός, τότε μπορεί να αναζητήσει τις ιστοσελίδες χρησιμοποιώντας το λεκτικό `localhost` ως εξής (Εικόνα 8- 7):



Εικόνα 8- 7: Διαδικτυακή διεύθυνση με τη χρήση του λεκτικού `localhost`

Και φυσικά αν έχουμε αγοράσει κάποιος domain πχ το `my_domain.gr` και το έχουμε αντιστοιχίσει σωστά στην IP μας μέσω κάποιου **Name Server** τότε οι χρήστες (τοπικοί και απομακρυσμένοι) μπορούν να αναζητήσουν την ιστοσελίδα που φιλοξενεί ο Apache στη διεύθυνση URL (Εικόνα 8- 8).



Εικόνα 8- 8: Διαδικτυακή διεύθυνση με τη χρήση domain

### 8.2.2 Virtual Servers στον Apache

Εκτός από τον κανονικό Web Server στον Apache μπορούν να οριστούν και εικονικοί (virtual Servers

Στη γραμμή 84, κάνουμε την παρακάτω εγγραφή

```
NameVirtualHost www.test.com:80
NameVirtualHost www.test2.com:80

<VirtualHost www.test.com:80>
  DocumentRoot "/var/www/example1"
  ServerName test.com
  ServerAlias www.test.com
</VirtualHost>

<VirtualHost www.test2.com:80>
  DocumentRoot "/var/www/example2"
  ServerName test2.com
  ServerAlias www.test2.com
</VirtualHost>
```

Αυτό που καταφέρνουμε ορίζοντας virtual hosts είναι να φιλοξενούμε πολλαπλά domains στον ίδιο Server. Όπως φαίνεται στην παραπάνω εγγραφή το περιεχόμενο του domain test.com βρίσκεται στη διαδρομή του Linux Server

/var/www/example1/

Και οι χρήστες το ζητάνε από τον browser ως εξής (Εικόνα 8-9) :



Εικόνα 8- 9: URL εικονικού Server στον Apache

Ενώ το περιεχόμενο του domain test2.com βρίσκεται στη διαδρομή του Linux Server

/var/www/example2/

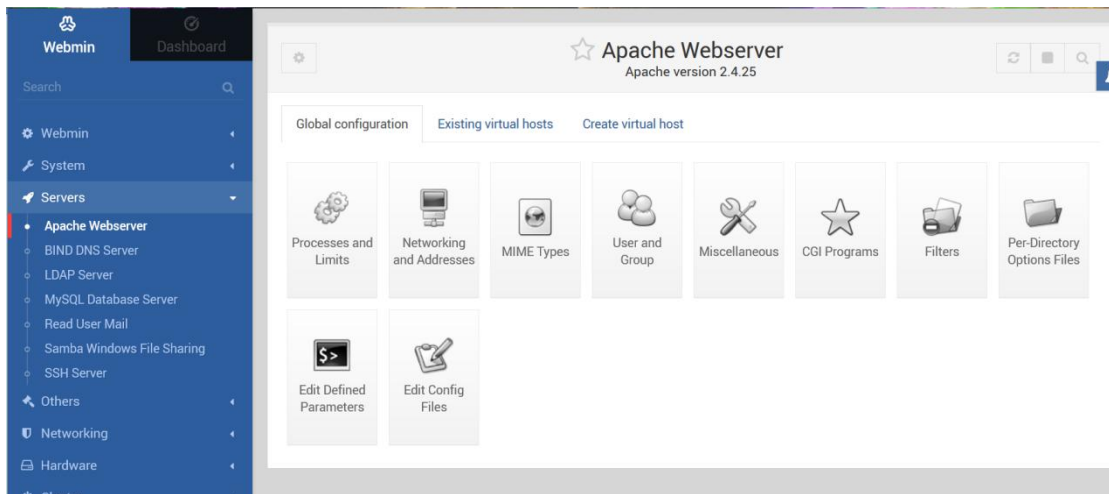
Και οι χρήστες το ζητάνε από τον browser ως εξής (Εικόνα 8- 10) :



Εικόνα 8- 10: URL εικονικού Server στον Apache

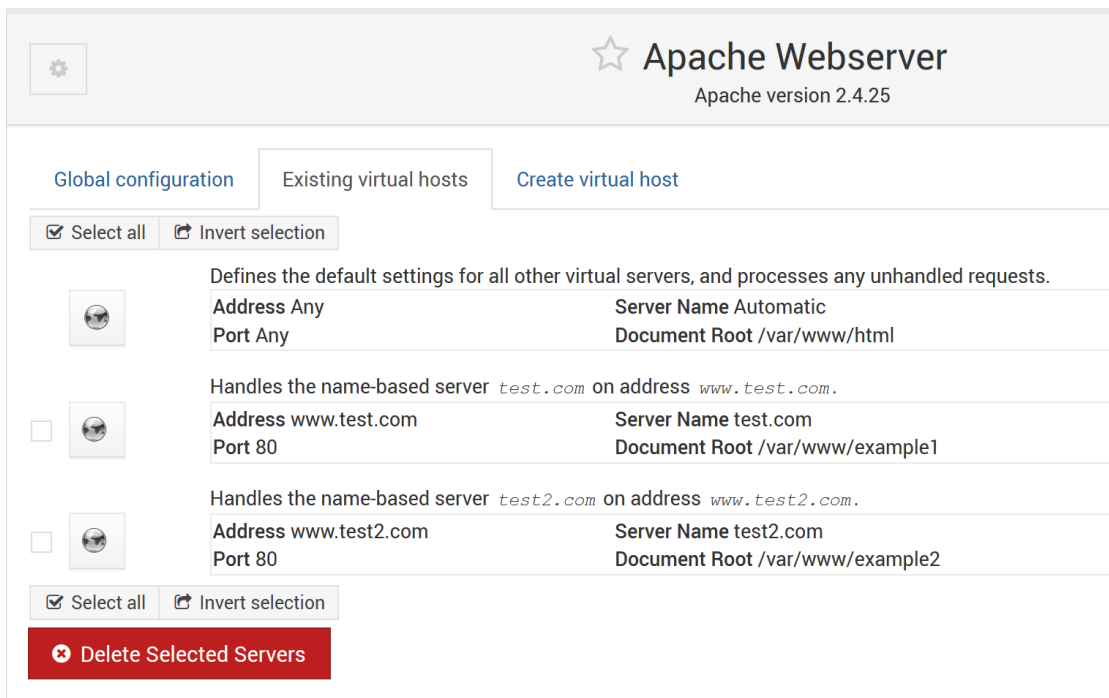
### 8.2.3 Ο Apache Web Server στο Webmin

Ο Apache Web Server μπορεί επίσης να εγκατασταθεί και να παραμετροποιηθεί από το περιβάλλον του Webmin (Εικόνα 8- 11)



Εικόνα 8- 11: Διαχείριση του Apache μέσα από το διαχειριστικό εργαλείο Webmin

Στην παρακάτω Εικόνα 8- 12 μπορούμε να δούμε και τους δυο Virtual Servers που δημιουργήσαμε πιο πάνω



Εικόνα 8- 12: Virtual Hosts στον Apache έτσι όπως φαίνονται από το Webmin

Επιπλέον μπορούμε στην παραπάνω Εικόνα 8- 12 να διακρίνουμε τόσο τους virtual όσο και τον κανονικό Server καθώς και στις διαδρομές στις οποίες πρέπει να βρίσκεται το διαδικτυακό περιεχόμενο του καθενός

### *Ασκήσεις - Δραστηριότητες*

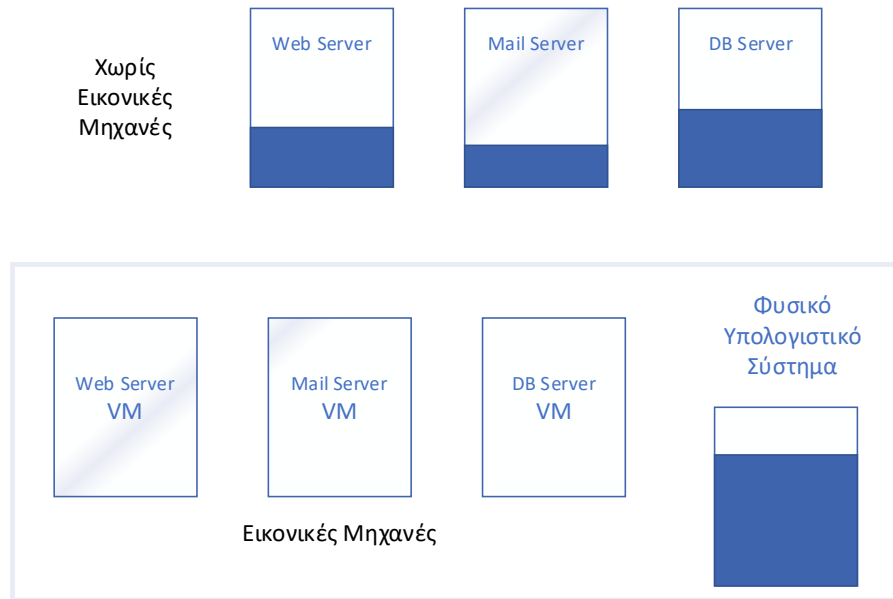
- Πραγματοποιήστε την εγκατάσταση του Apache web Server σύμφωνα με τις οδηγίες και εκτελέστε τις εντολές της παραγράφου και δείτε τα αποτελέσματα που φαίνονται στις εικόνες
- Δημιουργήστε μια ιστοσελίδα ο καθένας και δείτε την πώς φαίνεται από τον browser
- Δημιουργήστε έναν δικτυακό τόπου σε wordpress
- Παραμετροποιήστε τον Apache από το περιβάλλον του Webmin

## 9 Διαχείριση εικονικών μηχανών

### 9.1 Βασικές έννοιες

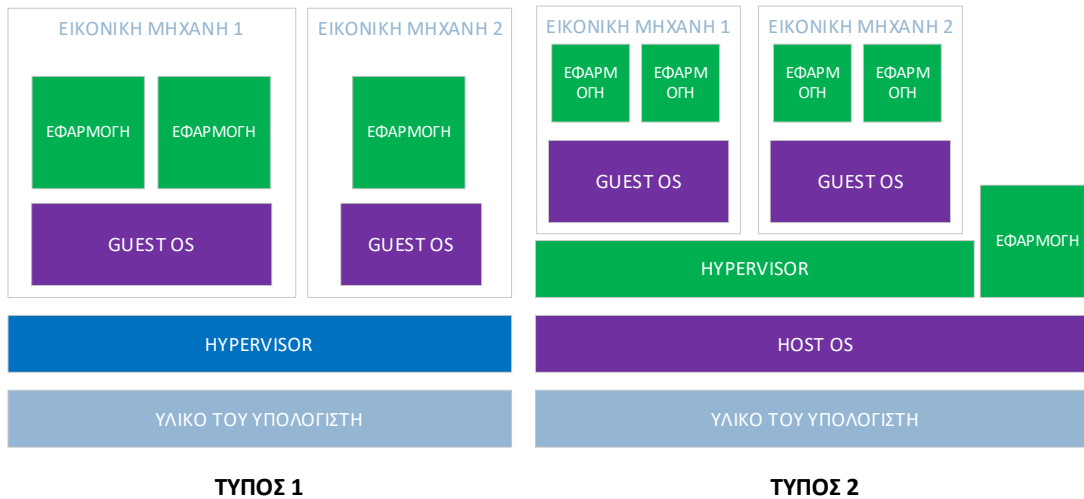
Η τεχνολογία χρήσης των εικονικών μηχανών (virtualization) είναι η δημιουργία «εικονικών» υπολογιστών με τη χρήση λογισμικού μέσα σε ένα κανονικό υπολογιστικό σύστημα (RedHat-Virtualization, n.d.). Οι «εικονικοί» υπολογιστές ονομάζονται «**εικονικές μηχανές**» (**Virtual Machines**) ή «φιλοξενούμενοι» (guest) ενώ το υπολογιστικό σύστημα που τους «φιλοξενεί» ονομάζεται «οικοδεσπότης» (host). Η κάθε μία εικονική μηχανή λειτουργεί σαν ξεχωριστός υπολογιστής με διαφορετικό λειτουργικό σύστημα και εφαρμογές. Μερικά από τα πλεονεκτήματα που προσφέρει η χρήση εικονικών μηχανών είναι τα παρακάτω (IBM, 2019):

- Εξοικονόμηση πόρων. Με τη χρήση εικονικών μηχανών μπορούμε να χρησιμοποιήσουμε τους ίδιους υπολογιστικούς πόρους, π.χ. επεξεργαστές, μνήμες κτλ., για να εκτελέσουμε διάφορες λειτουργικότητες ακόμα και αν αυτές απαιτούν διαφορετικό λειτουργικό σύστημα.
- Ευκολία στη διαχείριση. Με την αντικατάσταση των πραγματικών υπολογιστών με εικονικές μηχανές είναι ευκολότερη η δημιουργία και εφαρμογή πολιτικών που είναι γραμμένες σε λογισμικό. Για παράδειγμα η δημιουργία και εφαρμογή πολλών εικονικών μηχανών «κλώνων» είναι πολύ ευκολότερη από τη δημιουργία πραγματικών υπολογιστών «κλώνων».
- Ελάχιστος χρόνος διακοπής. Η λειτουργία μιας εικονικής μηχανής απαιτεί σημαντικά μικρότερο κόστος από τη λειτουργία ενός αντίστοιχου πραγματικού υπολογιστή. Επομένως, πλέον είναι δυνατή η ύπαρξη πλεοναζόντων εικονικών μηχανών για την υποστήριξη της λειτουργικότητας σε περίπτωση που υπάρχει πρόβλημα με την πρωταρχική.
- Ταχύτερη τροφοδότηση. Η αγορά, εγκατάσταση και ρύθμιση ενός πραγματικού υπολογιστικού συστήματος είναι πολύ πιο χρονοβόρα από την ανάπτυξη μιας εικονικής μηχανής.



Εικόνα 9- 1: Χρήση Εικονικών Μηχανών στους Servers

Τα πλεονεκτήματα των εικονικών μηχανών βρήκαν πρόσφορο πεδίο εφαρμογής στους Servers και οδήγησαν τελικά στη δημιουργία της τεχνολογίας του **υπολογιστικού νέφους (cloud computing)**. Πριν τη ύπαρξη των εικονικών μηχανών, κάθε μια από τις υπηρεσίες που προσέφερε ένας οργανισμός λειτουργούσε σε διαφορετικό υπολογιστή, όπως παρουσιάζεται στο πάνω μέρος της Εικόνα 9- **1Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε..** Με αυτόν το τρόπο, μπορούσαν να δημιουργηθούν διαφορετικές πολιτικές σε κάθε υπολογιστή έτσι ώστε να παρέχεται περισσότερη ασφάλεια. Επιπρόσθετα, ένα τεχνικό πρόβλημα σε έναν υπολογιστή δεν θα σταματούσε όλες τις υπηρεσίες. Όμως η συγκεκριμένη σχεδιαστική επιλογή δημιουργούσε το πρόβλημα ότι γινόταν μειωμένη αξιοποίηση του υλικού επειδή πολλές υπηρεσίες δεν είχαν τόσο μεγάλο φόρτο εργασίας. Με τη χρήση των εικονικών μηχανών υπάρχει η δυνατότητα χρήσης ενός υπολογιστή που φιλοξενεί μέσα του όλες τις εικονικές μηχανές που υποστηρίζουν τις διάφορες υπηρεσίες. Η αρχιτεκτονική αυτή επιλογή παρουσιάζεται στο κάτω μέρος της Εικόνα 9- 1. Επομένως έχουμε από τη μία πλήρη αξιοποίηση του υλικού αλλά από την άλλη διατηρούμε την «απομόνωση» της λειτουργίας της κάθε εικονικής μηχανής.



Εικόνα 9- 2: Τύποι Hypervisor

Για την υποστήριξη εικονικών μηχανών από ένα σύστημα είναι απαραίτητη η ύπαρξη ενός ειδικού λογισμικού που ονομάζεται **hypervisor** (IBM, 2019). Το λογισμικό αυτό αποτελεί έναν ενδιάμεσο ανάμεσα στις εικονικές μηχανές και το υλικό του υπολογιστή. Με αυτό τον τρόπο η κάθε εικονική μηχανή έχει πρόσβαση στους πόρους του υλικού που χρειάζεται για να λειτουργήσει. Επιπρόσθετα ο hypervisor είναι υπεύθυνος ώστε οι εικονικές μηχανές να μην «παρεμβάλλονται» η μία στην άλλη αλλά να έχουν οι κάθε μια τη δική της μνήμη και αποθηκευτικό χώρο. Υπάρχουν δύο τύποι hypervisor (Εικόνα 9- 2):

- Τύπος 1. Σε αυτόν τον τύπο το πρόγραμμα hypervisor επιδρά κατευθείαν πάνω στο υλικό παρακάμπτοντας το λειτουργικό σύστημα. Αυτός είναι ο τύπος που χρησιμοποιείται πιο συχνά στους servers.
- Τύπος 2. Σε αυτόν τον τύπο το πρόγραμμα hypervisor λειτουργεί σαν κανονικό πρόγραμμα που χρησιμοποιεί το υπάρχον λειτουργικό σύστημα για να έχει πρόσβαση στο υλικό του υπολογιστή. Όπως είναι φυσικό σε αυτή την περίπτωση η απόδοση είναι μειωμένη.

Το λειτουργικό σύστημα Linux υποστηρίζει διάφορες τεχνολογίες εικονικών μηχανών πολλές από τις οποίες είναι και «ανοιχτού κώδικα» (Son, 2020). Μερικές από τις τεχνολογίες αυτές ανήκουν στον τύπο 1, π.χ. το KVM, και άλλες στον τύπο 2, π.χ. το Virtual Box ή το VM Ware. Στα παραδείγματα που ακολουθούν χρησιμοποιούμε το KVM (KVM, n.d.) επειδή υποστηρίζεται από το απευθείας Kernel (έκδοση 2.6.20 και μετά). Επίσης θα χρησιμοποιήσουμε και το Libvirt (LibVirt, n.d.) που είναι ένα API για τη σύνδεση με τον KVM. Τέλος θα παρουσιάσουμε και το QEMU (QEMU, n.d.) που χρησιμοποιείται σαν γραφική πλατφόρμα για την εκτέλεση των εικονικών μηχανών.

## 9.2 Προετοιμασία για την υποστήριξη εικονικών μηχανών

Αρχικό στάδιο για την υποστήριξη εικονικών μηχανών είναι να γνωρίζουμε αν ο επεξεργαστής μας διαθέτει δυνατότητα εγγενούς υποστήριξης εικονικών μηχανών. Αυτή τη στιγμή και οι δύο βασικές εταιρίες παραγωγής επεξεργαστών (Intel & AMD) παρέχουν αυτή τη δυνατότητα η οποία όμως θα πρέπει να είναι ενεργοποιημένη από το BIOS (FedoraDocs-



Virtualization, n.d.). Για να διαπιστώσουμε αν ο επεξεργαστής μας διαθέτει υποστήριξη εικονικών μηχανών και αν αυτή είναι ενεργοποιημένη εκτελούμε την παρακάτω εντολή

```
[user@ekdda ~]$ grep -E 'svm|vmx' /proc/cpuinfo
```

Η παραπάνω εντολή θα πρέπει να μας δώσει τη λέξη `vmx` αν διαθέτουμε επεξεργαστή της εταιρίας Intel ή τη λέξη `svm` αν διαθέτουμε επεξεργαστή της εταιρίας AMD.

Επόμενο στάδιο είναι να δούμε αν ο kernel που διαθέτουμε υποστηρίζει το KVM. Αυτό μπορούμε να το διαπιστώσουμε αν είναι «φορτωμένο» το KVM module με την παρακάτω εντολή.

```
[user@ekdda ~]$ lsmod | grep kvm
kvm_intel          299008  0
kvm                765952  1 kvm_intel
irqbypass         16384   1 kvm
```

Τελευταίο στάδιο είναι η εγκατάσταση όλων των λογισμικών που είναι απαραίτητα για τη δημιουργία και την εκτέλεση μιας εικονικής μηχανής. Στη διανομή Fedora τα λογισμικά αυτά έχουν συγκεντρωθεί στην ομάδα λογισμικών *Virtualization*.

```
[user@ekdda ~]$ dnf group info Virtualization
Last metadata expiration check: 0:07:06 ago on Tue 22 Dec 2020
12:42:47 PM EET.

Group: Virtualization
Description: These packages provide a graphical virtualization
environment.
Mandatory Packages:
  virt-install
Default Packages:
  libvirt-daemon-config-network
  libvirt-daemon-kvm
  qemu-kvm
  virt-manager
  virt-viewer
Optional Packages:
  guestfs-browser
  libguestfs-tools
  python3-libguestfs
  virt-top
```

Επομένως για να εγκαταστήσουμε την ομάδα λογισμικών εκτελούμε την παρακάτω εντολή

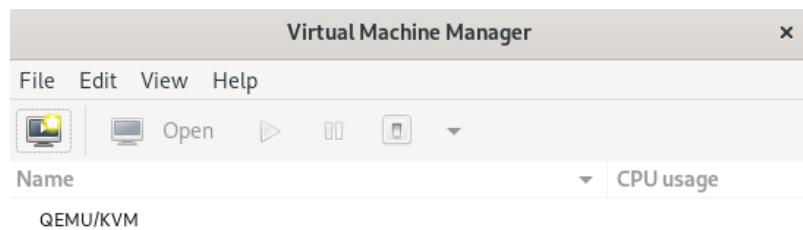
```
[user@ekdda ~]$ sudo dnf group install Virtualization
```

### 9.3 Δημιουργία και εκτέλεση μιας εικονικής μηχανής

Η δημιουργία και η εκτέλεση μιας εικονικής μηχανής μπορεί να γίνει είτε με το γραφικό εργαλείο virtual manager είτε με τη χρήση εντολών από το τερματικό.

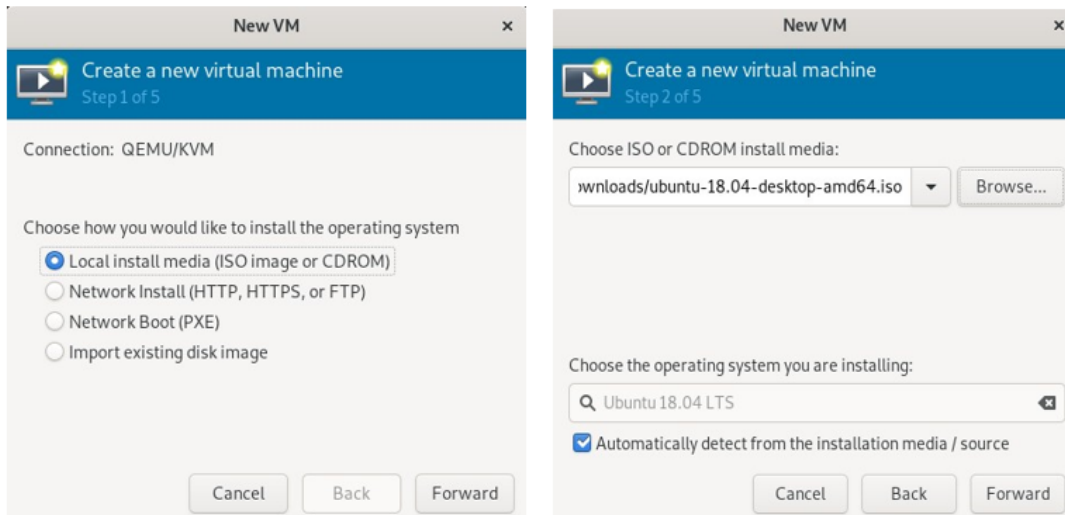
#### 9.3.1 Δημιουργία εικονικής μηχανής με γραφικό τρόπο

Το εργαλείο **Virtual Machine Manager** (VM Manager) είναι ένα γραφικό εργαλείο που επικοινωνεί με το KVM και επιτρέπει τη δημιουργία και διαχείριση εικονικών μηχανών. Μπορούμε να ξεκινήσουμε το εργαλείο είτε από το *Activities -> Virtual Machine Manager* είτε μπορούμε να γράψουμε στο τερματικό την εντολή **virt-manager**. Το VM Manager θέλει δικαιώματα διαχειριστή και σε περίπτωση που το εκτελέσουμε σαν απλός χρήστης θα εμφανίσει ένα μήνυμα που θα ζητάει το συνθηματικό του root και στη συνέχεια θα παρουσιάσει το παράθυρο της Εικόνα 9- 3.



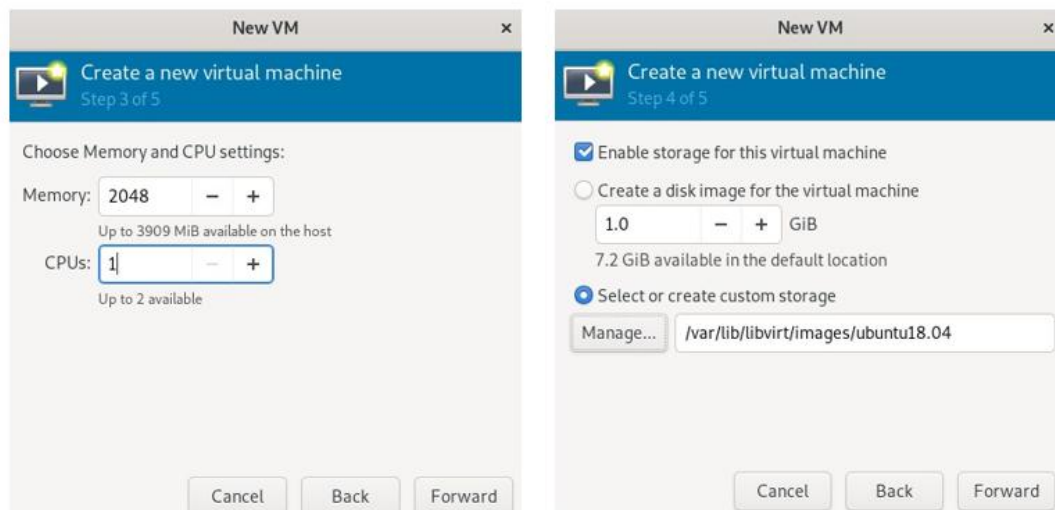
Εικόνα 9- 3: Το εργαλείο Virtual Machine Manager

Από το μενού επιλέγουμε *File->New Virtual Machine* ή το αντίστοιχο εικονίδιο που βρίσκεται στο πάνω αριστερό μέρος του παραθύρου. Αποτέλεσμα της ενέργειας αυτής είναι η εμφάνιση του οδηγού δημιουργίας μιας εικονικής μηχανής που αποτελείται από πέντε βήματα. Τα δύο πρώτα βήματα του οδηγού εμφανίζονται στην Εικόνα 9- 4.



Εικόνα 9- 4: Βήματα 1 & 2 του οδηγού δημιουργίας VM

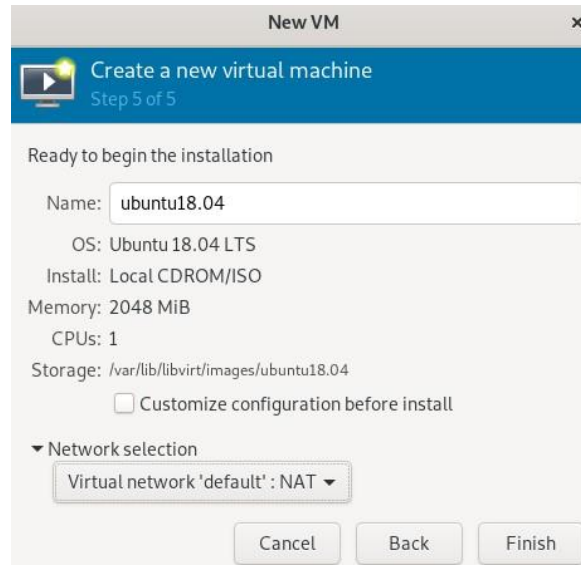
Όπως παρατηρούμε και στο αριστερό μέρος της Εικόνα 9- 4, το πρώτο βήμα του οδηγού είναι η επιλογή του μέσου από το οποίο θα γίνει η εγκατάσταση του λειτουργικού της εικονικής μηχανής. Μπορούμε να επιλέξουμε από τοπικό μέσο, ISO ή CD ROM είτε από μία τοποθεσία δικτύου. Μπορεί ακόμα να γίνει εισαγωγή της εικόνας δίσκου μιας υπάρχουσας εικονικής μηχανής. Στο δεύτερο βήμα του οδηγού δίνουμε περισσότερες λεπτομέρειες για το μέσο εγκατάστασης. Στη συγκεκριμένη περίπτωση, δεξί μέρος της Εικόνα 9- 4, έχουμε επιλέξει ISO που βρίσκεται αποθηκευμένο σε έναν κατάλογο στο δίσκο. Με βάση αυτό το ISO ο VM Manager μπορεί να διαλέξει αυτόματα το είδος του λειτουργικού που θα γίνει εγκατάσταση ή μπορούμε να το ρυθμίσουμε και εμείς.



Εικόνα 9- 5: Βήματα 3 & 4 του οδηγού δημιουργίας VM

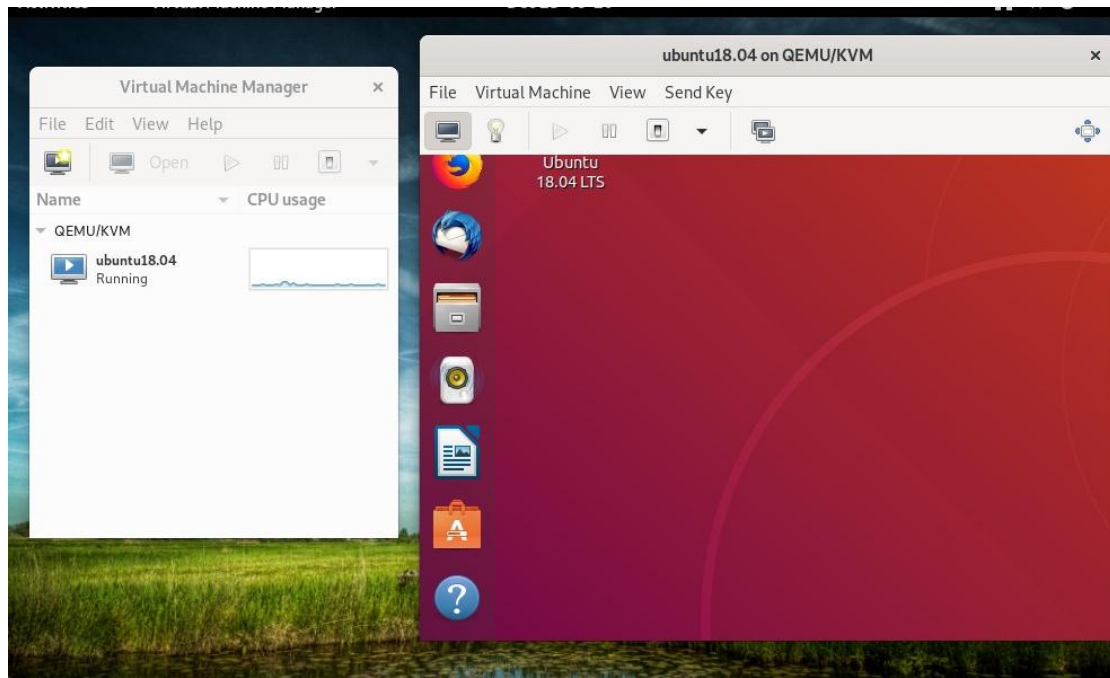
Στο τρίτο βήμα του οδηγού δημιουργίας εικονικής μηχανής, αριστερό μέρος της Εικόνα 9- 5 **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.**, θα πρέπει να επιλέξουμε το μέγεθος της μνήμης που θα έχει η εικονική μηχανή καθώς και το πλήθος των επεξεργαστών. Για να μας βοηθήσει ο VM Manager μας εμφανίζει ακριβώς κάτω από την

επιλογή μας το μέγιστο που μπορούμε να διαθέσουμε. Στο τέταρτο βήμα του οδηγού που εμφανίζεται στο δεξί μέρος της Εικόνα 9- 5 πρέπει να επιλέξουμε τον αποθηκευτικό χώρο που θα διαθέτει η εικονική μας μηχανή. Μπορούμε να επιλέξουμε είτε από τον προκαθορισμένο χώρο είτε να δημιουργήσουμε εμείς ένα καινούργιο αρχείο που θα λειτουργήσει σαν εικονικό αποθηκευτικός χώρος. Όπως παρατηρούμε και στην Εικόνα 9- 5 ο προκαθορισμένος κατάλογος για τη δημιουργία των εικονικών αποθηκευτικών μέσων είναι ο `/var/lib/libvirt/images`.



Εικόνα 9- 6: Βήμα 5 του οδηγού δημιουργίας VM

Το πέμπτο και τελευταίο βήμα του οδηγού δημιουργίας εικονικής μηχανής (Εικόνα 9- 6) , είναι μια ανασκόπηση των επιλογών μας καθώς επίσης μας παρέχει τη δυνατότητα να ρυθμίσουμε τη σύνδεση δικτύου που θα έχει η εικονική μας μηχανή. Επιλέγοντας *Finish* ξεκινά, σε ένα καινούργιο παράθυρο, μια εφαρμογή που ονομάζεται QEMU όπου και πραγματοποιείται η εγκατάσταση του λειτουργικού συστήματος στην εικονική μηχανή. Αφού τελειώσει η εγκατάσταση πλέον θα έχουμε ανοιχτά δύο παράθυρα. Το πρώτο είναι ο VM Manager που περιέχει τις εικονικές μηχανές που έχουμε δημιουργήσει και το δεύτερο είναι η εφαρμογή QEMU όπου θα εκτελείται το λειτουργικό σύστημα της εικονικής μηχανής που λειτουργεί όπως παρουσιάζεται και στην Εικόνα 9- 7.



Εικόνα 9- 7: Λειτουργία εικονικής μηχανής

### 9.3.2 Δημιουργία εικονικής μηχανής από τη γραμμή εντολών

Για να δημιουργήσουμε μια εικονική μηχανή από τη γραμμή εντολών θα πρέπει να γνωρίζουμε πόση μνήμη και πόσους επεξεργαστές θα της αναθέσουμε. Μπορούμε να δούμε πόση μνήμη και επεξεργαστές έχει ο υπολογιστής μας με τη χρήση των εντολών **free -h** και **lscpu**. Στη συνέχεια θα πρέπει να δημιουργήσουμε ένα ειδικό αρχείο που θα περιέχει τον αποθηκευτικό χώρο της εικονικής μηχανής. Ο προκαθορισμένος κατάλογος που αποθηκεύονται τα αρχεία αυτά είναι ο **/var/lib/libvirt/images**. Τα αρχεία που είναι αποθηκευτικοί χώροι εικονικών μηχανών μπορούν να έχουν διάφορα formάτ, εμείς θα δημιουργήσουμε ένα αρχείο με το formάτ qcow2 (LibVirt, n.d.). Δημιουργούμε το αρχείο μεγέθους 10GB με την παρακάτω εντολή

```
[root@ekdda ~]# qemu-img create -f qcow2
/var/lib/libvirt/images/guest.qcow2 10240
```

Στη συνέχεια πραγματοποιούμε τη δημιουργία της εικονικής μηχανής μέσω του Virt API με τη χρήση της εντολής **virt-install**

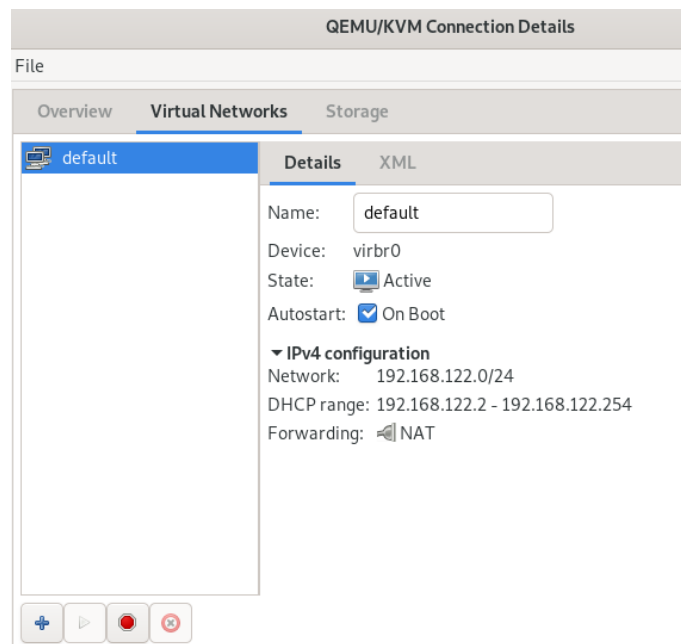
```
[root@ekdda ~]# virt-install --name Ubuntu \
--description 'Ubuntu 18.04 Workstation' \
--ram 2048 \
--vcpus 1 \
--disk path=/var/lib/libvirt/images/guest.img,size=10 \
--os-type linux \
--os-variant generic \
--cdrom /home/user/Downloads/ubuntu-18.04-desktop-amd64.iso
```

Όπως βλέπουμε στην εντολή `virt-install` δίνουμε με παραμέτρους τις διάφορες ρυθμίσεις που κάναμε με γραφικό τρόπο στο εργαλείο VM Manager. Για μια αναλυτική παράθεση των παραμέτρων μπορούμε να δούμε τη σελίδα αναφοράς της εντολής (ManPage-Virt, n.d.).

## 9.4 Υποστήριξη δικτύου στις εικονικές μηχανές

Ένα υπολογιστικό σύστημα συνδέεται στο δίκτυο με τη χρήση μιας κάρτας δικτύου και μιας δικτυακής συσκευής (switch ή router). Σε ένα σύστημα που υποστηρίζει εικονικές μηχανές κάθε μία από τις εικονικές μηχανές έχει μία εικονική κάρτα δικτύου και μπορεί να υπάρχουν παραπάνω από ένα εικονικά δίκτυα. Ο Hypervisor είναι υπεύθυνος για να αναθέτει εικονικές κάρτες δικτύου στις εικονικές μηχανές και να τις συνδέει με το κατάλληλο εικονικό δίκτυο. Τέλος όλη αυτή η πληροφορία αν χρειάζεται περνάει μέσα από την κανονική κάρτα δικτύου του υπολογιστή για να συνδεθεί με το internet.

Ο VM Manager μπορεί να χρησιμοποιηθεί για τη διαχείριση των εικονικών δικτύων. Από το μενού του VM Manager επιλέγουμε *Edit->Connection details* και εμφανίζεται ένα καινούργιο παράθυρο. Από αυτό επιλέγουμε την καρτέλα *Virtual Networks* και εμφανίζεται η Εικόνα 9- 8.



Εικόνα 9- 8: Ρύθμιση εικονικών δικτύων

Από το παράθυρο της Εικόνα 9- 8 μπορούμε να δημιουργήσουμε, να διαγράψουμε ή να αλλάξουμε τις ρυθμίσεις τα εικονικά δίκτυα.

Η ρύθμιση των εικονικών δικτύων από το τερματικό γίνεται με τη χρήση της εντολής `virsh`. Υπάρχει μία ομάδα εντολών της `virsh` που έχει ρόλο τη διαχείριση των εικονικών δικτύων. Για παράδειγμα για να εμφανίσουμε τα εικονικά δίκτυα που υπάρχουν στο σύστημα μας εκτελούμε την παρακάτω εντολή

```
[root@ekdda ~]# virsh net-list --all
Name      State      Autostart  Persistent
-----
default   active     yes        yes
```

Βλέπουμε ότι υπάρχει ένα εικονικό δίκτυο στο σύστημα μας και αυτό είναι ενεργό και γίνεται ενεργό κάθε φορά που ξεκινάμε το σύστημα μας. Για κάθε ένα από τα δίκτυα που υπάρχουν στο σύστημα μας υπάρχει και ένα αρχείο xml που έχει τις ρυθμίσεις του που βρίσκεται στον κατάλογο `/etc/libvirt/qemu/networks`. Για να δημιουργήσουμε ένα καινούργιο εικονικό δίκτυο με το όνομα `students` δημιουργούμε ένα καινούργιο αρχείο `students.xml` με το παρακάτω περιεχόμενο

```
1 <network>
2   <name>students</name>
3   <forward mode='nat' />
4   <bridge name='virbr1'>
5     <ip address='192.168.222.1' netmask='255.255.255.0'>
6       <dhcp>
7         <range start='192.168.222.2' end='192.168.222.254' />
8       </dhcp>
9     </ip>
10  </network>
```

Στη συνέχεια για να ενημερώσουμε το KVM για αυτό το εικονικό δίκτυο εκτελούμε τις παρακάτω εντολές

```
[root@ekdda ~]# virsh net-define
/etc/libvirt/qemu/networks/students.xml
Network students defined from
/etc/libvirt/qemu/networks/students.xml

[root@ekdda ~]# virsh net-list --all
Name      State      Autostart  Persistent
-----
default   active     yes        yes
students  inactive   no         yes

[root@ekdda ~]# virsh net-start students
Network students started

[root@ekdda ~]# virsh net-autostart students
Network students marked as autostarted
```

Μία συχνή ρύθμιση είναι η απόδοση μιας στατικής ip σε μία εικονική μηχανή. Αυτό μπορεί να γίνει με τις παρακάτω ρυθμίσεις. Αρχικά βρίσκουμε την mac address της εικονικής μηχανής που θέλουμε να έχει στατική ip

```
[root@ekdda ~]# virsh dumpxml Ubuntu | grep 'mac address'
<mac address='52:54:00:2f:69:47' />
```

Στη συνέχεια αλλάζουμε τις ρυθμίσεις του εικονικού δικτύου στο οποίο περιέχεται αυτή η εικονική μηχανή όπως παρακάτω

```
[root@ekdda ~]# virsh net-edit students
```

Και τοποθετούμε τις παρακάτω ρυθμίσεις

```
<ip address='192.168.222.1' netmask='255.255.255.0'>
  <dhcp>
    <range start='192.168.222.2' end='192.168.222.254' />
    <host mac='52:54:00:2f:69:47' name='vm1' ip='192.168.222.200' />
  </dhcp>
</ip>
```

Στη συνέχεια σταματάμε και ξεκινάμε ξανά το εικονικό δίκτυο με τις παρακάτω εντολές

```
[root@ekdda ~]# virsh net-destroy students
Network students destroyed

[root@ekdda ~]# virsh net-start students
Network students started
```

Πολλές φορές χρειάζεται η επανεκκίνηση της εικονικής μηχανής για να εφαρμοστούν οι καινούργιες ρυθμίσεις.

## 9.5 Διαχείριση των εικονικών μηχανών

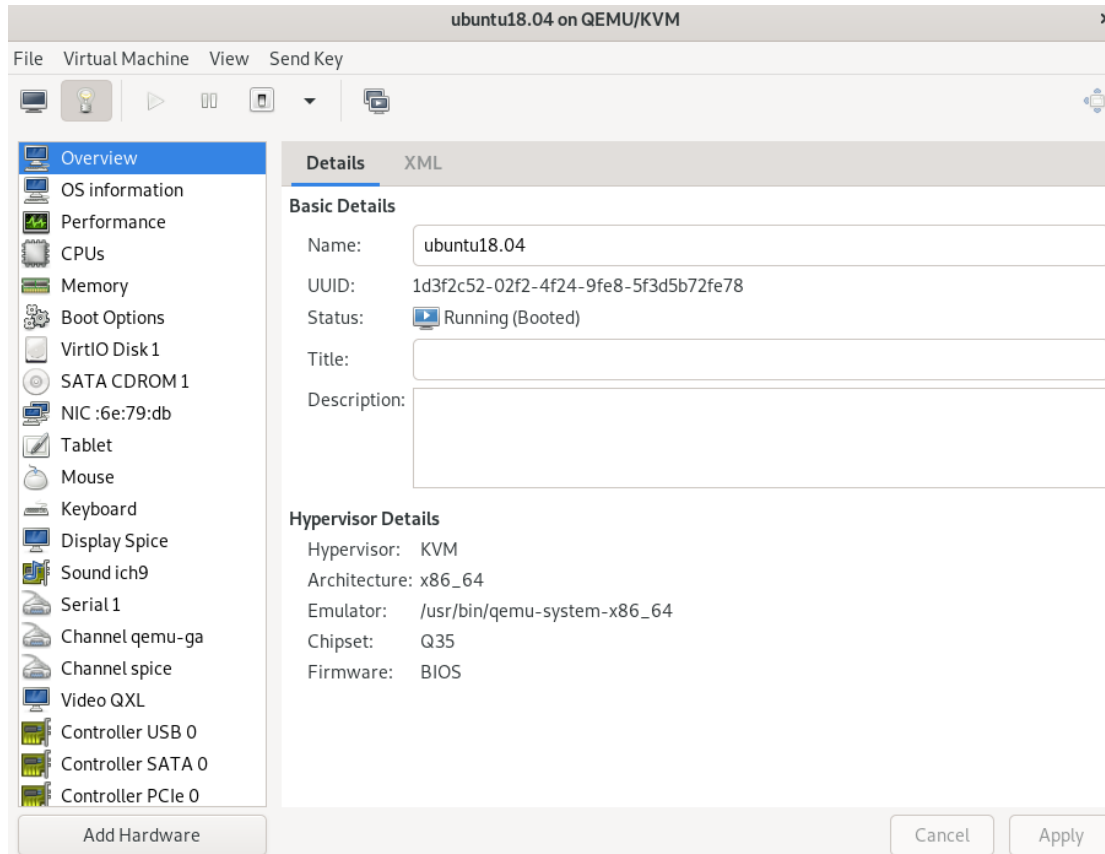
### 9.5.1 Διαχείριση εικονικής μηχανής από το γραφικό περιβάλλον

Η διαχείριση από το γραφικό περιβάλλον μιας εικονικής μηχανής μπορεί να γίνει με γραφικό τρόπο μέσα από το παράθυρο του VM Manager. Αν κάνουμε δεξί κλικ στο όνομα της εικονικής μηχανής που θέλουμε να διαχειριστούμε μας εμφανίζεται ένα μενού με επιλογές. Εκεί μπορούμε να κάνουμε κάποιες βασικές ρυθμίσεις της εικονικής μηχανής όπως να την ξεκινήσουμε, να την τερματίσουμε ή να την επανεκκινήσουμε. Από το ίδιο μενού μπορούμε να διαγράψουμε μια εικονική μηχανή ή και να δημιουργήσουμε ένα αντίγραφο της (Clone). Ιδιαίτερο ενδιαφέρον παρουσιάζει και η δυνατότητα που μας



παρέχεται να μεταφέρουμε την εικονική μηχανή που δημιουργήσαμε μέσω δικτύου σε έναν άλλο host που εκτελεί το KVM (RedHat-Virt, n.d.).

Επίσης μπορούμε να έχουμε πρόσβαση στις προδιαγραφές του υλικού για μια εικονική μηχανή από το παράθυρο του QEMU. Εκεί μπορούμε να κάνουμε κλικ στο *“Show virtual hardware details”* (εικονίδιο με λαμπτήρα) για να εμφανιστεί το παράθυρο της Εικόνα 9- 9.



Εικόνα 9- 9: Πληροφορίες υλικού εικονικής μηχανής

### 9.5.2 Διαχείριση εικονικής μηχανής από τη γραμμή εντολών

Η διαχείριση των εικονικών μηχανών από το τερματικό γίνεται με την εντολή **virsh**. Η εντολή αυτή παίρνει διάφορες παραμέτρους ανάλογα με τη λειτουργικότητα που θέλουμε να εκτελέσουμε. Για παράδειγμα μπορούμε να εμφανίσουμε όλες τις εικονικές μηχανές που υπάρχουν στο σύστημα μας. Αν παραλείψουμε το **-all** θα μας εμφανίσει μόνο αυτές τις εικονικές μηχανές που είναι ενεργές αυτή τη στιγμή.

```
[root@ekdda ~]# virsh list --all
 Id   Name           State
-----
  1   ubuntu18.04   running
```

Ενδιαφέρον παρουσιάζει το γεγονός ότι η κάθε εικονική μηχανή έχει ένα όνομα (name). Οι εικονικές μηχανές που εκτελούνται έχουν και ένα αναγνωριστικό (id). Αυτά χρησιμοποιούνται από τις παρακάτω παραμέτρους για να καθορίσουμε σε ποια εικονική

μηχανή θέλουμε να εκτελέσουμε μία ενέργεια. Για να τερματίσουμε μια εικονική μηχανή μπορούμε να χρησιμοποιήσουμε την παράμετρο `shutdown` ή την `destroy` αν θέλουμε να εξομοιώσουμε τη διακοπή ρεύματος σε έναν κανονικό υπολογιστή.

```
[root@ekdda ~]# virsh shutdown 1
Domain 1 is being shutdown
```

Ενώ για να ξεκινήσουμε μια εικονική μηχανή μπορούμε να χρησιμοποιήσουμε την παράμετρο `start`. Ιδιαίτερο ενδιαφέρον παρουσιάζει και η παράμετρος `autostart` που επιβάλλει σε μία εικονική μηχανή να ξεκινά αυτόματα κάθε φορά που θα ξεκινά και το πραγματικό σύστημα.

```
[root@ekdda ~]# virsh start ubuntu18.04
Domain ubuntu18.04 started
```

Για να διαγράψουμε μια εικονική μηχανή που την έχουμε δημιουργήσει θα πρέπει να φροντίσουμε αρχικά αυτή να μην λειτουργεί και στη συνέχεια να εκτελέσουμε την εντολή `virsh` με την παράμετρο `undefine`. Οι πρόσθετες παράμετροι ορίζουν τη διαγραφή του αρχείου που λειτουργεί σαν αποθηκευτικός χώρος της εικονικής μηχανής.

```
[root@ekdda ~]# virsh undefine ubuntu18.04 --remove-all-storage
```

Το σύνολο των χαρακτηριστικών μιας εικονικής μηχανής υπάρχουν σε ένα `xml` αρχείο που περιέχεται στον κατάλογο `/etc/libvirt/qemu/«όνομα εικονικής μηχανής»`. Μπορούμε να δούμε το αρχείο αυτό με την εντολή `virsh dumpxml «όνομα εικονικής μηχανής»`. Η εντολή `virsh` έχει ένα πολύ μεγάλο πλήθος εντολών που μπορούν να ρυθμίσουν τα χαρακτηριστικά μιας εικονικής μηχανής. Για μια αναλυτική παράθεσή τους μπορούμε να εκτελέσουμε την εντολή `virsh --help`.

## Παραπομπές

- Aboukhalil, R. (2018, May 31). *You don't know Bash: An introduction to Bash arrays*. Ανάκτηση από Open Source: <https://opensource.com/article/18/5/you-dont-know-bash-intro-bash-arrays>
- Andrew S Tanenbaum, D. W. (1996). *Computer networks*. Prentice hall.
- Apache\_org. (n.d.). *Apache.org*. Retrieved December 15, 2020, from Apache.org: <https://httpd.apache.org/docs/current/configuring.html>
- BashHackersWiki. (n.d.). *Handling positional parameters*. Retrieved December 15, 2020, from Bash Hackers Wiki: <https://wiki.bash-hackers.org/scripting/posparams>
- Both, D. (2017, March 16). *An introduction to GRUB2 configuration for your Linux machine*. Ανάκτηση από Open Source: <https://opensource.com/article/17/3/introduction-grub2-configuration-linux>
- Budzar, K. (2020, October 19). *How to use Variables in Bash*. Ανάκτηση από Linux Hint: [https://linuxhint.com/variables\\_bash/](https://linuxhint.com/variables_bash/)
- Carigan, T. (2020, October 15). *Linux permissions: SUID, SGID, and sticky bit*. Ανάκτηση από Enable Sysadmin - Red Hat: <https://www.redhat.com/sysadmin/suid-sgid-sticky-bit>
- Carrigan, T. (2020, June 19). *An introduction to the Linux /etc/fstab file*. Ανάκτηση από Enable Sysadmin - Red Hat: <https://www.redhat.com/sysadmin/etc-fstab>
- CISCO\_IP\_Networking. (χ.χ.). *CISCO\_IP\_Networking*. Ανάκτηση 16, 2021, από CISCO\_IP\_Networking: <https://www.cisco.com/en/US/docs/security/vpn5000/manager/reference/guide/ppA.html>
- CISCO\_SNMP. (n.d.). *CISCO\_SNMP*. Retrieved December 21, 2020, from CISCO\_SNMP: [https://www.cisco.com/c/en/us/td/docs/security/ips/6-1/configuration/guide/cli/cliguide/cli\\_snmp.pdf](https://www.cisco.com/c/en/us/td/docs/security/ips/6-1/configuration/guide/cli/cliguide/cli_snmp.pdf)
- Clausen, A., Kreuter, R. M., & Polzer, L. P. (2019). *GNU Parted User Manual*. Free Software Foundation.
- Common\_Ports\_MIT. (n.d.). *Common\_Ports*. Retrieved December 14, 2020, from MIT\_Common Ports: <https://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-sg-en-4/ch-ports.html>
- CONTA\_Lab. (n.d.). *εργαστήριο CONTA (COmputer Networks & Telematics Applications)*. Retrieved December 18, 2020, from εργαστήριο CONTA (COmputer Networks & Telematics Applications): [http://conta.uom.gr/conta/ekpaideysh/metaptyxiaka/technologies\\_diktywn/teaching\\_m/management/snmp.htm](http://conta.uom.gr/conta/ekpaideysh/metaptyxiaka/technologies_diktywn/teaching_m/management/snmp.htm)

- Cooper, M. (2014, March 10). *Advanced Bash-Scripting Guide*. Ανάκτηση από The Linux Programming Interface: <https://tldp.org/LDP/abs/html/>
- DNF. (n.d.). *DNF Command Reference*. Retrieved December 15, 2020, from DNF: [https://dnf.readthedocs.io/en/latest/command\\_ref.html](https://dnf.readthedocs.io/en/latest/command_ref.html)
- Docile, E. (2020, May 28). *How dd command works in Linux with examples*. Ανάκτηση από Linux Config: <https://linuxconfig.org/how-dd-command-works-in-linux-with-examples>
- FedoraDocsBasic-Recovery. (n.d.). *Basic System Recovery*. Retrieved November 21, 2020, from Fedora Docs: [https://docs.fedoraproject.org/en-US/Fedora/14/html/Installation\\_Guide/ap-rescuemode.html](https://docs.fedoraproject.org/en-US/Fedora/14/html/Installation_Guide/ap-rescuemode.html)
- FedoraDocs-Boot. (n.d.). *Boot Process, Init, and Shutdown*. Retrieved November 21, 2020, from Fedora Docs: [https://docs.fedoraproject.org/en-US/Fedora/18/html/Installation\\_Guide/ch-boot-init-shutdown.html](https://docs.fedoraproject.org/en-US/Fedora/18/html/Installation_Guide/ch-boot-init-shutdown.html)
- FedoraDocs-FileSystem. (n.d.). *Overview of File System Hierarchy Standard (FHS)*. Retrieved December 15, 2020, from Fedora Documentation: [https://docs.fedoraproject.org/en-US/Fedora/14/html/Storage\\_Administration\\_Guide/s1-filesystem-fhs.html](https://docs.fedoraproject.org/en-US/Fedora/14/html/Storage_Administration_Guide/s1-filesystem-fhs.html)
- FedoraDocsInitProcess. (n.d.). *FedoraDocsInitProcess*. Retrieved January 5, 2021, from FedoraDocsInitProcess: [https://docs.fedoraproject.org/en-US/Fedora/18/html/Installation\\_Guide/s2-boot-init-shutdown-systemd.html](https://docs.fedoraproject.org/en-US/Fedora/18/html/Installation_Guide/s2-boot-init-shutdown-systemd.html)
- FedoraDocs-Install. (n.d.). *Installation Guide*. Retrieved November 20, 2020, from Fedora Docs: <https://docs.fedoraproject.org/en-US/fedora/rawhide/install-guide/>
- FedoraDocs-InstImage. (n.d.). *Creating and using a live installation image*. Retrieved November 20, 2020, from Fedora Documentaion: <https://docs.fedoraproject.org/en-US/quick-docs/creating-and-using-a-live-installation-image/>
- FedoraDocsNamingInterface. (n.d.). *FedoraDocsNamingInterface*. Retrieved January 16, 2021, from FedoraDocsNamingInterface: [https://docs.fedoraproject.org/en-US/Fedora/22/html/Networking\\_Guide/ch-Consistent\\_Network\\_Device\\_Naming.html](https://docs.fedoraproject.org/en-US/Fedora/22/html/Networking_Guide/ch-Consistent_Network_Device_Naming.html)
- FedoraDocsProcesses. (n.d.). *FedoraDocsProcesses*. Retrieved December 21, 2020, from FedoraDocsProcesses: [https://docs.fedoraproject.org/en-US/Fedora/24/html/System\\_Administrators\\_Guide/ch-System\\_Monitoring\\_Tools.html#s1-sysinfo-system-processes](https://docs.fedoraproject.org/en-US/Fedora/24/html/System_Administrators_Guide/ch-System_Monitoring_Tools.html#s1-sysinfo-system-processes)
- FedoraDocs-ShadowPasswords. (n.d.). *Shadow Passwords*. Retrieved December 15, 2015, from Fedora Docs: [https://docs.fedoraproject.org/en-US/Fedora/15/html/Deployment\\_Guide/s1-users-groups-shadow-utilities.html](https://docs.fedoraproject.org/en-US/Fedora/15/html/Deployment_Guide/s1-users-groups-shadow-utilities.html)

- FedoraDocs-Sudo. (n.d.). *Performing administration tasks using sudo*. Retrieved December 15, 2020, from Fedora Documentation: <https://docs.fedoraproject.org/en-US/quick-docs/performing-administration-tasks-using-sudo/>
- FedoraDocsSystemdProcess. (n.d.). *FedoraDocsSystemdProcess*. Retrieved January 2, 2021, from FedoraDocsSystemdProcess: <https://docs.fedoraproject.org/en-US/quick-docs/understanding-and-administering-systemd/index.html>
- FedoraDocs-Tar. (n.d.). *Archiving Files with tar*. Retrieved December 15, 2020, from Fedora Documentation: [https://docs.fedoraproject.org/en-US/Fedora/12/html/Security-Enhanced\\_Linux/sect-Security-Enhanced\\_Linux-Maintaining\\_SELinux\\_Labels\\_-\\_Archiving\\_Files\\_with\\_tar.html](https://docs.fedoraproject.org/en-US/Fedora/12/html/Security-Enhanced_Linux/sect-Security-Enhanced_Linux-Maintaining_SELinux_Labels_-_Archiving_Files_with_tar.html)
- FedoraDocsTty. (n.d.). *FedoraDocsTty*. Retrieved January 16, 2021, from FedoraDocsTty: [https://docs.fedoraproject.org/en-US/fedora/rawhide/system-administrators-guide/basic-system-configuration/Opening\\_GUI\\_Applications/](https://docs.fedoraproject.org/en-US/fedora/rawhide/system-administrators-guide/basic-system-configuration/Opening_GUI_Applications/)
- FedoraDocs-Virtualization. (n.d.). *Enabling Intel VT and AMD-V virtualization hardware extensions in BIOS*. Retrieved December 15, 2020, from Fedora Docs: [https://docs.fedoraproject.org/en-US/Fedora/13/html/Virtualization\\_Guide/sect-Virtualization-Troubleshooting-Enabling\\_Intel\\_VT\\_and\\_AMD\\_V\\_virtualization\\_hardware\\_extensions\\_in\\_BIOS.html](https://docs.fedoraproject.org/en-US/Fedora/13/html/Virtualization_Guide/sect-Virtualization-Troubleshooting-Enabling_Intel_VT_and_AMD_V_virtualization_hardware_extensions_in_BIOS.html)
- FedoraPackages. (n.d.). *Fedora Packages*. Retrieved December 15, 2020, from Fedora Packages Sources: <https://src.fedoraproject.org/>
- FedoraWiki. (2020, June 5). *DNF*. Ανάκτηση από Fedora Wiki: <https://fedoraproject.org/wiki/DNF>
- FedoraWiki. (n.d.). *Samba Server*. Retrieved November 22, 2020, from Fedora Wiki: [https://fedoraproject.org/wiki/Administration\\_Guide\\_Draft/Samba](https://fedoraproject.org/wiki/Administration_Guide_Draft/Samba)
- Fellows, R., & Crocetti, P. (2020, August 18). *Types of Backup Explained*. Ανάκτηση από Search Data Backup: <https://searchdatabackup.techtarget.com/feature/Full-incremental-or-differential-How-to-choose-the-correct-backup-type>
- Freebsd. (n.d.). *Xorg Configuration*. Retrieved November 29, 2020, from Xorg Configuration: <https://www.freebsd.org/doc/handbook/x-config.html>
- Friedl, J. E. (2006). *Mastering Regular Expressions*. O'Reilly Media, Inc.
- Garrels, B. (2008, December 27). *The while loop*. Ανάκτηση από Bash Guide for Beginners: [https://tldp.org/LDP/Bash-Beginners-Guide/html/sect\\_09\\_02.html](https://tldp.org/LDP/Bash-Beginners-Guide/html/sect_09_02.html)
- geeksforggeeks\_connection-less-service. (χ.χ.). *geeksforggeeks\_connection-less-service*. Ανάκτηση 14, 2021, από geeksforggeeks\_connection-less-service: <https://www.geeksforggeeks.org/connection-less-service/>

- geeksforgEEKS\_connection-oriented-service. (χ.χ.). *geeksforgEEKS\_connection-oriented-service*. Ανάκτηση 14, 2021, από geeksforgEEKS\_connection-oriented-service: <https://www.geeksforgEEKS.org/connection-oriented-service/>
- GNU\_GPL. (n.d.). *GNU GPL Licences*. Retrieved December 20, 2020, from GNU GPL Licences: <https://www.gnu.org/licenses/gpl-3.0.html>
- Hess, K. (2019, December 10). *Linux sysadmin basics: User account management with UIDs and GIDs*. Ανάκτηση από Red Hat : <https://www.redhat.com/sysadmin/user-account-gid-uid>
- IAS. (2010, January 28). *The How and Why of User Private Groups in Unix*. Ανάκτηση από Institute for Advanced Study Network Security: <https://security.ias.edu/how-and-why-user-private-groups-unix>
- IBM. (2019, June 19). *Virtualization*. Ανάκτηση από IBM Cloud Education: <https://www.ibm.com/cloud/learn/virtualization-a-complete-guide>
- IETF\_Internet\_Engineering\_Task\_Force. (n.d.). *IETF\_Internet\_Engineering\_Task\_Force*. Retrieved January 7, 2021, from IETF\_Internet\_Engineering\_Task\_Force: <https://www.ietf.org/>
- James, A. (2012, November 25). *Conditions in bash scripting (if statements)*. Ανάκτηση από A Cloud Guru: <https://acloudguru.com/blog/engineering/conditions-in-bash-scripting-if-statements>
- Jayden, F. (2018, February 6). *Advantages of Linux Dedicated Server*. Ανάκτηση από Medium: <https://medium.com/@jaydenfernando89/advantages-of-linux-dedicated-server-b3fabb4ee15f>
- Jevtic, G. (2020, March 23). *Rsync Command: 20 Helpful Examples In Linux*. Ανάκτηση από PhoenixNAP: <https://phoenixnap.com/kb/rsync-command-linux-examples>
- Kili, A. (2019, March 13). *Why Linux System Administrators Need Programming Skills*. Ανάκτηση από Techmint: <https://www.techmint.com/why-linux-system-administrators-need-programming-skills/>
- KVM. (n.d.). *KVM Documentation*. Retrieved December 15, 2020, from Linux KVM: <https://www.linux-kvm.org/page/Documents>
- LandOfLinux. (n.d.). *Read Command*. Retrieved December 15, 2020, from Land of Linux: [https://landoflinux.com/linux\\_bash\\_scripting\\_read.html](https://landoflinux.com/linux_bash_scripting_read.html)
- LibVirt. (n.d.). *LibVirt Virtualization API*. Retrieved December 15, 2020, from LibVirt Virtualization API: <https://libvirt.org>
- Linuxize-Chown. (2019, September 6). *Chown Command in Linux (File Ownership)*. Ανάκτηση από Linuxize: <https://linuxize.com/post/linux-chown-command/>

- Linuxize-Functions. (2020, May 30). *Bash Functions*. Ανάκτηση από Linuxize: <https://linuxize.com/post/bash-functions/>
- LinuxOrgKillSignals. (n.d.). *LinuxOrgKillSignals*. Retrieved January 16, 2021, from LinuxOrgKillSignals: <https://www.linux.org/threads/kill-commands-and-signals.8881/>
- Lonston, B. (2018, October 2). *Create and mount filesystems in Linux*. Ανάκτηση από Linux SysAdmins: <https://www.linuxsysadmins.com/create-and-mount-filesystems-in-linux/>
- ManPage-Virt. (n.d.). *virt-install(1) - Linux man page*. Retrieved December 15, 2020, from Linux Die: <https://linux.die.net/man/1/virt-install>
- Moon, S. (2020, August 14). *5 Linux Commands to Shutdown and Reboot the System*. Ανάκτηση από Binary Tides: <https://www.binarytides.com/linux-command-shutdown-reboot-restart-system/>
- Morelo, D. (2020, January 1). *Best Linux Distributions for Servers*. Ανάκτηση από Linuxhint.com: [https://linuxhint.com/best\\_linux\\_distribution\\_servers/](https://linuxhint.com/best_linux_distribution_servers/)
- Muntaha, S. (2020, January 1). *Linux File Compression Options and Comparison*. Ανάκτηση από Linuxhint: [https://linuxhint.com/linux\\_file\\_compression/](https://linuxhint.com/linux_file_compression/)
- Nemeth, E., Snyder, G., Hein, T. R., & Whaley, B. (2017). *Unix and Linux System Administration Handbook*. Addison Wesley.
- Newham, C., & Rosenblett, B. (2005). Command Substitution. Στο *Learning the bash shell: Unix shell programming* (σ. 354). O'Reilly Media.
- OpenSource. (n.d.). *What is Bash?* Retrieved December 15, 2020, from Open Source-Bash: <https://opensource.com/resources/what-bash>
- QEMU. (n.d.). *QEMU*. Retrieved December 15, 2020, from QEMU the FAST! processor emulator: [qemu.org](http://qemu.org)
- RedHat-LVM. (n.d.). *CHAPTER 1. THE LVM LOGICAL VOLUME MANAGER*. Retrieved December 20, 2020, from Red Hat Documentation: [https://access.redhat.com/documentation/en-us/red\\_hat\\_enterprise\\_linux/4/html/cluster\\_logical\\_volume\\_manager/lvm\\_overview](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/4/html/cluster_logical_volume_manager/lvm_overview)
- RedhatOrphanZombiProcess. (n.d.). *RedhatOrphanZombiProcess*. Retrieved January 2, 2021, from RedhatOrphanZombiProcess: <https://www.redhat.com/sysadmin/killing-zombies-linux-style>
- RedHat-Virt. (n.d.). *15.6. Migrating with Virt-Manager*. Retrieved December 15, 2020, from Red Hat Topics: <https://access.redhat.com/documentation/en->

us/red\_hat\_enterprise\_linux/7/html/virtualization\_deployment\_and\_administration\_guide/sect-kvm\_live\_migration-migrating\_with\_virt\_manager

RedHat-Virtualization. (n.d.). *What is Virtualization*. Retrieved December 15, 2020, from Red Hat Topics: <https://www.redhat.com/en/topics/virtualization>

Salter, J. (2018, April 02). *Understanding Linux filesystems: ext4 and beyond*. Ανάκτηση από Open Source: <https://opensource.com/article/18/4/ext4-file-system>

SambaOrgSmbConf. (n.d.). *SambaOrgSmbConf*. Retrieved January 16, 2021, from SambaOrgSmbConf: <https://www.samba.org/samba/docs/current/man-html/smb.conf.5.html>

Sambaproject. (n.d.). *Official Web Site*. Retrieved November 22, 2020, from samba org: <https://www.samba.org/>

SelinuxBooleans. (n.d.). *SelinuxBooleans*. Retrieved December 18, 2020, from SelinuxBooleans: <https://wiki.centos.org/TipsAndTricks/SelinuxBooleans>

ServerWorld. (n.d.). *ServerWorldFedora31*. Retrieved November 22, 2020, from ServerWorldFedora31: [https://www.server-world.info/en/note?os=Fedora\\_31&p=download](https://www.server-world.info/en/note?os=Fedora_31&p=download)

Sheer, P. (2019, December 15). *User Accounts and User Ownerships*. Ανάκτηση από Rute User's Tutorial and Exposition: <https://linuxcourse.rutgers.edu/rute/node14.html>

Son, B. (2020, August 25). *6 Open Source Virtualization Technologies to know in 2020*. Ανάκτηση από Open Source: <https://opensource.com/article/20/8/virt-tools>

TechMint. (2017, January 27). *The Story Behind 'init' and 'systemd': Why 'init' Needed to be Replaced with 'systemd' in Linux*. Retrieved from TechMint.

TigerVNC. (n.d.). *TigerVNC*. Retrieved December 21, 2020, from TigerVNC: <https://tigervnc.org/>

Warfield, C. (2018, April 23). *How to reset a root password on Fedora*. Ανάκτηση από Fedora Magazine: <https://fedoramagazine.org/reset-root-password-fedora/>

Webmin\_releases. (n.d.). *Webmin\_releases*. Retrieved December 20, 2020, from Webmin\_releases: <https://sourceforge.net/projects/webadmin/files/webmin/>

Wiki\_X\_Window\_Managers. (n.d.). *Wiki\_X\_Window\_Managers*. Retrieved November 29, 2020, from Wiki\_X\_Window\_Managers: [https://en.wikipedia.org/wiki/Comparison\\_of\\_X\\_window\\_managers](https://en.wikipedia.org/wiki/Comparison_of_X_window_managers)

WikiIPProtocol. (n.d.). *WikiIPProtocol*. Retrieved January 9, 2020, from WikiIPProtocol: [https://en.wikipedia.org/wiki/Internet\\_Protocol](https://en.wikipedia.org/wiki/Internet_Protocol)

WikiMetasploit. (n.d.). *WikiMetasploit*. Retrieved January 10, 2021, from WikiMetasploit: [https://en.wikipedia.org/wiki/Metasploit\\_Project](https://en.wikipedia.org/wiki/Metasploit_Project)



Wikipedia:Client-Server. (n.d.). *Client–server model*. Retrieved December 10, 2020, from Wikipedia: [https://en.wikipedia.org/wiki/Client%E2%80%93server\\_model](https://en.wikipedia.org/wiki/Client%E2%80%93server_model)

Wikipedia\_OSI\_model. (n.d.). *Wikipedia\_OSI\_model*. Retrieved December 15, 2020, from Wikipedia\_OSI\_model: [https://el.wikipedia.org/wiki/%CE%9C%CE%BF%CE%BD%CF%84%CE%AD%CE%BB%CE%BF\\_%CE%B1%CE%BD%CE%B1%CF%86%CE%BF%CF%81%CE%AC%CF%82\\_OSI](https://el.wikipedia.org/wiki/%CE%9C%CE%BF%CE%BD%CF%84%CE%AD%CE%BB%CE%BF_%CE%B1%CE%BD%CE%B1%CF%86%CE%BF%CF%81%CE%AC%CF%82_OSI)

Wikipedia-HardDisk. (n.d.). *Hard disk drive interface*. Retrieved December 20, 2020, from Wikipedia: [https://en.wikipedia.org/wiki/Hard\\_disk\\_drive\\_interface](https://en.wikipedia.org/wiki/Hard_disk_drive_interface)

WikiRunlevels. (n.d.). *WikiRunlevels*. Retrieved November 29, 2020, from WikiRunlevels: <https://en.wikipedia.org/wiki/Runlevel>

Κωνσταντοπούλου Μ., Ξ. Ν. (2015). *ΔΙΚΤΥΑ ΥΠΟΛΟΓΙΣΤΩΝ - Γ' Τάξη ΤΟΜΕΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΕΠΑ.Λ.* Πάτρα: ΙΝΣΤΙΤΟΥΤΟ ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ & ΕΚΔΟΣΕΩΝ ΔΙΟΦΑΝΤΟΣ.