ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ ΕΣΩΤΕΡΙΚΩΝ

εκδδα
ΕΘΝΙΚΟ ΚΕΝΤΡΟ ΔΗΜΟΣΙΑΣ ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΑΥΤΟΔΙΟΙΚΗΣΗΣ

ΥΠΟΕΡΓΟ: ΥΠΟΕΡΓΟ 3 «ΔΡΑΣΕΙΣ ΒΕΛΤΙΩΣΗΣ ΠΟΙΟΤΗΤΑΣ ΕΠΙΜΟΡΦΩΤΙΚΩΝ ΠΡΟΓΡΑΜΜΑΤΩΝ 2022-2023»

της Πράξης «ΔΡΑΣΕΙΣ ΣΥΝΕΧΙΖΟΜΕΝΗΣ ΚΑΤΑΡΤΙΣΗΣ 2022-2023 (Β΄ ΦΑΣΗ ΔΡΑΣΕΩΝ ΚΑΤΑΡΤΙΣΗΣ)»
κωδ. ΟΠΣ

ΤΙΤΛΟΣ ΠΡΟΓΡΑΜΜΑΤΟΣ:

ΕΞ ΑΠΟΣΤΑΣΕΩΣ ΕΚΠΑΙΔΕΥΣΗ ΣΤΗ ΔΙΑΧΕΙΡΙΣΗ ΣΥΣΤΗΜΑΤΩΝ LINUX (ΜΕΡΟΣ Α)

ΕΚΠΑΙΔΕΥΤΙΚΟ ΥΛΙΚΟ

Κωδικός εκπαιδευτικού υλικού:

Κωδικός Πιστοποίησης προγράμματος: 747

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Διαρθρωτικό και Επενδυτικό Ταμείο

Ε.Π. ΜΕΤΑΡΡΥΘΜΙΣΗ ΔΗΜΟΣΙΟΥ ΤΟΜΕΑ

ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

ΥΠΟΕΡΓΟ: ΥΠΟΕΡΓΟ 3 «ΔΡΑΣΕΙΣ ΒΕΛΤΙΩΣΗΣ ΠΟΙΟΤΗΤΑΣ ΕΠΙΜΟΡΦΩΤΙΚΩΝ ΠΡΟΓΡΑΜΜΑΤΩΝ 2022-2023»

ΤΙΤΛΟΣ ΠΡΟΓΡΑΜΜΑΤΟΣ:

ΕΞ ΑΠΟΣΤΑΣΕΩΣ ΕΚΠΑΙΔΕΥΣΗ ΣΤΗ ΔΙΑΧΕΙΡΙΣΗ ΣΥΣΤΗΜΑΤΩΝ LINUX (ΜΕΡΟΣ Α)

ΟΜΑΔΑ ΕΡΓΑΣΙΑΣ

Μέλη Ομάδας

Συντονιστής/στρια:
Ιωάννης Ματζαβάκης

Συγγραφείς:
Δρ. Γεώργιος Παπαμιχαήλ
Αλέξανδρος Γιοχάλας
Παναγιώτης Παπαϊωάννου

Αξιολογητές/τριες:

Δρ. Γεώργιος Μαυρομμάτης
Δρ. Αναστάσιος Σαλής

Ευρωπαϊκή Ένωση
Ευρωπαϊκά Διαρθρωτικά και Επενδυτικά Ταμεία

Ε.Π. ΜΕΤΑΡΡΥΘΜΙΣΗ ΔΗΜΟΣΙΟΥ ΤΟΜΕΑ

ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Περιεχόμενα

1. Εγκατάσταση του λειτουργικού συστήματος - Εργασία στη γραμμή εντολών

Μαθησιακοί στόχοι

Οι επιμορφούμενοι θα πρέπει να είναι σε θέση να:

- προετοιμάζουν μια εγκατάσταση του λειτουργικού συστήματος Linux
- πραγματοποιούν μια «χειροκίνητη» εγκατάσταση του λειτουργικού συστήματος Linux
- εισέρχονται σε ένα λειτουργικό σύστημα Linux και να εκτελούν απλές εντολές

1.1 Προετοιμασία εγκατάστασης του λειτουργικού συστήματος

Για τις ανάγκες αυτού του επιμορφωτικού προγράμματος θα χρησιμοποιηθεί το λειτουργικό σύστημα Rocky Linux (στην έκδοση 9.2). Το Rocky Linux είναι μια ανοικτού κώδικα διανομή Linux που αναπτύσσεται από το Rocky Enterprise Software Foundation. Είναι ένα εταιρικό λειτουργικό σύστημα που σχεδιάστηκε να είναι ένα απευθείας παράγωγο, με πλήρη συμβατότητα, που χρησιμοποιεί τον πηγαίο κώδικα του Red Hat Enterprise Linux OS. Ο στόχος του είναι η απόλυτη συμβατότητα προς το οικοσύστημα της διανομής Red Hat.

1.1.1 Προαπαιτούμενα

Οι προτεινόμενες απαιτήσεις συστήματος είναι οι παρακάτω:

- **Αρχιτεκτονική:** x86-64, ARM64, ppc64le ή s390x.
- **Μνήμη RAM:** τουλάχιστον 2 GB.
- **Αποθηκευτικός χώρος:** 40 GB χώρος δίσκου.
- **Μέγεθος λειτουργικού συστήματος:** Η εικόνα ISO έχει μέγεθος 8.8 GB

1.1.2 Δημιουργία ενός ιδεατού περιβάλλοντος

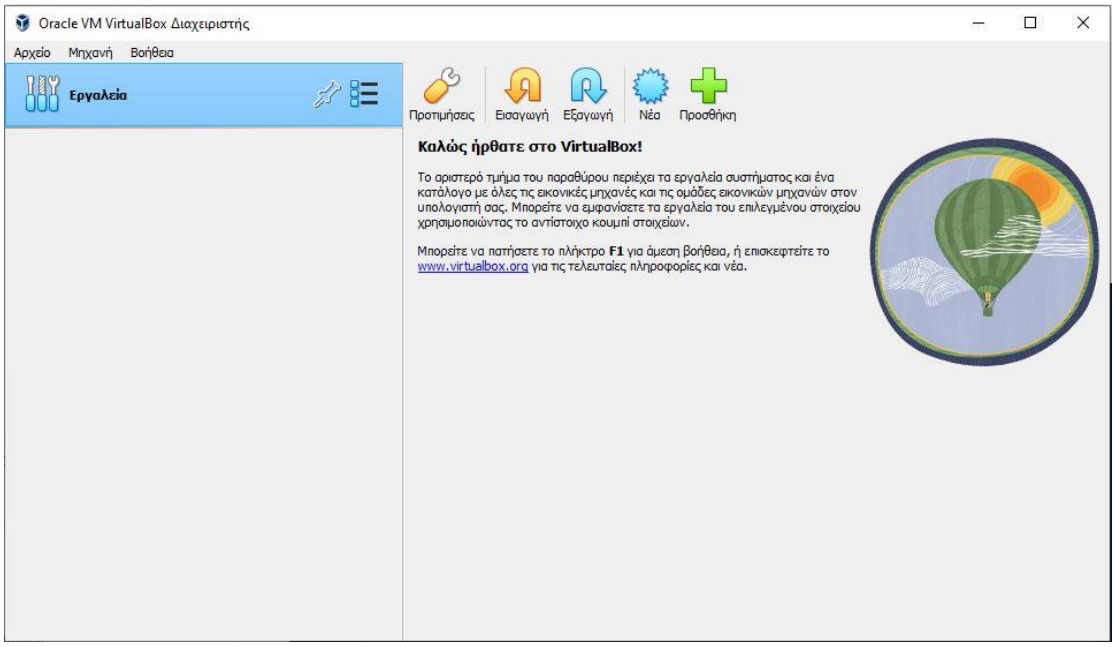
Η δημιουργία ενός ιδεατού περιβάλλοντος για την εγκατάσταση και λειτουργία μιας ή περισσότερων Linux μηχανών είναι ένας αποτελεσματικός και αποδοτικός τρόπος, αφού επιτρέπει τη συνύπαρξη διαφορετικών διανομών. Έτσι μπορεί οποιοσδήποτε να συγκρίνει χαρακτηριστικά, να μελετήσει διαφορετικές διανομές και να εξερευνήσει δικτυακές εφαρμογές μεταβαίνοντας από το ένα σύστημα στο άλλο.

Για τις ανάγκες αυτού του επιμορφωτικού προγράμματος θα χρησιμοποιήσουμε ένα δωρεάν περιβάλλον ανάπτυξης ιδεατών μηχανών (hypervisor), το *Oracle VirtualBox* (εναλλακτικά μπορείτε να χρησιμοποιήσετε το VMWare Workstation Player ή το Microsoft Hyper-V).

Το Oracle VirtualBox είναι μια ισχυρή και ευέλικτη λύση δημιουργίας και διαχείρισης εικονικών μηχανών. Με το VirtualBox, μπορείτε να δημιουργήσετε εικονικές μηχανές που εκτελούν διάφορα λειτουργικά συστήματα, όπως Windows, Linux, macOS και πολλά άλλα, σε ένα μόνο φυσικό σύστημα.

Αυτή η ευέλικτη εικονική πλατφόρμα προσφέρει πλούσια χαρακτηριστικά, συμπεριλαμβανομένης της δυνατότητας κοινής χρήσης αρχείων μεταξύ του φυσικού και του εικονικού συστήματος, της ενσωματωμένης δυνατότητας δημιουργίας δικτύων, και της υποστήριξης γρήγορης επιτάχυνσης 3D για εικονικά μηχανήματα που υποστηρίζουν γραφικά.

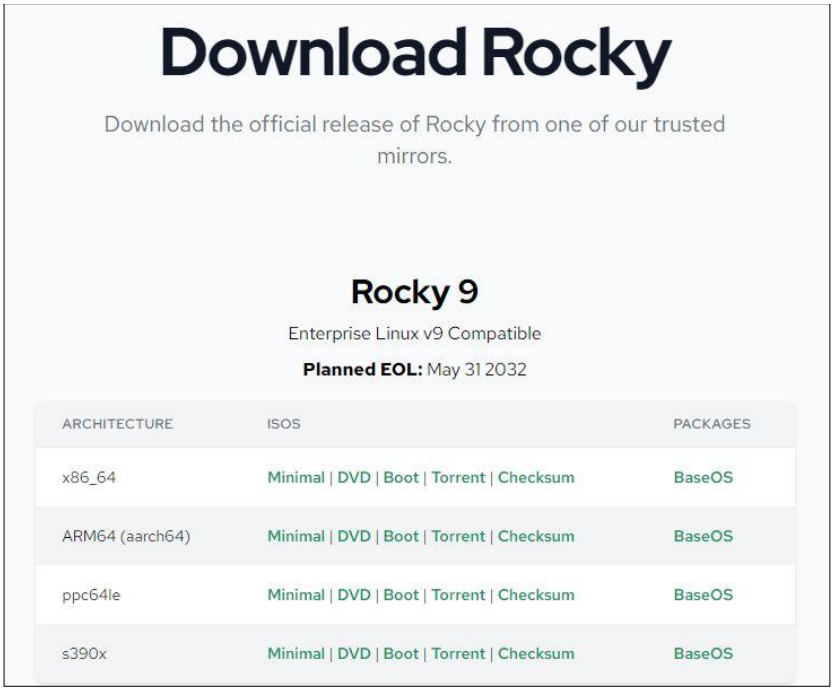
Έτσι, ξεκινώντας μπορείτε να κατευθυνθείτε στην ηλεκτρονική τοποθεσία https://www.virtualbox.org/wiki/Downloads, να κατεβάσετε την έκδοση που ταιριάζει στο λειτουργικό σας σύστημα και να την εγκαταστήσετε. Στο τέλος της εγκατάστασης θα εμφανιστεί η οθόνη του διαχειριστή των ιδεατών μηχανών, όπως παρακάτω:



Εικόνα 1 Oracle VirtualBox (Διαχειριστής)

1.1.3 Λήψη του λειτουργικού συστήματος

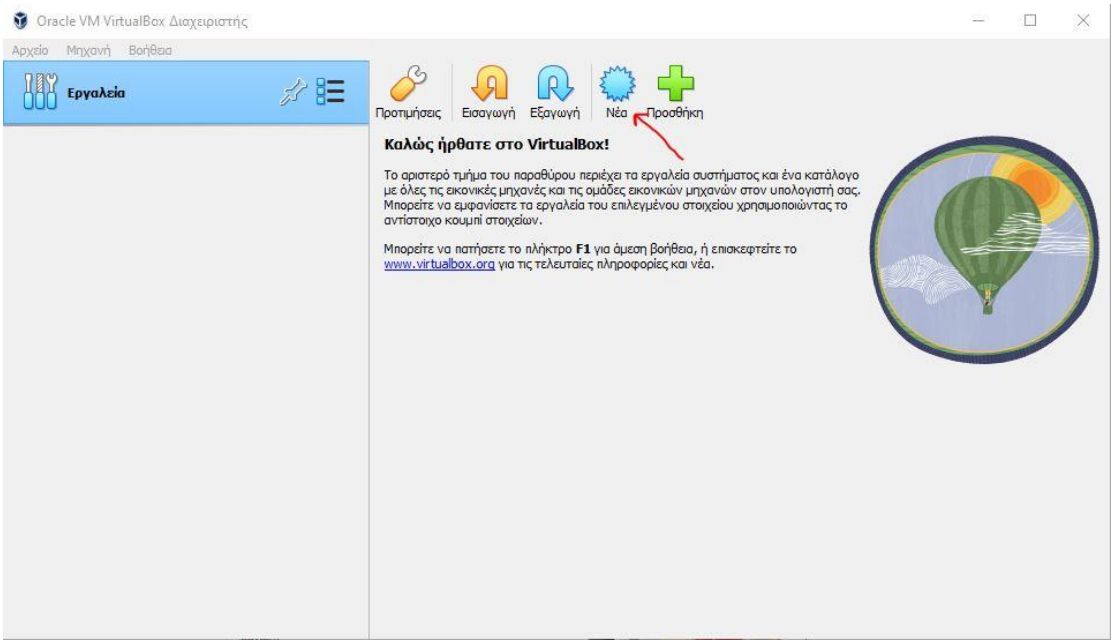
Χρησιμοποιώντας ένα πρόγραμμα πλοήγησης στο Διαδίκτυο, μπορείτε να κατευθυνθείτε στην ηλεκτρονική τοποθεσία https://rockylinux.org/download/, και να "κατεβάσετε" την έκδοση που ταιριάζει καλύτερα στο σύστημα σας από τα διαθέσιμα αρχεία ISO.



Εικόνα 2 Κατεβάζοντας το Rocky Linux (Έκδοση 9)

Σημείωση: Για την περίπτωση εγκατάστασης σε σύστημα με Windows 64-bit σας προτείνουμε να επιλέξετε την επιλογή **DVD** για την αρχιτεκτονική **X86_64**.

Στη συνέχεια στο πρόγραμμα VirtualBox επιλέγουμε **Νέα** για να δημιουργήσουμε μια νέα εικονική μηχανή.



Εικόνα 3 Δημιουργία νέας εικονικής μηχανής

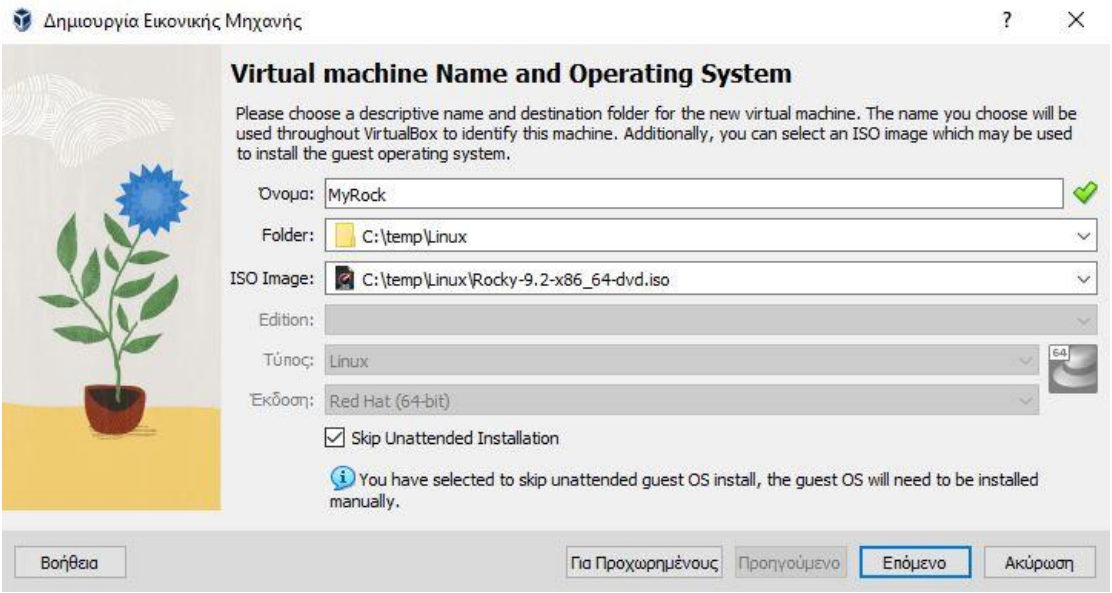
Επιλέγουμε τον τύπο του λειτουργικού συστήματος δίνοντας τις παρακάτω ρυθμίσεις:

Όνομα: Γράφουμε το όνομα της ιδεατής μηχανικής (συνήθως υποδηλώνει το λειτουργικό σύστημα που εκτελείται).

Folder: ορίζουμε το φάκελο που θα αποθηκευτούν τα αρχεία που απαιτούνται για την εκτέλεση της ιδεατής μηχανής.

ISO Image: δηλώνουμε τη θέση στον κατάλογο αρχείων και το όνομα του αρχείου του λειτουργικού συστήματος που θα εγκατασταθεί.

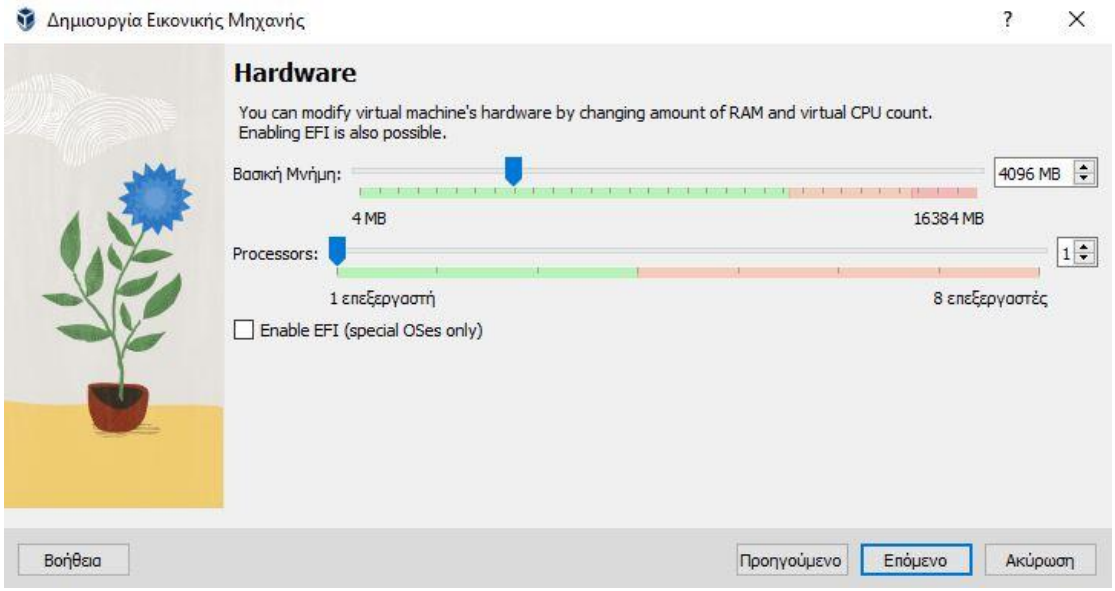
Edition-Τύπος-Έκδοση: Συμπληρώνονται αυτόματα από το όνομα του αρχείου ISO που χρησιμοποιούμε. Εναλλακτικά επιλέγουμε Red Hat (64-bit) εφόσον χρησιμοποιούμε αρχιτεκτονική 64-bit.



Εικόνα 4 Ρυθμίσεις εγκατάστασης εικονικής μηχανής

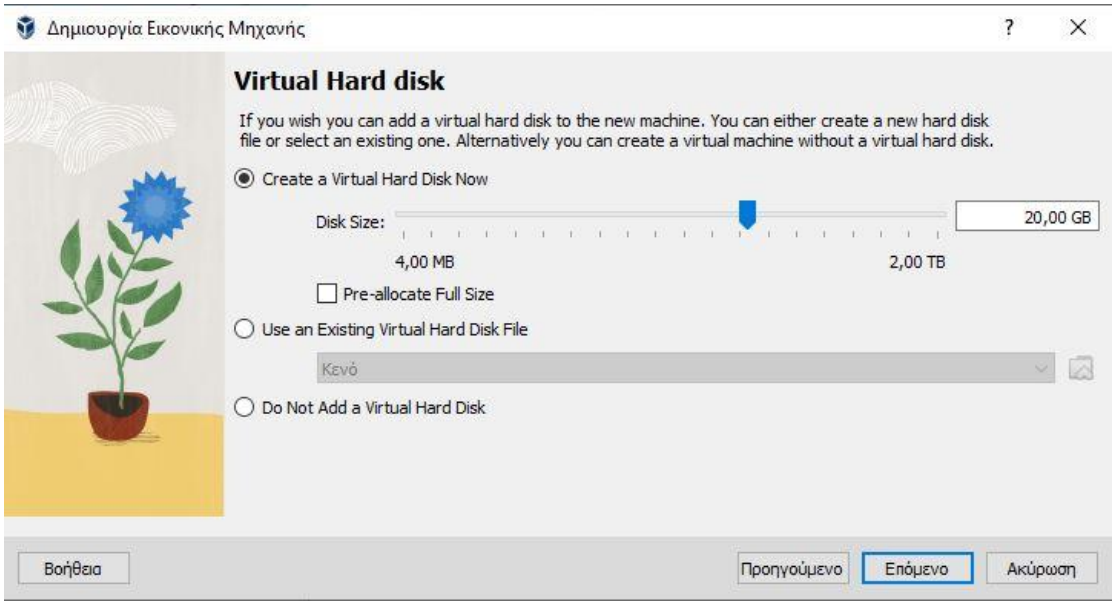
Στη συνέχεια προσδιορίζουμε το μέγεθος της RAM που θα χρησιμοποιηθεί από την ιδεατή μηχανή (VM – virtual machine). Όσο μεγαλύτερη είναι αυτή, τόσο γρηγορότερα θα αποκρίνεται το σύστημα μας.

Έτσι, είτε εισάγουμε τον αριθμό των MB που θα χρησιμοποιηθούν είτε μετακινούμε το slider στην επιθυμητή τιμή.



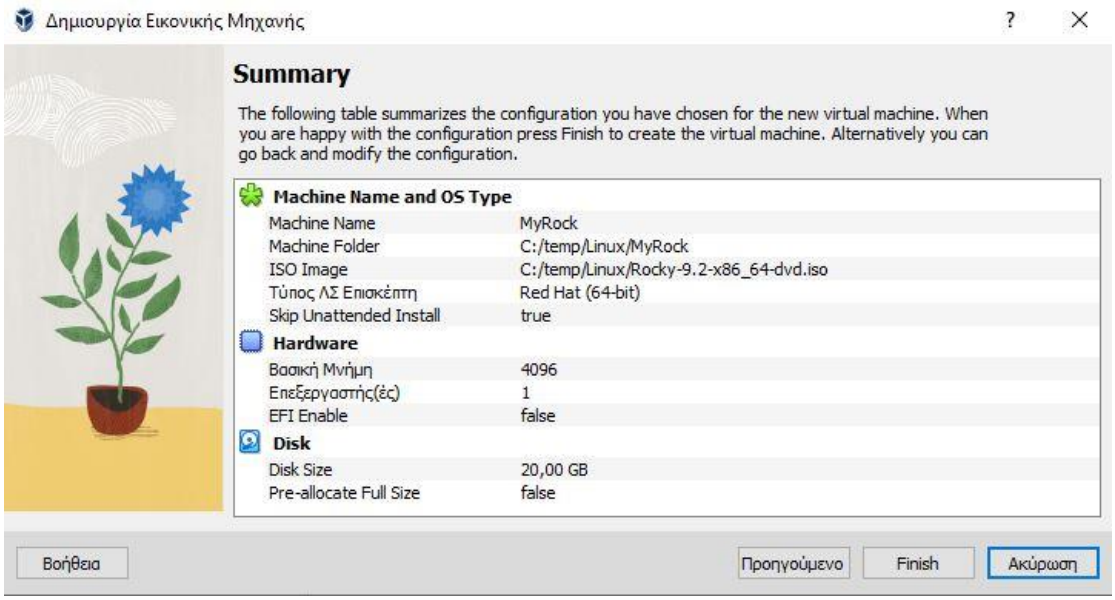
Εικόνα 5 Καθορισμός του μεγέθους της μνήμης RAM

Στο επόμενο βήμα επιλέγουμε τη δημιουργία ενός νέου ιδεατού σκληρού δίσκου επιλέγοντας *Create a Virtual Hard Disk Now*. Εναλλακτικά μπορούμε να χρησιμοποιήσουμε έναν υπάρχοντα σκληρό δίσκο ή να παραλείψουμε αυτό το βήμα στην περίπτωση που επιθυμούμε να δημιουργήσουμε ένα σκληρό δίσκο αργότερα.



Εικόνα 6 Καθορισμός του μεγέθους του σκληρού δίσκου

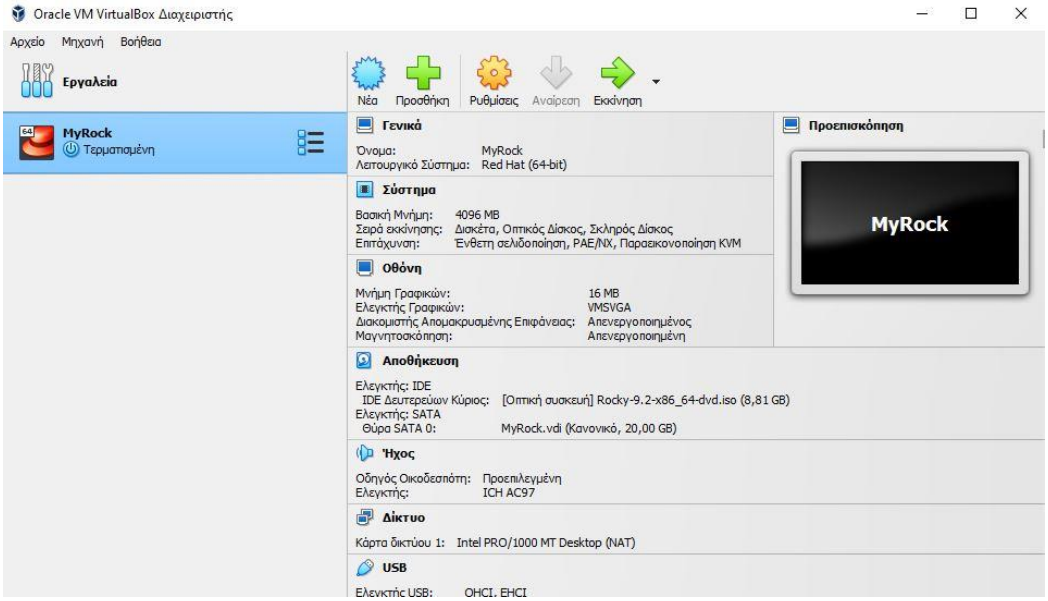
Στο τελευταίο βήμα εμφανίζεται μια σύνοψη των επιλογών που έχουν γίνει και επιλέγουμε *Finish* για την ολοκλήρωση των ρυθμίσεων της ιδεατής μηχανής.



Εικόνα 7 Ολοκλήρωση των ρυθμίσεων της ιδεατής μηχανής

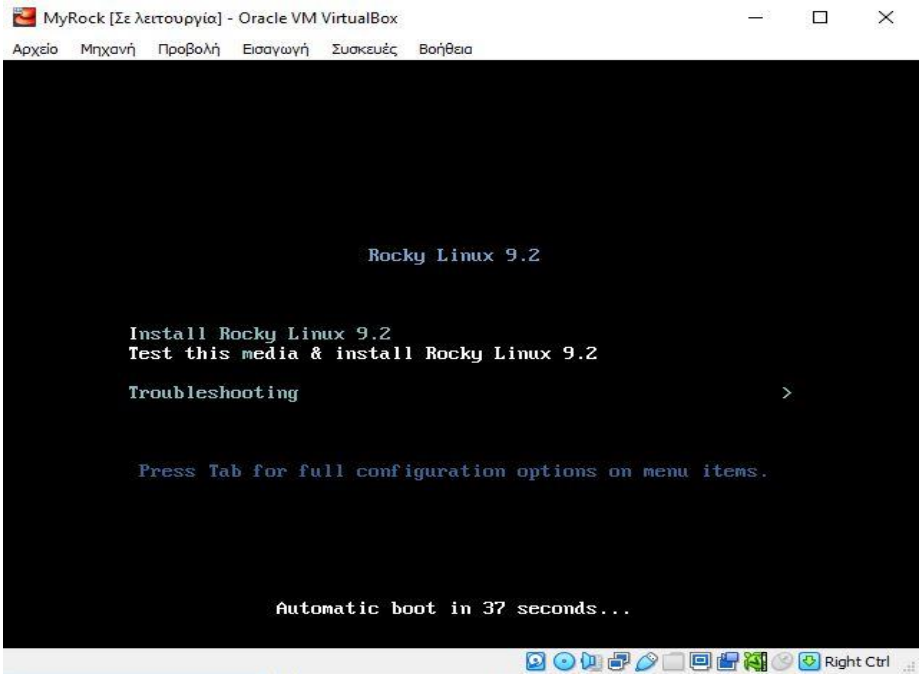
1.2 Πραγματοποίηση μιας εγκατάστασης βήμα-βήμα

Έχοντας πραγματοποιήσει όλα τα βήματα δημιουργία της ιδεατής μηχανής, αυτή εμφανίζεται στο αριστερό μέρος της οθόνης του διαχειριστή Oracle VM VirtualBox. Έχοντας επιλεγμένη την ιδεατή μηχανή *MyRock* πατάμε το πλήκτρο *Εκκίνηση*.



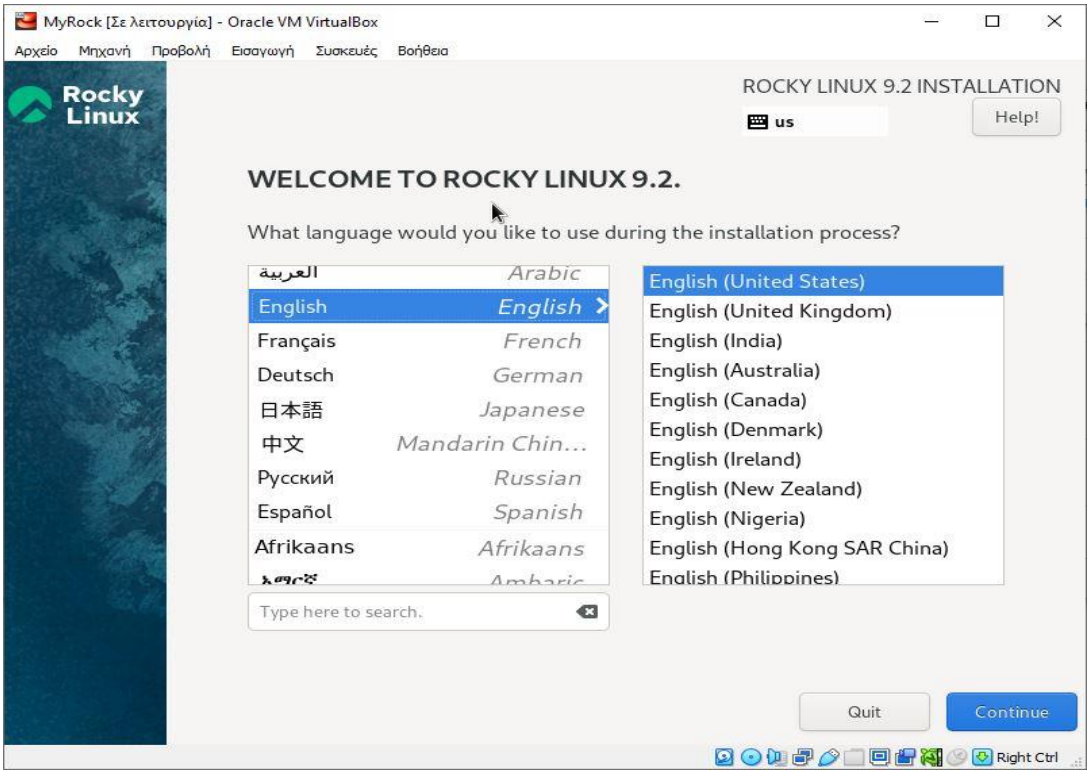
Εικόνα 8 Αρχική οθόνη διαχειριστή ιδεατών μηχανών

Μετά την εκκίνηση της ιδεατής μηχανής, εμφανίζεται το μενού επιλογής εγκατάστασης του Rocky Linux. Επιλέγουμε την πρώτη επιλογή με τα πλήκτρα κατεύθυνσης είτε πατώντας το πλήκτρο *I* (Install) και στη συνέχεια *Enter*.



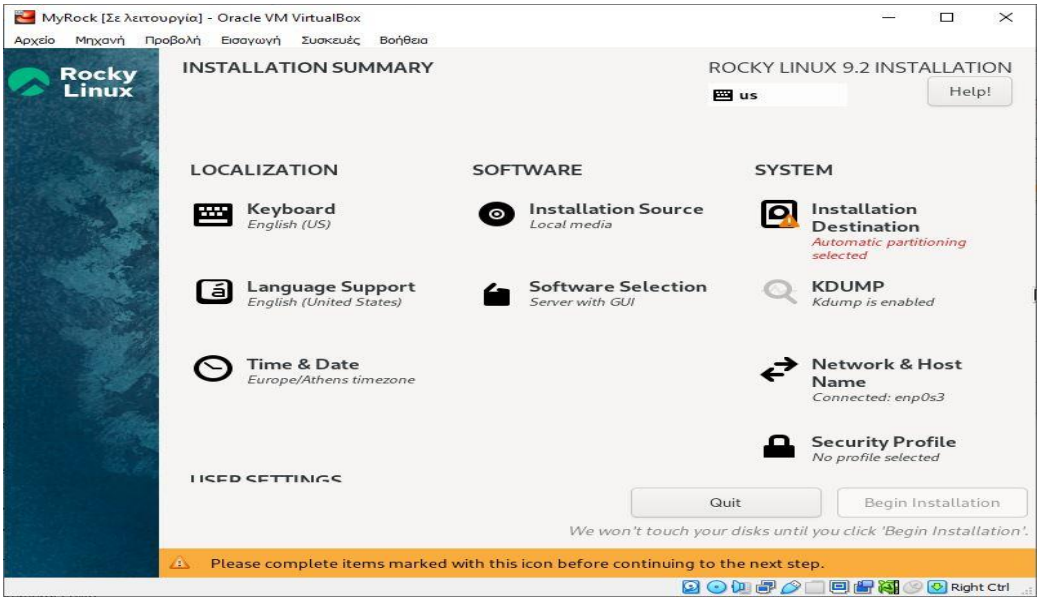
Εικόνα 9 Εκκίνηση της εγκατάστασης

Όταν φορτωθεί το πρόγραμμα εγκατάστασης στη μνήμη, εμφανίζεται η οθόνη επιλογής γλώσσας. Επιλέγουμε *English – English (United States)* και πατάμε *Continue*.



Εικόνα 10 Επιλογή γλώσσας εγκατάστασης

Τώρα το πρόγραμμα εγκατάστασης παρουσιάζει μια οθόνη που συνοψίζει τις πιο σημαντικές παραμέτρους για την εγκατάσταση του λειτουργικού συστήματος.



Εικόνα 11 Σύνοψη παραμέτρων εγκατάστασης του λειτουργικού συστήματος

Αυτές οι παράμετροι συνοψίζονται σε τέσσερις κατηγορίες:

Localization: περιλαμβάνονται ρυθμίσεις που αφορούν στη γλώσσα, την ώρα και την ημερομηνία.

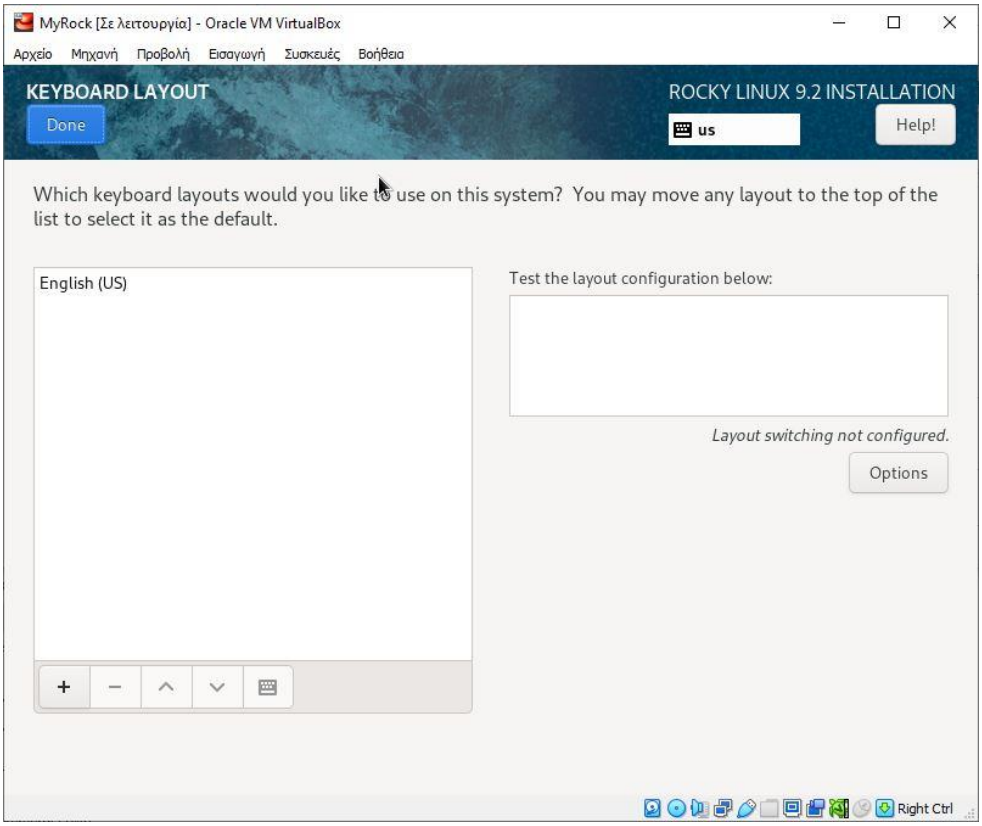
Software: περιλαμβάνονται ρυθμίσεις επιλογής του βασικού αλλά και του επιπλέον λογισμικού που θα εγκατασταθεί.

System: είναι οι ρυθμίσεις σχετικά με τις διαμερίσεις του ρίσκου, τις ρυθμίσεις δικτύου και την ασφάλεια.

User Settings: περιλαμβάνονται οι ρυθμίσεις λογαριασμού χρήστη και κωδικού πρόσβασης.

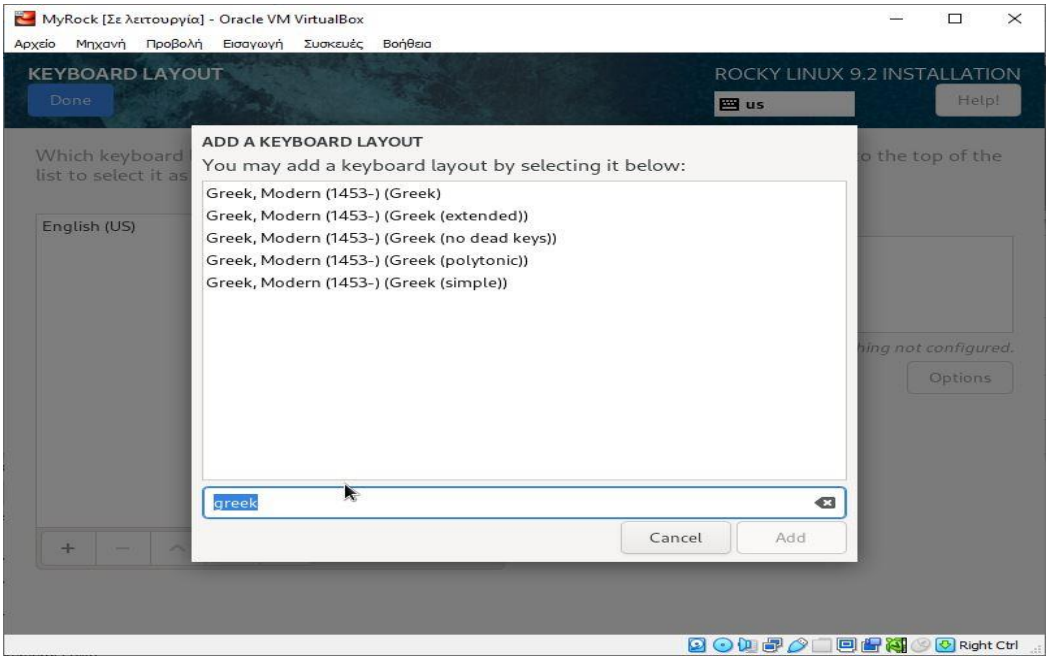
Ξεκινώντας από την επιλογή πληκτρολογίου έχουμε τη δυνατότητα να επιλέξουμε διαφορετικές διατάξεις πληκτρολογίου. Η προεπιλεγμένη διάταξη πληκτρολογίου έχει ήδη προστεθεί στο αριστερό τμήμα βάσει της γλώσσας που έχει επιλεγεί κατά τη διαδικασία εγκατάστασης.

Πατώντας στο πλήκτρο + στο αριστερό μέρος της οθόνης μπορούμε να εγκαταστήσουμε νέα διάταξη από λίστα επιλογών αλλά και να δοκιμάσουμε αυτή τη διάταξη στο πλαίσιο κειμένου στα δεξιά προκειμένου να βεβαιωθούμε ότι είναι σωστή.



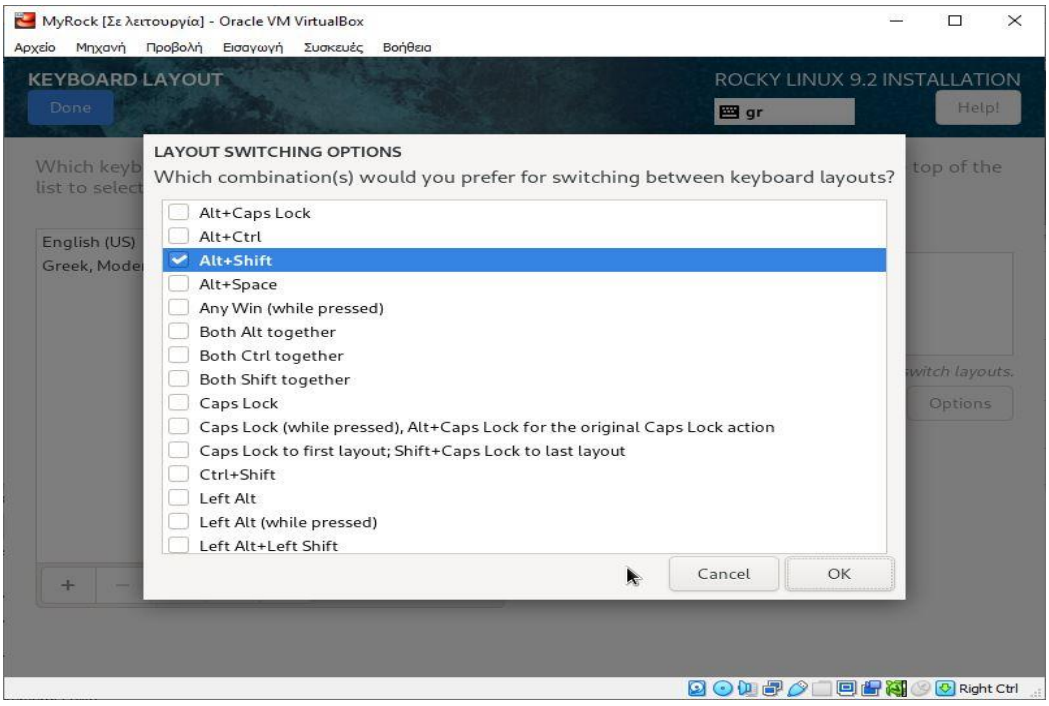
Εικόνα 12 Επιλογή διάταξης πληκτρολογίου

Από τη λίστα διατάξεων πληκτρολογίου γράφοντας **greek** στο πλαίσιο εισαγωγής κειμένου εμφανίζονται οι διαφορετικές διατάξεις για την ελληνική γλώσσα και επιλέγουμε **Greek, Modern (1453-)(Greek).**



Εικόνα 13 Επιλογή διάταξης πληκτρολογίου (ελληνικά)

Στη συνέχεια επιλέγοντας Options έχουμε τη δυνατότητα να ορίσουμε το συνδυασμό πλήκτρων εναλλαγής γλώσσας από το πληκτρολόγιο (εισάγουμε Alt + Shift).

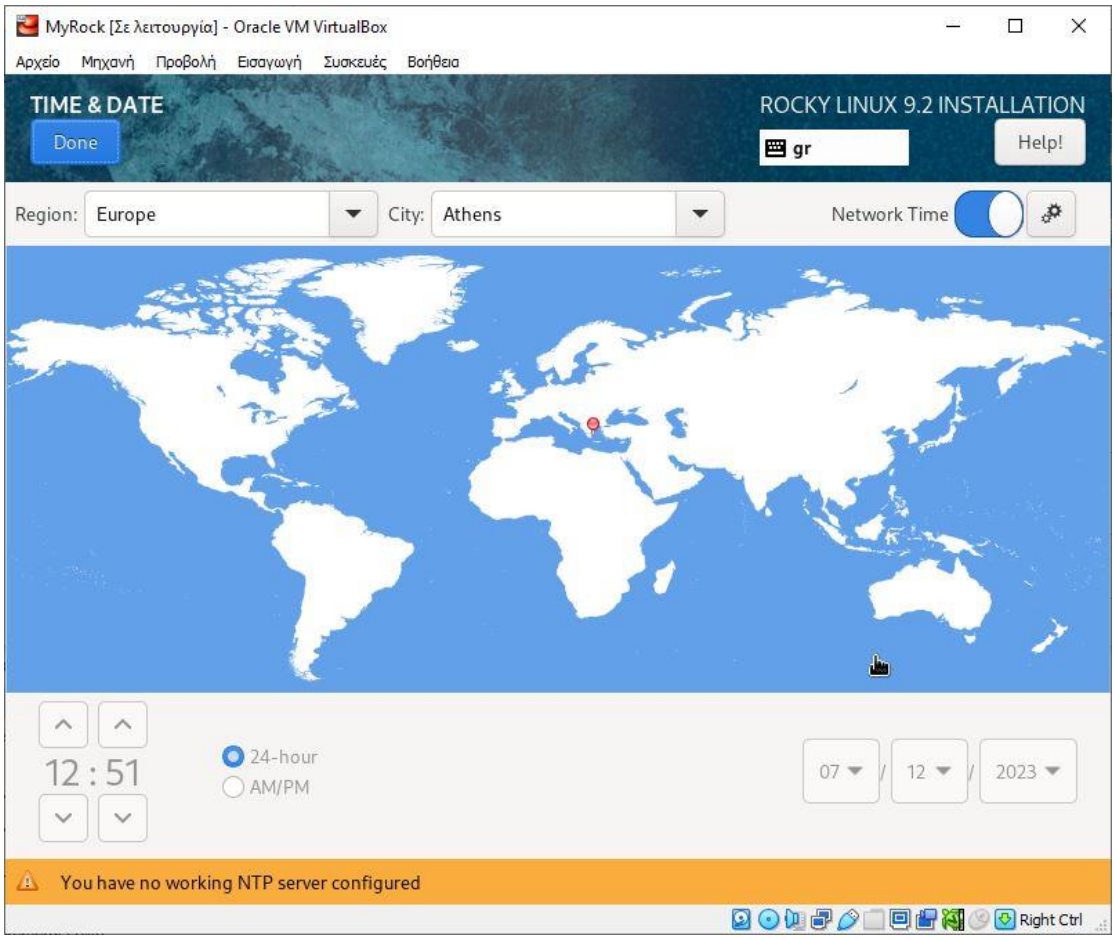


Εικόνα 14 Επιλογή συνδυασμού πλήκτρων για εναλλαγή γλώσσας

Επιλέγουμε το πλήκτρο ***Done*** όταν ολοκληρώσουμε την επιλογή μας.

Από την επιλογή Time & Date έχουμε τη δυνατότητα να ρυθμίσουμε την ώρα και ημερομηνία για το λειτουργικό μας σύστημα. Αυτά αναγνωρίζονται αυτόματα στην περίπτωση που έχουμε ενεργή σύνδεση στο Διαδίκτυο και είναι ενεργοποιημένος και ο διακόπτης Network Time (στο επάνω δεξιό μέρος της οθόνης).

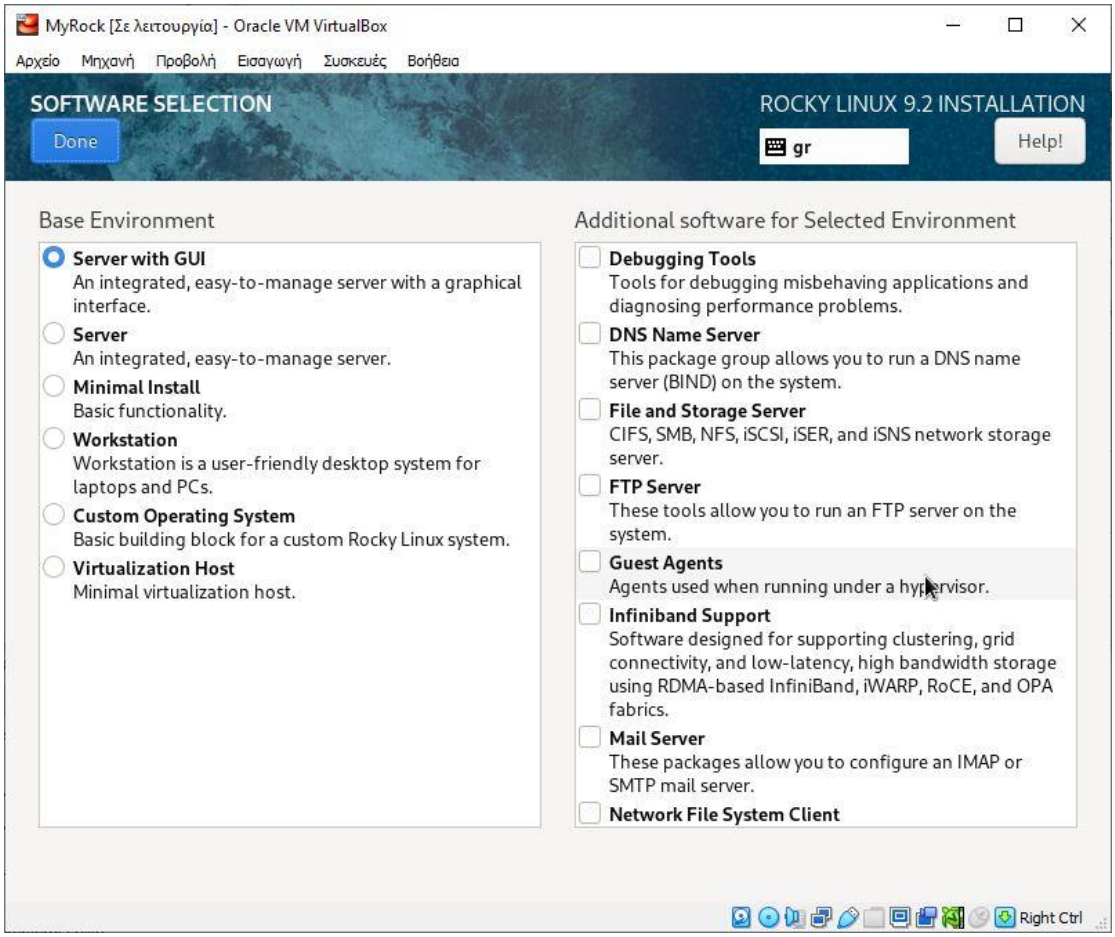
Στην περίπτωση που δεν υπάρχει ενεργή σύνδεση στο Διαδίκτυο, επιλέγουμε την περιοχή και τη πόλη που βρισκόμαστε και ρυθμίζουμε την ώρα είτε σε μορφή **24-hour** είτε σε μορφή **AM/PM.**



Εικόνα 15 Ρύθμιση Ημερομηνίας και ώρας

Στη συνέχεια επιλέγουμε ***Software Selection*** όπου μας δίνεται η δυνατότητα επιλογής ενός από περισσότερα περιβάλλοντα (**Base Environment**) και επιπλέον λογισμικό από το μενού **Additional software for Selected Environment**.

Ολοκληρώνοντας την επιλογή λογισμικού πατάμε Done για επιστροφή στο κεντρικό μενού εγκατάστασης.



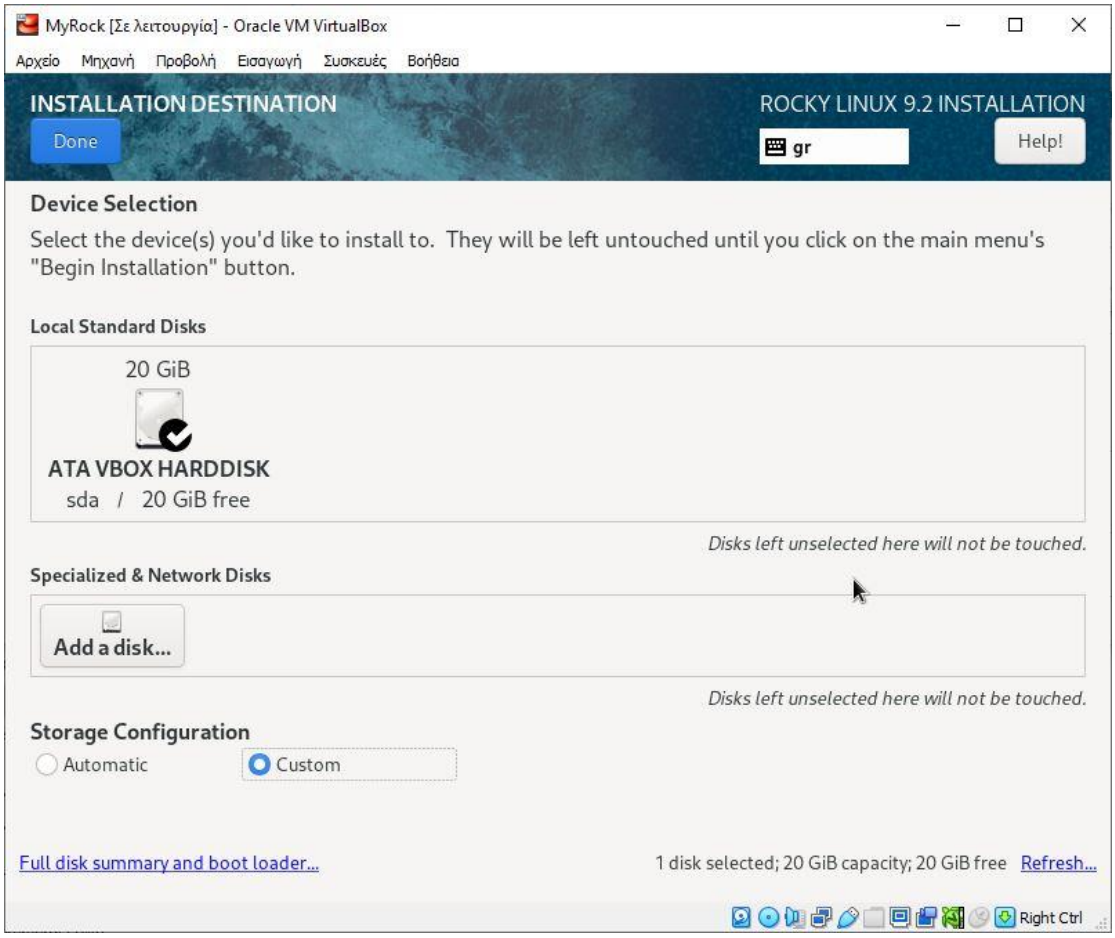
Εικόνα 16 Επιλογή λογισμικού

Από την περιοχή System επιλέγουμε ***Installation Destination*** για να εισάγουμε ότι αφορά στην τοποθεσία εγκατάστασης και τις επιλογές διαμερίσεων.

Αρχικά βεβαιωνόμαστε ότι έχει επιλεγεί ο δίσκος στον οποίο θα εγκαταστήσουμε το Rocky Linux και ότι υπάρχει *σημάδι επιλογής* (**check mark**) πάνω του στην περιοχή **Local Standard Disks**.

Στην περιοχή *Διαμόρφωση Αποθήκευσης* (**Storage Configuration**) επιλέγουμε είτε την επιλογή *Αυτόματο* (**Automatic**) για να ρυθμίσουμε αυτόματα την αποθήκευση ή την επιλογή *Προσαρμοσμένο* (**Custom**) για να δημιουργήσουμε προσαρμοσμένες διαμερίσεις στον δίσκο.

Επιλέγοντας *Αυτόματο* (**Automatic**) μας προτείνεται η δομή που εμφανίζεται στην επόμενη οθόνη.

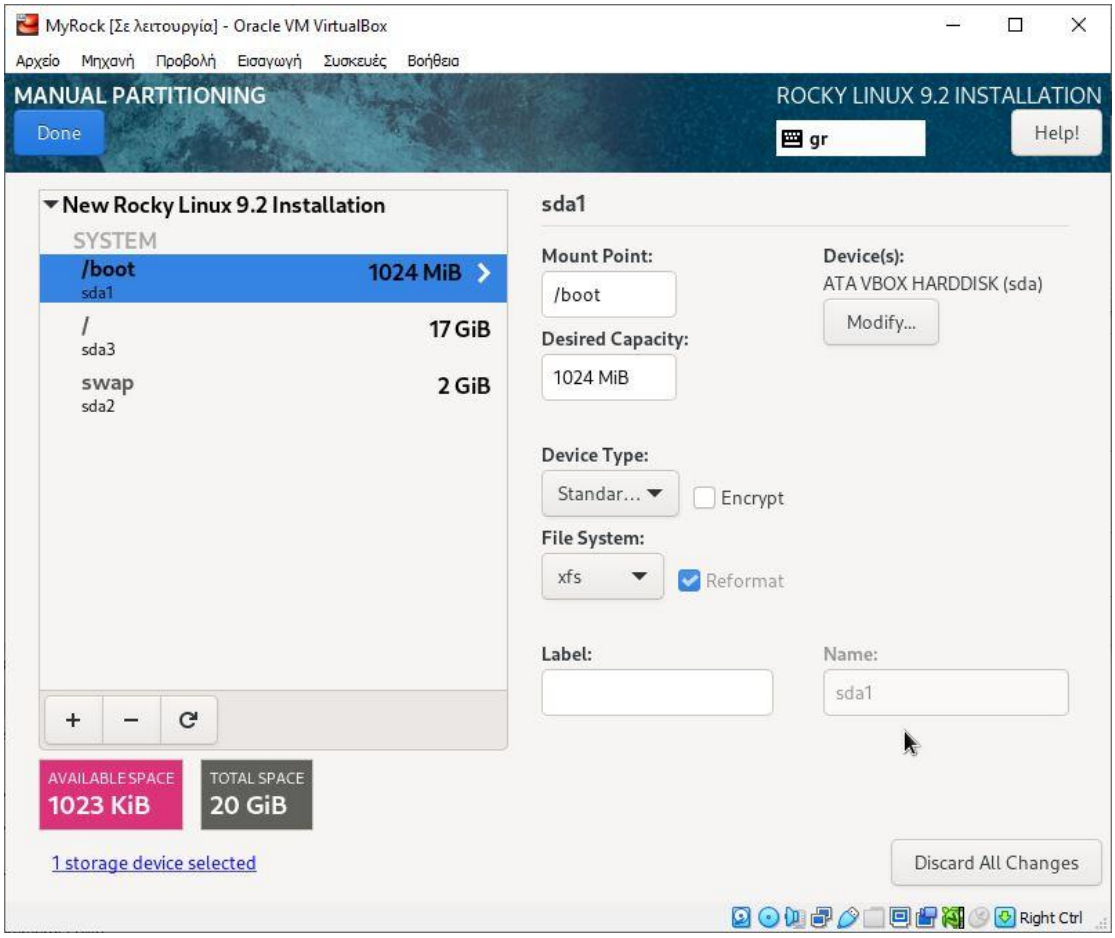


Εικόνα 17 Προορισμός εγκατάστασης (Installation Destination)

Παρατηρούμε ότι έχουν δημιουργηθεί τρεις διαμερίσεις που μας είναι επαρκείς για τα εκπαιδευτικά εργαστήρια που θα πραγματοποιήσουμε. Αυτές είναι:

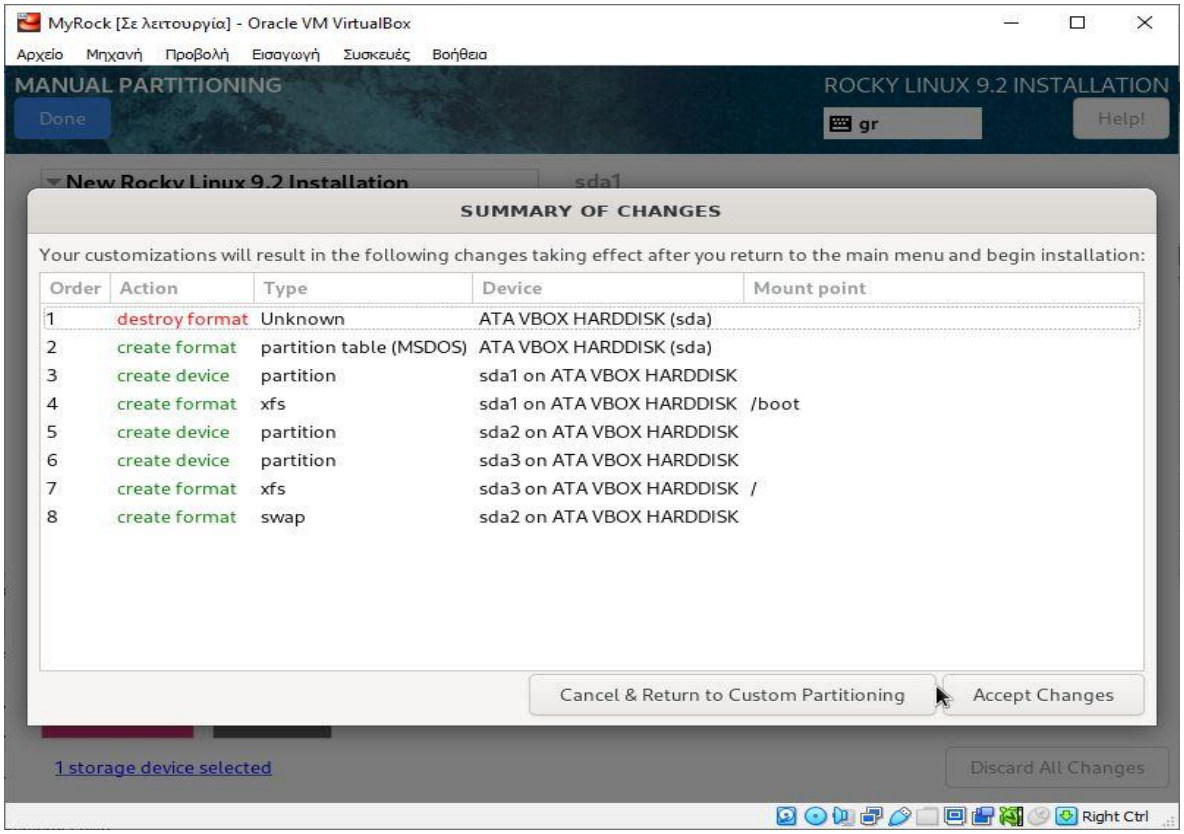
- **/boot** – 1024 MiB
- / – 17 GiB
- **swap** – 2 GiB

Μετά την ολοκλήρωση της ρύθμισης των διαμερίσεων επιλέγουμε το πλήκτρο **Done**.



Εικόνα 18 Πίνακας διαμερίσεων

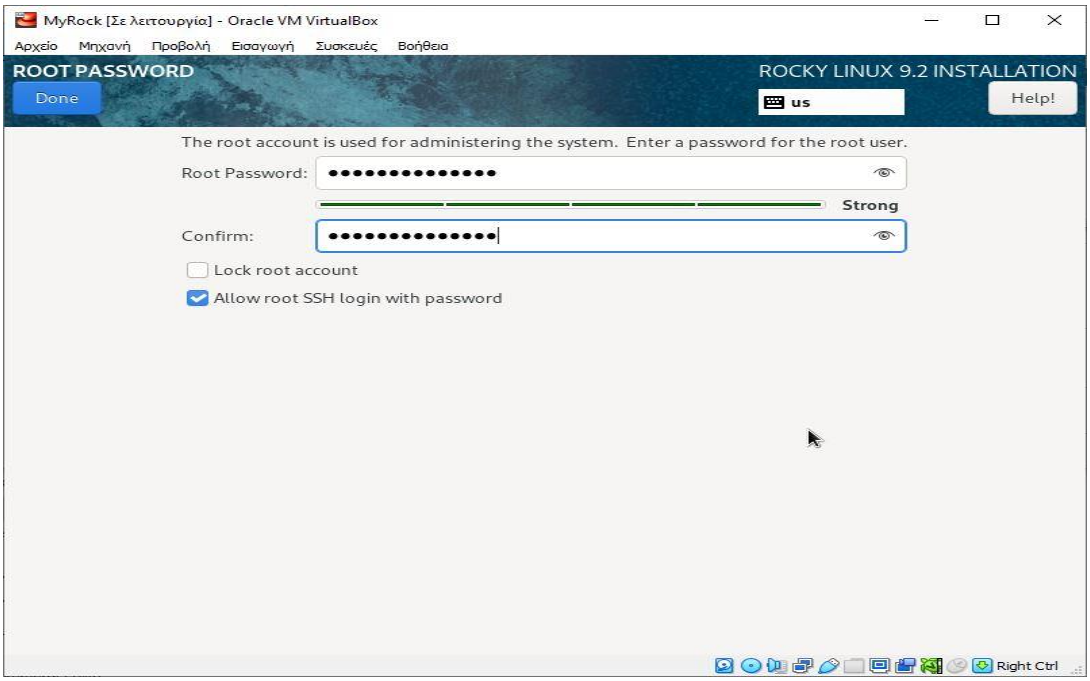
Σε αυτό το σημείο εμφανίζεται η σύνοψη των αλλαγών που θα γραφούν στο δίσκο οπότε πατάμε *Αποδοχή αλλαγών* (**Accept Changes**)



Εικόνα 19 Σύνοψη διαμερίσεων

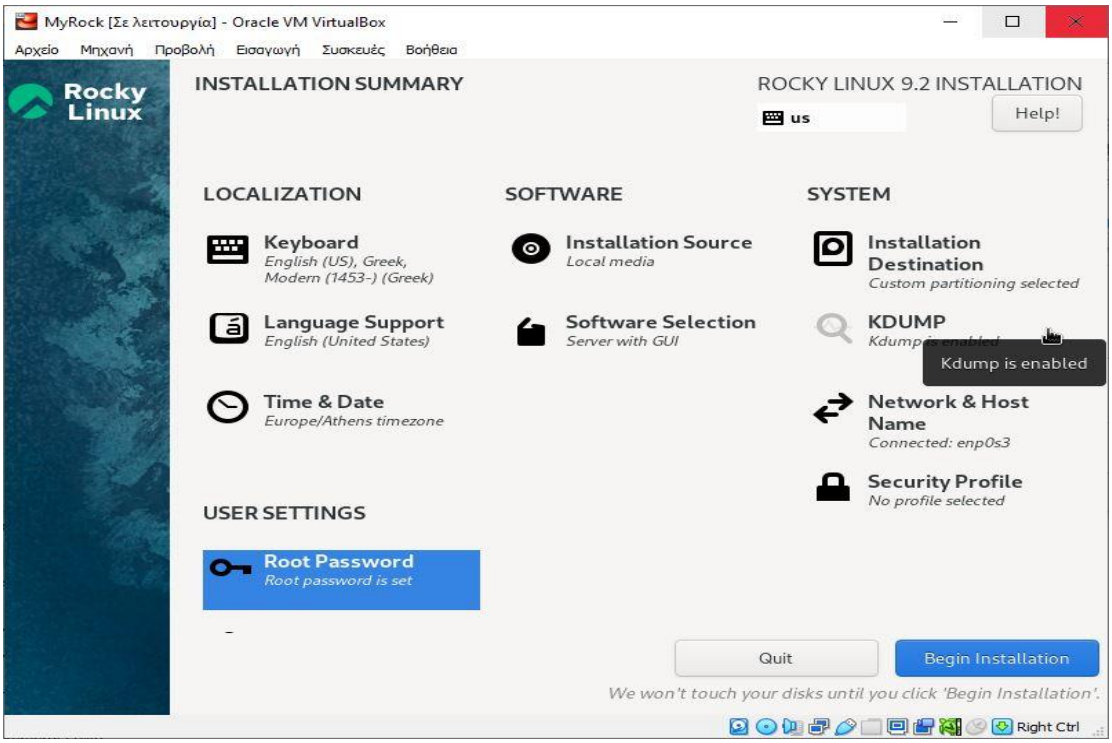
Πατώντας την επιλογή **Root Password** έχουμε τη δυνατότητα να ορίσουμε τον κωδικό του διαχειριστή του συστήματος. Εισάγουμε έναν *ισχυρό κωδικό* στο πρώτο πεδίο ενώ τον επιβεβαιώνουμε στο δεύτερο πεδίο.

Προσοχή: Επειδή η τελευταία έκδοση του Rocky Linux απενεργοποιεί τις απομακρυσμένες συνδέσεις του διαχειριστή (root) μέσω SSH από προεπιλογή, εμείς επιλέγουμε το **Allow root SSH login with password** (*Να επιτραπεί η σύνδεση root μέσω SSH με κωδικό*) για να την ενεργοποιήσουμε.



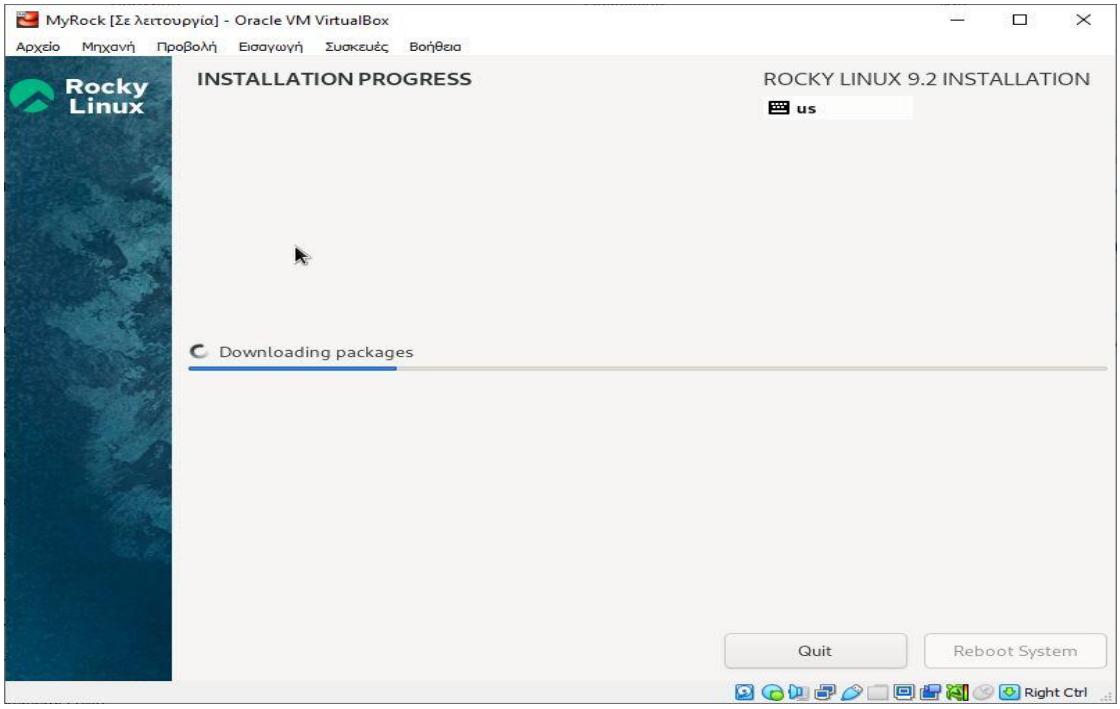
Εικόνα 20 Εισαγωγή κωδικού για το λογαριασμό Root

Σε αυτό το σημείο έχουμε ολοκληρώσει τις ρυθμίσεις οπότε επιλέγουμε το πλήκτρο Begin Installation (εκκίνηση εγκατάστασης) στο κάτω δεξί μέρος της οθόνης.



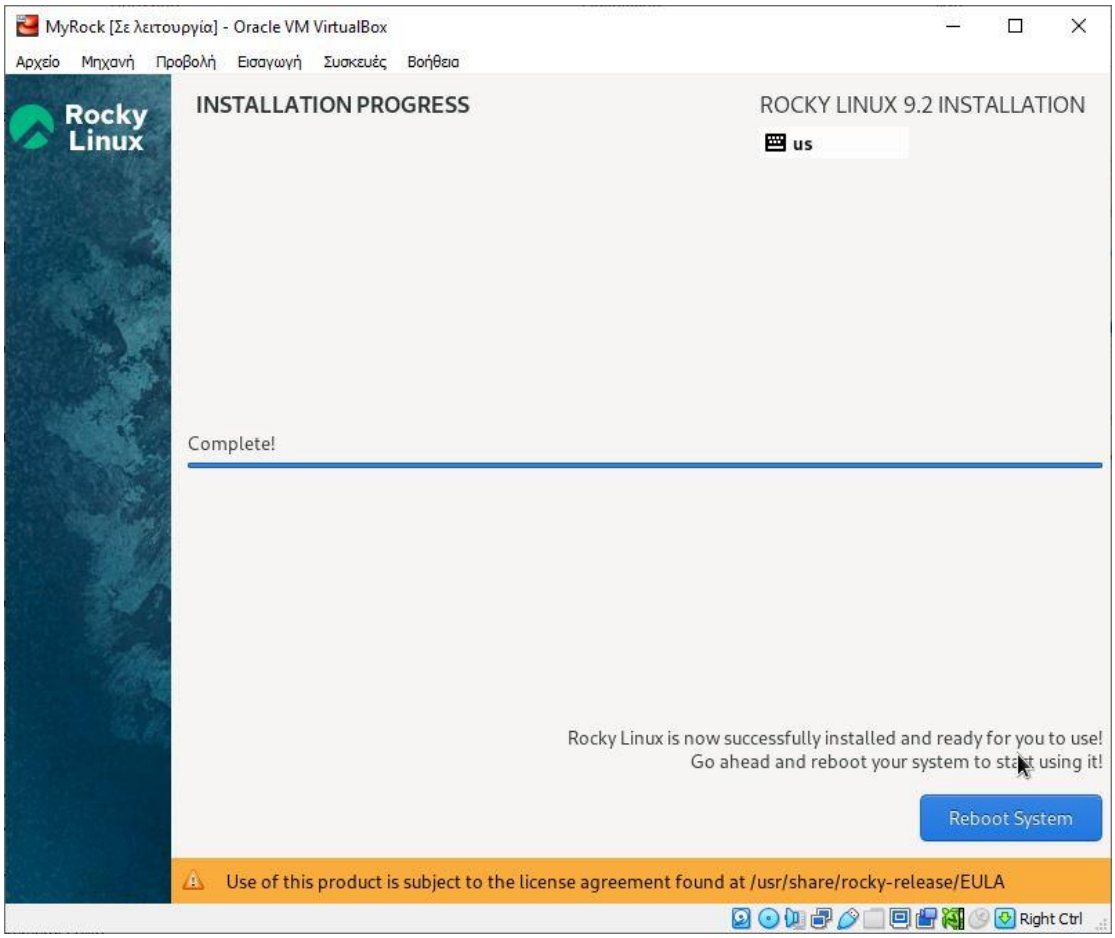
Εικόνα 21 Εκκίνηση εγκατάστασης

Κατά τη διαδικασία εγκατάστασης γίνεται κατέβασμα των απαιτούμενων πακέτων.



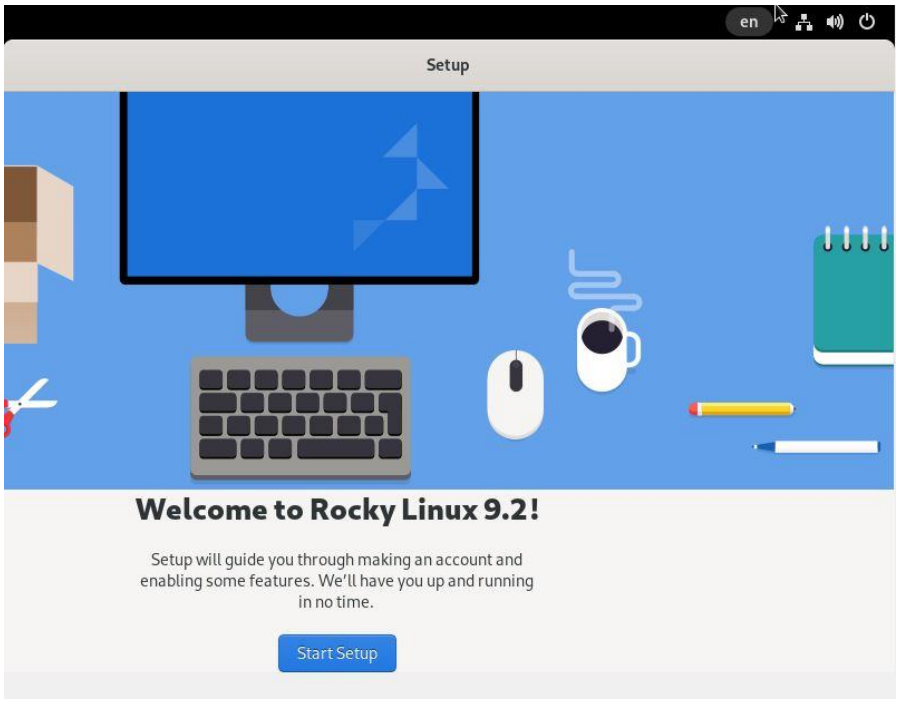
Εικόνα 22 Εγκατάσταση (κατέβασμα πακέτων)

Και σε αυτό το σημείο ολοκληρώνεται η εγκατάσταση οπότε επιλέγουμε την επανεκκίνηση του συστήματος.



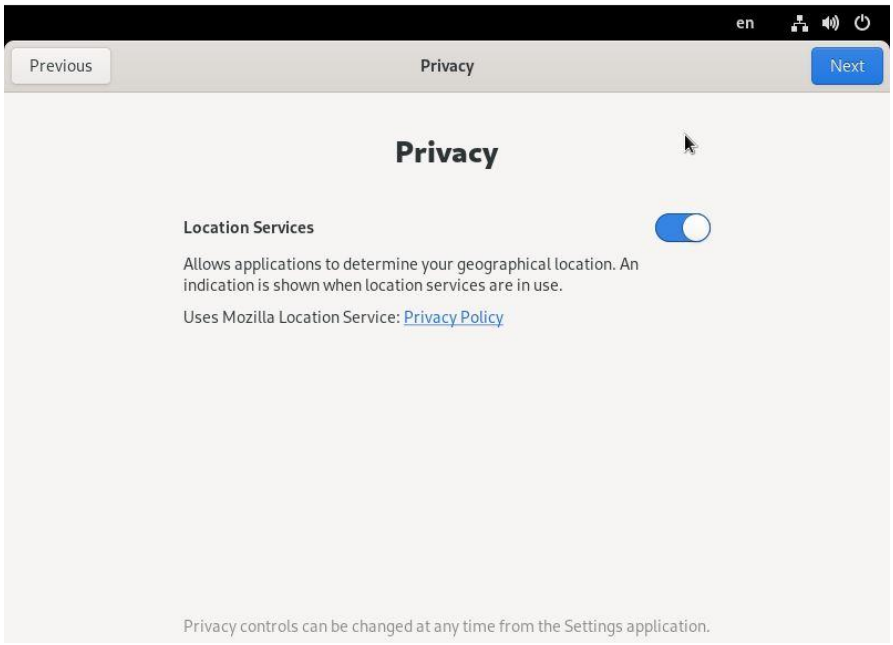
Εικόνα 23 Ολοκλήρωση εγκατάστασης και επανεκκίνηση του συστήματος

Μετά την επανεκκίνηση του συστήματος εμφανίζεται η οθόνη καλωσορίσματος από όπου συνεχίζουμε τη ρύθμιση του συστήματος.



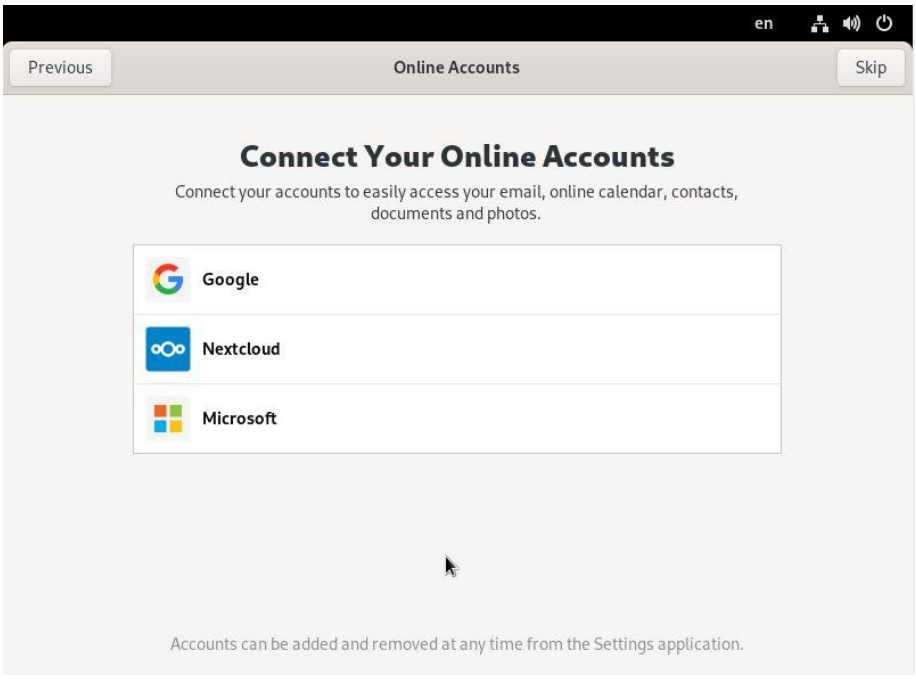
Εικόνα 24 Οθόνη καλωσορίσματος

Αρχικά ενεργοποιούμε τον γεωχωρικό εντοπισμό της μηχανής.



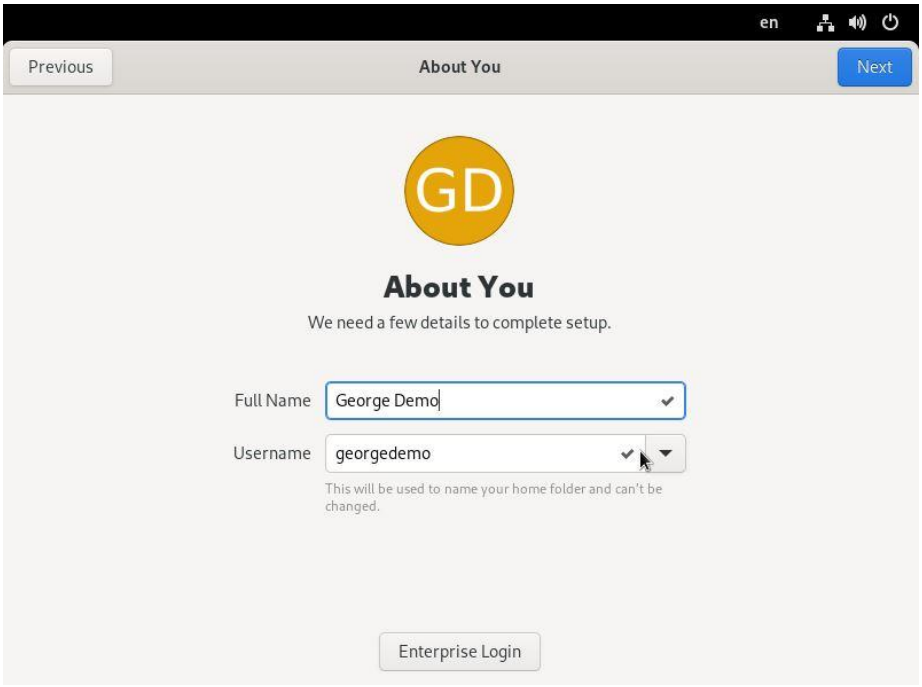
Εικόνα 25 Ρύθμιση υπηρεσιών γεωχωρικού εντοπισμού

Στην περίπτωση μη κοινόχρηστου υπολογιστή μπορούμε να πραγματοποιήσουμε σύνδεση με κάποιον από τους λογαριασμούς μας στο νέφος (cloud).



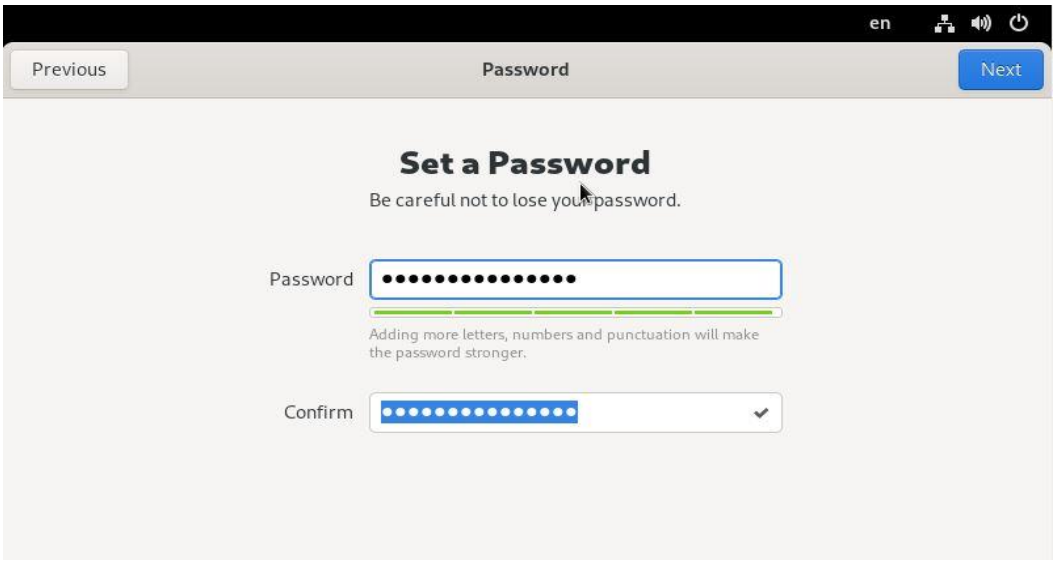
Εικόνα 26 Σύνδεση με λογαριασμό στο νέφος

Εισάγουμε το πλήρες όνομα και το όνομα χρήστη.



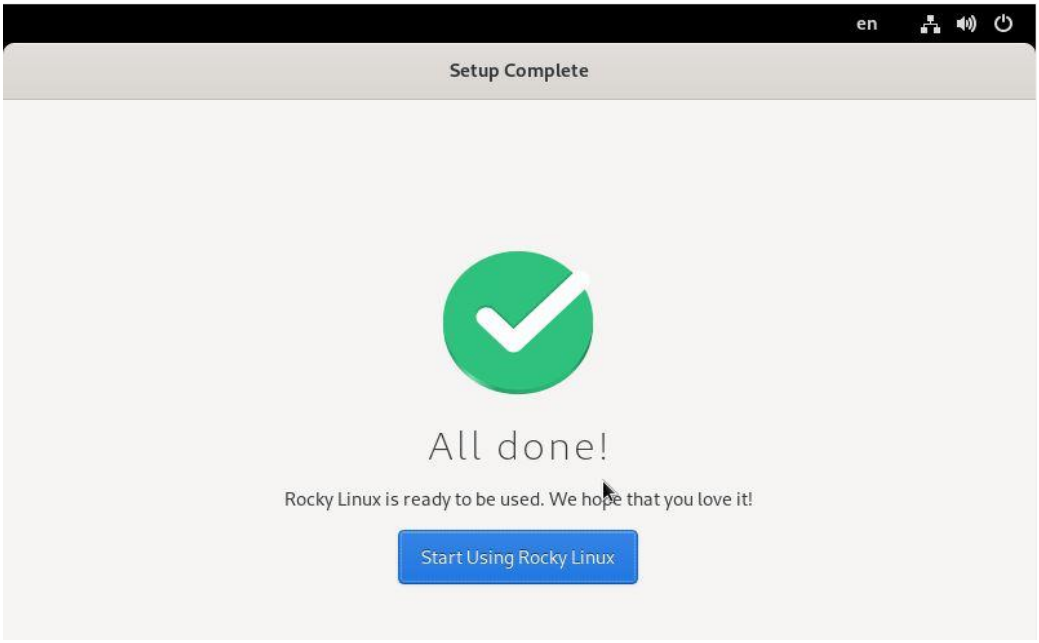
Εικόνα 27 Εισαγωγή πλήρους ονόματος και ονόματος χρήστη

Εισάγουμε τον έναν ισχυρό κωδικό στο πρώτο πεδίο ενώ τον επαληθεύουμε στο δεύτερο πεδίο.



Εικόνα 28 Εισαγωγή και επαλήθευση κωδικού χρήστη

Σε αυτό το σημείο έχει ολοκληρωθεί η εγκατάσταση και μπορούμε να ξεκινήσουμε να χρησιμοποιούμε το σύστημα μας.



Εικόνα 29 Έναρξη χρήσης του συστήματος

1.4 Είσοδος στο λειτουργικό σύστημα από μια τοπική κονσόλα κειμένου και εκτέλεση απλών εντολών στον φλοιό bash

Η *γραμμή εντολών* είναι ένας τρόπος εισαγωγής εντολών σε ένα υπολογιστικό σύστημα, βασισμένος στο κείμενο. Στο Linux, όπως και σε όλα τα λειτουργικά συστήματα τύπου unix, η πρόσβαση στη γραμμή εντολής γίνεται μέσω ενός προγράμματος που ονομάζεται *φλοιός (shell)*. Υπάρχουν αρκετές επιλογές για το πρόγραμμα φλοιού, έχουν αναπτυχθεί διάφορα μέσα στα χρόνια και κάθε χρήστης μπορεί να ρυθμιστεί να χρησιμοποιεί τον δικό του φλοιό. Η πλειοψηφία κρατάει τον προκαθορισμένο.

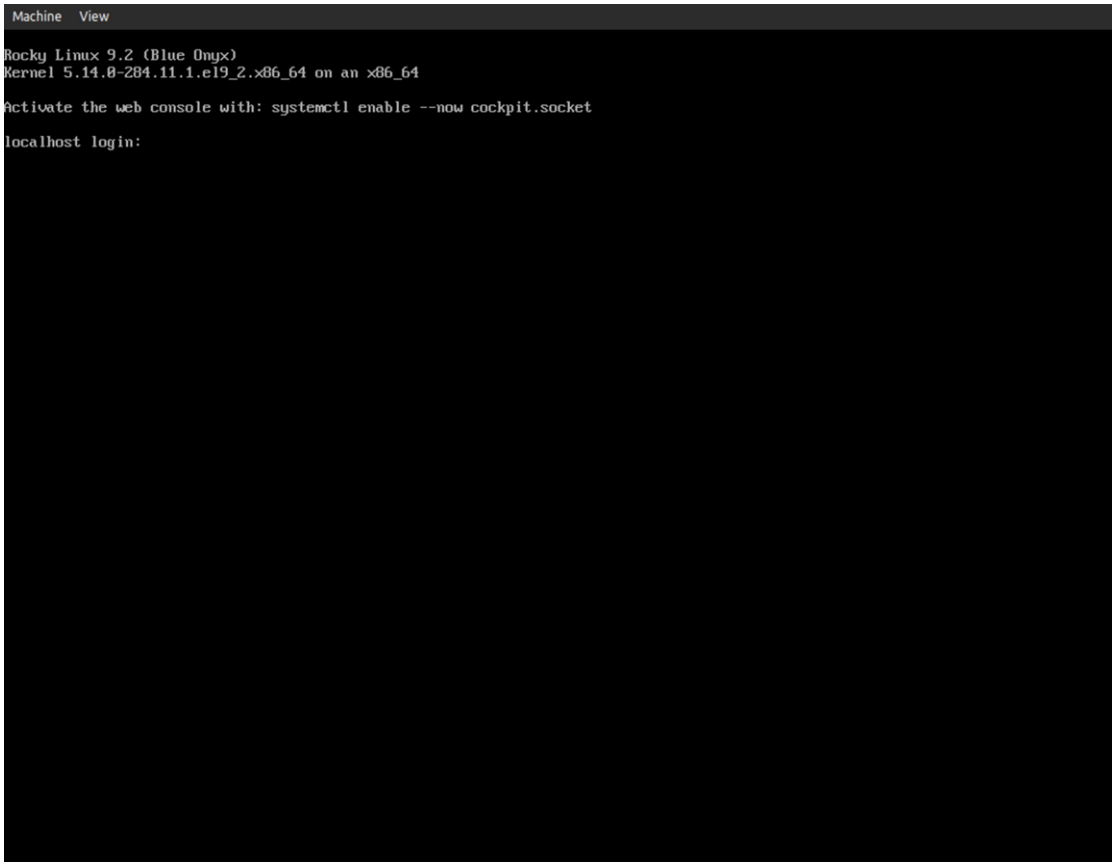
Για το Red Hat Enterprise Linux (και τη συντριπτική πλειοψηφία), αυτός είναι ο **bash**, ο GNU Bourne-Again Shell. Είναι μια βελτιωμένη έκδοση ενός από τους πιο επιτυχημένους φλοιούς, του Bourne shell (**sh**).

Σημείωση: Τα λογοπαίγνια είναι αρκετά συνηθισμένα στον κόσμο του unix και των υπολογιστών γενικότερα. Το αν είναι πετυχημένα ή όχι είναι και θέμα προσωπικού γούστου. Εδώ είναι διπλό: bash σημαίνει «χτυπάω κάτι δυνατά, λιώνω», ενώ το Bourne-Again προέρχεται από το born-again: αναγεννημένος/η και το Bourne, το όνομα του δημιουργού του αρχικού Bourne shell.

Για να τρέξουμε το φλοιό, πρέπει να συνδεθούμε (**log in**) στον υπολογιστή με ένα *τερματικό*. Το τερματικό είναι μια διεπαφή που χρησιμοποιεί κείμενο για την εισαγωγή εντολών και την εκτύπωση εξόδου από ένα υπολογιστικό σύστημα. Υπάρχουν αρκετοί τρόποι για να γίνει αυτό.

Ο υπολογιστής μπορεί να έχει συνδεμένα πληκτρολόγιο και οθόνη (εννοούμε φυσικές συσκευές) για είσοδο και έξοδο, που αποτελούν τη *φυσική κονσόλα* του Linux. Αυτή υποστηρίζει πολλές *εικονικές κονσόλες*, που μπορούν να τρέχουν διαφορετικά τερματικά. Κάθε εικονική κονσόλα υποστηρίζει μια ανεξάρτητη συνεδρία σύνδεσης. Η εναλλαγή μεταξύ τους γίνεται πατώντας τα **Ctrl + Alt** και ένα πλήκτρο λειτουργίας (function key), από το **F2** ως το **F6**, ταυτόχρονα. Οι εικονικές κονσόλες έχουν τα ονόματα tty2 – tty6, αντίστοιχα. Σε αυτές τρέχει ένα τερματικό που ζητάει την εισαγωγή ενός ονόματος χρήστη και ενός συνθηματικού. Αν τα εισάγουμε σωστά, συνδεόμαστε και μας υποδέχεται ένας φλοιός, στον οποίο μπορούμε να δώσουμε εντολές. Με τα πλήκτρα **F2** ως **F6**, μπορούμε να έχουμε ταυτόχρονα έως πέντε ανεξάρτητες εικονικές κονσόλες.

Σημείωση: αν το σύστημα Linux που χρησιμοποιείτε τρέχει σε VM, τότε πιθανό να πρέπει να συμβουλευτείτε τις οδηγίες του προγράμματος ανάπτυξης ιδεατών μηχανών που χρησιμοποιείτε (π.χ. VirtualBox) για το πώς μπορείτε να περάσετε συνδυασμούς πλήκτρων όπως το **Ctrl + Alt +F1** ή **Ctrl + Alt + Delete** στην εικονική μηχανή, γιατί αυτούς τους συνδυασμούς τους «κλέβει» το λειτουργικό σύστημα του φυσικού μηχανήματος για τον εαυτό του. Αυτό σημαίνει επίσης ότι χρειάζεται προσοχή όταν δίνετε τέτοιους συνδυασμούς, ώστε να μην έχετε ανεπιθύμητες επανεκκινήσεις του φυσικού μηχανήματος.



Εικόνα 30 Η εικονική κονσόλα του Rocky Linux (είσοδος χρήστη)

Όταν ο φλοιός χρησιμοποιείται διαδραστικά, εμφανίζει ένα αλφαριθμητικό, που δείχνει ότι αναμένει μια εντολή από την/ον χρήστρια/η και ονομάζεται *shell prompt*. Για τους κανονικούς χρήστες, όταν ξεκινάει ο φλοιός έχει prompt που τελειώνει με τον χαρακτήρα **$**:

```
[user@host ~]$
```

Ο χαρακτήρας **$** αντικαθίσταται από τον χαρακτήρα # αν ο φλοιός τρέχει σαν υπερχρήστης (διαχειριστής, root). Έτσι είναι εμφανέστερο ότι πρόκειται για φλοιό υπερχρήστη και βοηθάει στη αποφυγή λαθών και ατυχημάτων που μπορούν να επηρεάσουν ολόκληρο το σύστημα. Το prompt γίνεται κάπως έτσι:

```
[root@host ~]#
```

Η χρήση του **bash** για την εκτέλεση εντολών μπορεί να αποτελέσει ένα ισχυρό εργαλείο. Ο **bash** παρέχει μια γλώσσα σεναρίων (scripting language) που μπορεί να υποστηρίξει τον αυτοματισμό των εργασιών. Επίσης έχει επιπλέον δυνατότητες που μπορούν να απλοποιήσουν ή να καταστήσουν δυνατές λειτουργίες που είναι δύσκολο να επιτευχθούν με γραφικά εργαλεία.

Σημείωση: Ο **bash** είναι παρόμοιος σαν σύλληψη με τον διερμηνέα γραμμής εντολών στις πρόσφατες εκδόσεις των Microsoft Windows, τον **cmd,exe**, αν και ο **bash** έχει πιο εξεζητημένη γλώσσα σεναρίων. Είναι επίσης παρόμοιος με τον Windows PowerShell στα Windows 7 και Windows Server 2008 R2 και νεότερα. Αν χρησιμοποιείτε την εφαρμογή **Terminal** σε Apple Mac, θα χαρείτε να μάθετε ότι ο **bash** είναι ο προεπιλεγμένος φλοιός στο macOS.

ΒΑΣΙΚΑ ΣΤΟΙΧΕΙΑ ΤΟΥ ΦΛΟΙΟΥ

Οι εντολές που δίνονται στο φλοιό αποτελούνται από τρία βασικά μέρη:

- την προς εκτέλεση *εντολή*
- *επιλογές* που επηρεάζουν τη συμπεριφορά της εντολής
- *ορίσματα*, τυπικά τα αντικείμενα-στόχοι της εντολής

Η *εντολή* είναι το όνομα του προγράμματος που θα τρέξει. Μπορεί να ακολουθείται από μία ή περισσότερες *επιλογές*, που ρυθμίζουν τη συμπεριφορά της εντολής ή το τι θα κάνει. Οι επιλογές συνήθως αρχίζουν με μία ή δύο παύλες, ώστε να ξεχωρίζουν από τα ορίσματα. Οι εντολές μπορεί επίσης να ακολουθούνται και από ένα ή περισσότερα *ορίσματα*, τα οποία συνήθως είναι τα αντικείμενα στα οποία δρα η εντολή.

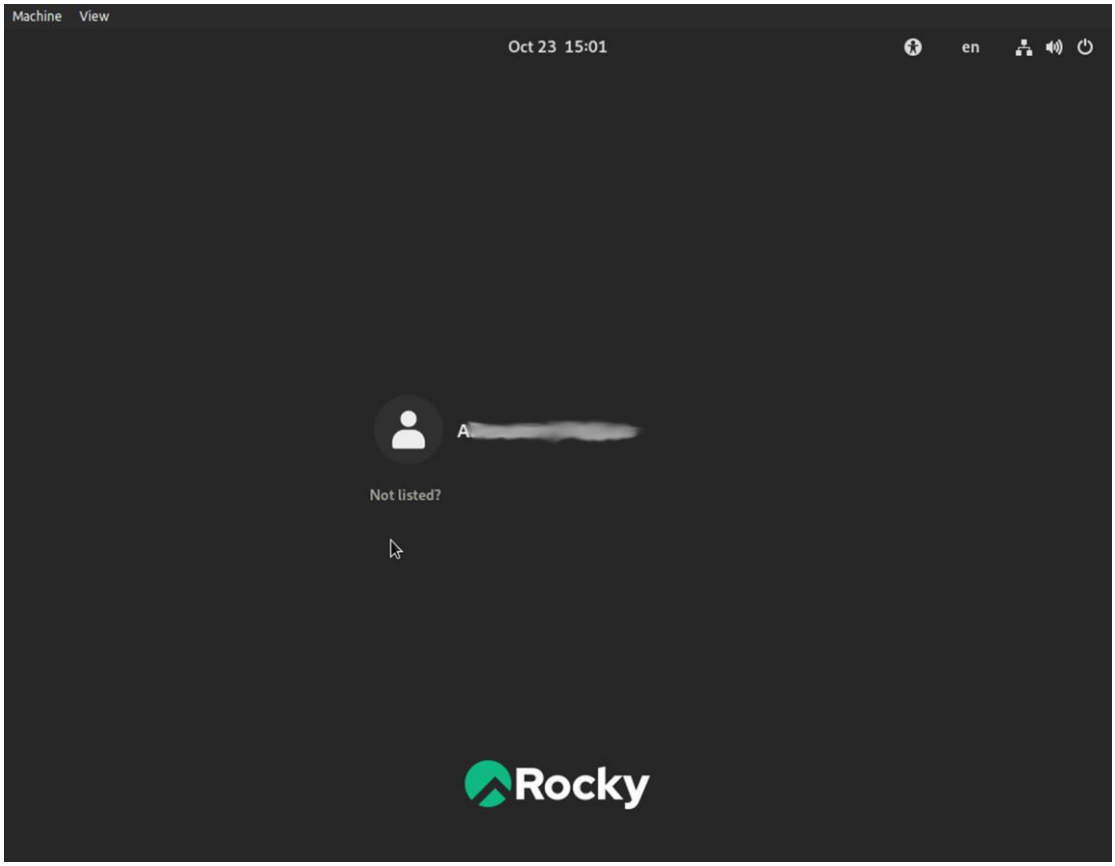
Σαν παράδειγμα, η εντολή **`ls -l /usr/bin`** αποτελείται από μια εντολή (**`ls`**), μια επιλογή (**`-l`**) και ένα όρισμα (**`/usr/bin`**). Η εκτέλεση της εντολής παράγει μια λεπτομερή λίστα με τα περιεχόμενα του καταλόγου `/usr/bin`.

1.5 Είσοδος στο λειτουργικό σύστημα με χρήση του γραφικού περιβάλλοντος και εκτέλεση εντολών από ένα πρόγραμμα τερματικού μέσω του φλοιού

Το Red Hat Enterprise Linux αποτελεί εξαίρεση στις διανομές Linux για servers, με την έννοια ότι εγκαθιστά και γραφικό περιβάλλον, πράγμα ασυνήθιστο για Linux servers. Η γραφική κονσόλα εισόδου τρέχει στην πρώτη εικονική κονσόλα (tty1).

Σημείωση: στις διανομές Red Hat Enterprise Linux 5 και παλιότερες, όπως και στην συντριπτική πλειοψηφία διανομών Linux, η γραφική κονσόλα εισόδου εμφανίζεται στην εικονική κονσόλα επτά (tty7), ενώ οι εικονικές κονσόλες 1 έως 6 (tty1 – tty6) χρησιμοποιούνται για τερματικά κειμένου. Επίσης, αν η εγκατάσταση του Red Hat έχει γίνει χωρίς γραφικό περιβάλλον, τότε και η εικονική κονσόλα 1 (tty1) είναι κονσόλα κειμένου.

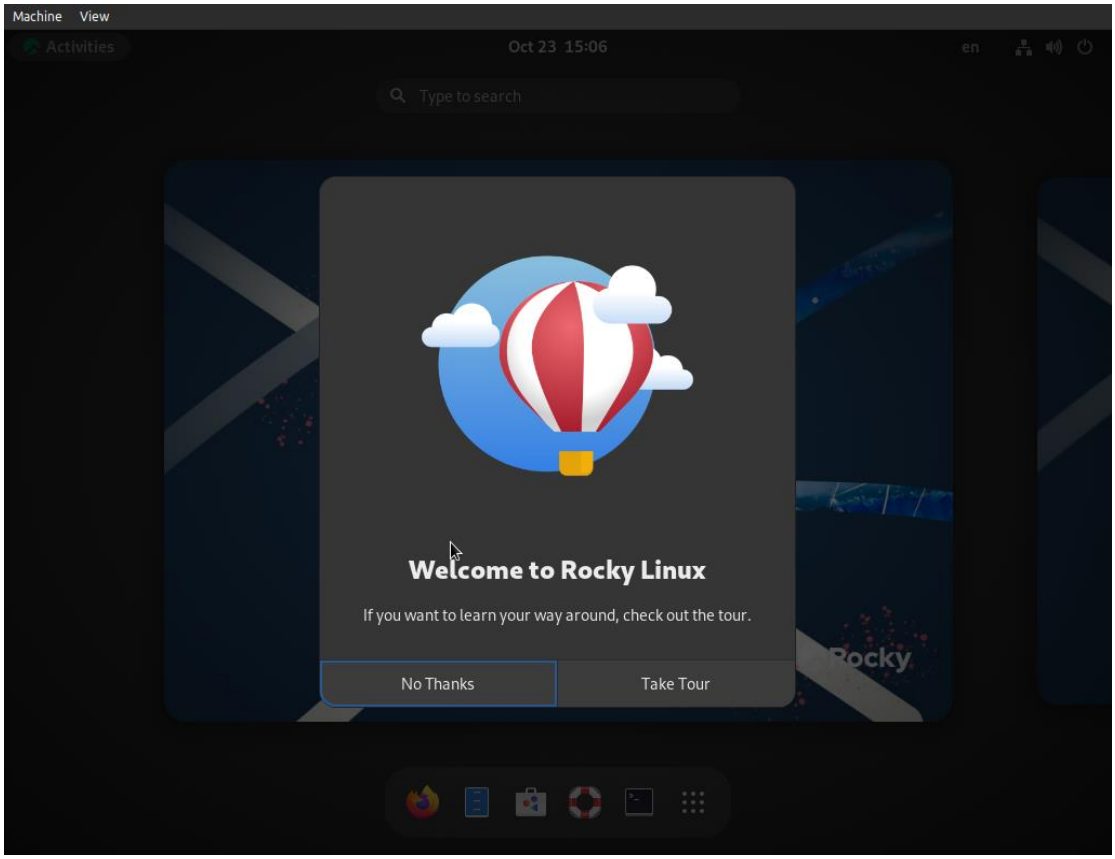
Με το Rocky Linux εκγαταστήσαμε το γραφικό περιβάλλον, οπότε όταν ολοκληρωθεί η εκκίνηση της εικονικής μηχανής βλέπουμε κάτι σαν αυτό:



Εικόνα 31 Οθόνη γραφικής εισόδου χρήστη του Rocky Linux

Κανονικά το όνομα που εμφανίζεται πρέπει να είναι αυτό του χρήστη που έχουμε δημιουργήσει (εικόνες 27 και 28). Πατώντας στο όνομα, μας ζητείται ο κωδικός που έχουμε ορίσει για το χρήστη. Μόλις τον εισάγουμε σωστά, μπαίνουμε στο γραφικό περιβάλλον του χρήστη μας για πρώτη φορά. Εμφανίζεται και μια οθόνη

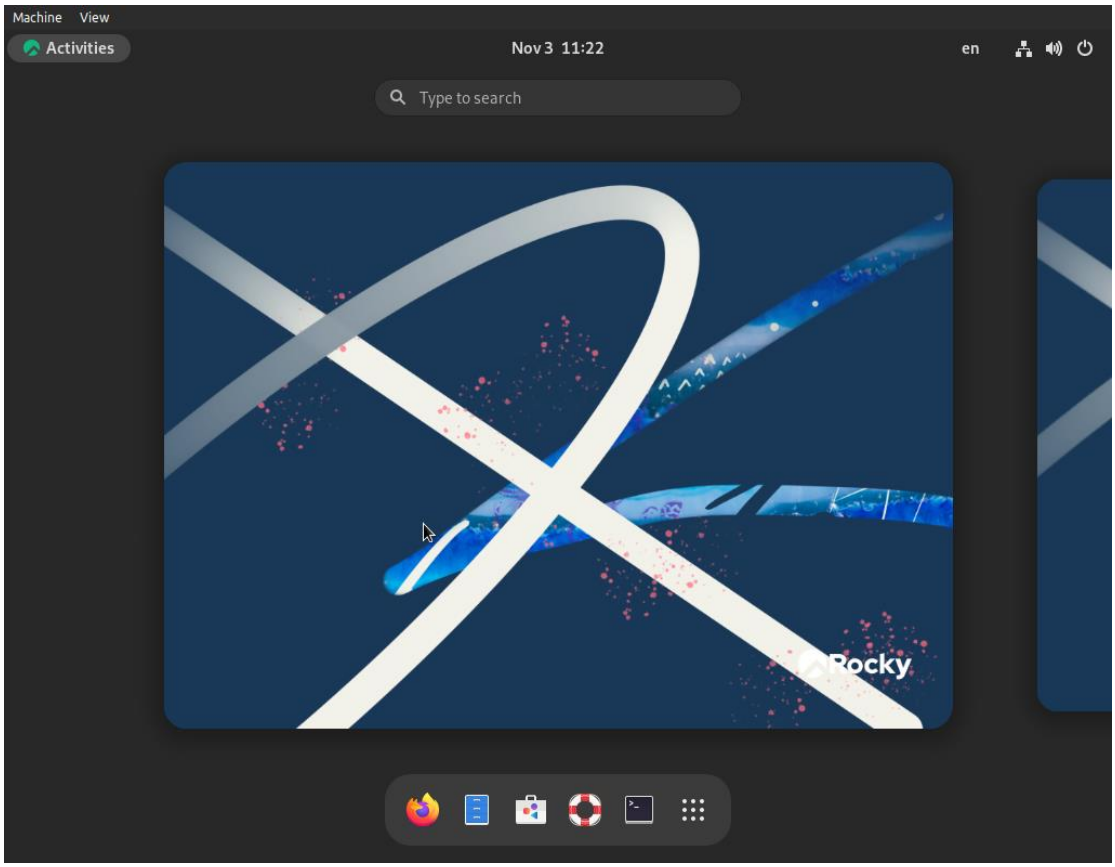
καλωσορίσματος με χρήσιμες πληροφορίες σχετικά με τα πρώτα βήματα για άτομα που είναι νέα στο Rocky Linux.



Εικόνα 32 Οθόνη καλωσορίσματος

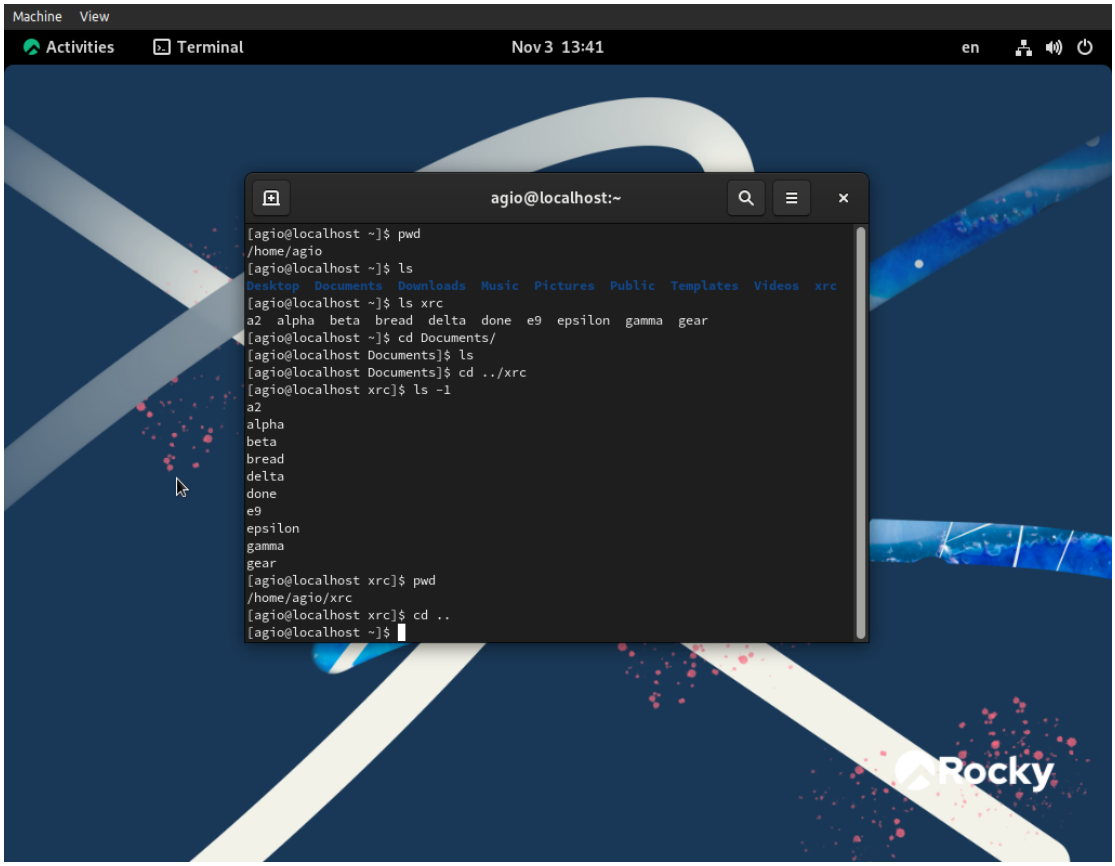
Αν πατήσουμε το εικονίδιο του τερματικού (δεύτερο από δεξιά στη μπάρα στο κάτω μέρος της οθόνης), εκκινεί ένα πρόγραμμα τερματικού με το φλοιό που έχουμε επιλέξει. Σε αυτό μπορούμε να δώσουμε εντολές ακριβώς όπως και στην εικονική κονσόλα.

Αν δε φαίνεται η μπάρα στο κάτω μέρος της οθόνης, αρκεί να πατήσουμε την επιλογή «Activities» στο πάνω αριστερό άκρο και η μπάρα θα εμφανιστεί.



Εικόνα 33 Εμφάνιση μπάρας

Εδώ βλέπουμε το τερματικό να έχει εμφανιστεί και να έχουμε δώσει ήδη κάποιες εντολές.



Εικόνα 34 Το τερματικό

Η εντολή **pwd** (από το «print working directory») τυπώνει τον τρέχοντα κατάλογο, αυτόν που βρισκόμαστε. Βλέπουμε ότι είμαστε στον αρχικό μας κατάλογο (home directory), πράγμα λογικό, αφού μόλις ξεκίνησε η εφαρμογή του τερματικού.

Η εντολή **ls** τυπώνει μια λίστα με τα περιεχόμενα του τρέχοντα καταλόγου, ενώ η **ls xrc** τυπώνει τα περιεχόμενα του καταλόγου xrc, που είναι υποκατάλογος του αρχικού μας. (Ο xrc κατάλογος δεν υπάρχει αρχικά, τον έχουμε δημιουργήσει, όπως και τα περιεχόμενά του,ώστε να μην είναι εντελώς άδειος ο αρχικός μας κατάλογος)

Με την εντολή **cd Documents** μεταβαίνουμε στον κατάλογο Documents και με τη βοήθεια της **ls** βλέπουμε ότι είναι κενός.

Η **cd ../xrc** μας μεταφέρει στον κατάλογο xrc, όπου η **ls -1** μας δίνει μια λίστα των περιεχομένων του σε μία στήλη. Δίνοντας τώρα ξανά την **pwd**, βλέπουμε ότι βρισκόμαστε στον κατάλογο /home/agio/xrc, όπως είναι το πλήρες μονοπάτι.

Τέλος, μια εντολή **cd ..** (προσοχή στο κενό μεταξύ του «cd» και των δύο τελειών) μας ξαναφέρνει στον παραπάνω κατάλογο, που είναι ο αρχικός μας.

2. Διαχείριση αρχείων από τη γραμμή εντολών – Λήψη Βοήθειας

Μαθησιακοί στόχοι

Οι επιμορφούμενοι θα πρέπει να είναι σε θέση να:

- Δημιουργούν, αντιγράφουν, μετακινούν, διαγράφουν και οργανώνουν αρχεία χρησιμοποιώντας τον φλοιό Bash.
- Επιλύουν προβλήματα χρησιμοποιώντας τοπικά συστήματα λήψης βοήθειας

2.1 Ιεραρχία του συστήματος αρχείων (filesystem)

Για να εργαστούμε αποδοτικά στο λειτουργικό σύστημα Linux θα χρειαστεί να γνωρίσουμε τους καταλόγους που υπάρχουν και τη δομή που αυτοί έχουν. Τα αρχεία και οι κατάλογοι δομούνται και αποθηκεύονται με χρήση του συστήματος αρχείων FHS (File System Hierarchy). Η τελευταία έκδοση του πραγματοποιήθηκε το 2015 (FHS 3.0) και μπορούμε να δούμε μια περιγραφή εκτελώντας την εντολή **`man 7 hierarchy`**.

Το σύστημα αρχείων έχει δενδρική δομή ξεκινώντας από την ρίζα / που βρίσκεται στην κορυφή της ιεραρχίας. Εκτελώντας την εντολή **`tree`** και χρησιμοποιώντας την παράμετρο **d** (εμφανίζει μόνο τους καταλόγους) και την παράμετρο **L** (εμφανίζει συγκεκριμένα επίπεδα υποκαταλόγων) μπορούμε να δούμε τη βασική δομή του όπως παρακάτω:

```
[georgedemo@localhost /]$ tree -d -L 1
.
├── afs
├── bin -> usr/bin
├── boot
├── dev
├── etc
├── home
├── lib -> usr/lib
├── lib64 -> usr/lib64
├── media
├── mnt
├── opt
├── proc
├── root
├── run
├── sbin -> usr/sbin
├── srv
├── sys
├── tmp
├── usr
└── var
```

Κάτω από τον ριζικό κατάλογο βρίσκονται όλοι οι άλλοι κατάλογοι. Έτσι για παράδειγμα ο κατάλογος boot είναι ένας υποκατάλογος και συμβολίζεται ως /boot

όπου ο χαρακτήρας / χρησιμοποιείται ως διαχωριστής. Έτσι ο κατάλογος loader που βρίσκεται μέσα στον boot θα γραφτεί ως /boot/loader.

Στον παρακάτω πίνακα εμφανίζονται οι κυριότεροι κατάλογοι που χρησιμοποιούνται σε ένα σύστημα Linux.

Κατάλογος	Χρήση-Σκοπός
/	Αυτός είναι ο κατάλογος ρίζας του συστήματος αρχείων. Όλοι οι άλλοι κατάλογοι και αρχεία βρίσκονται κάτω από αυτόν.
/boot	Περιέχει τα αρχεία που απαιτούνται για την εκκίνηση του συστήματος, όπως τον πυρήνα του Linux (kernel) και τα αρχεία εκκίνησης (bootloader configuration).
/dev	Περιέχει εικονικά αρχεία που αντιστοιχούν σε συσκευές και περιφερειακά. Τα αρχεία αυτά χρησιμοποιούνται για την επικοινωνία με τις συσκευές.
/etc	Αυτός ο κατάλογος περιέχει τις διαμορφώσεις και τις ρυθμίσεις του συστήματος, όπως τα αρχεία ρυθμίσεων για τον πυρήνα, τις υπηρεσίες, και τις εφαρμογές.
/home	Σε αυτόν τον κατάλογο βρίσκονται οι κατάλογοι των χρηστών, όπου αποθηκεύονται τα αρχεία και οι φάκελοί τους.
/media	Είναι ο κατάλογος όπου συνήθως προσαρτούνται (mount) εξωτερικές συσκευές, όπως USB flash drives ή εξωτερικοί σκληροί δίσκοι, προκειμένου να αποκτήσουμε πρόσβαση σε αυτούς από το λειτουργικό μας σύστημα.
/mnt (Mount)	Οι κατάλογοι σε αυτήν την τοποθεσία χρησιμοποιούνται για την προσωρινή προσάρτηση (mount) εξωτερικών συσκευών.
/opt	Περιέχει πρόσθετο λογισμικό (software) που δεν είναι εγκατεστημένο από τη διανομή Linux, αλλά είναι προσαρτημένο από τον διαχειριστή του συστήματος.
/root	Είναι ο προσωπικός κατάλογος του διαχειριστή (root) του συστήματος. Ο διαχειριστής έχει πλήρη δικαιώματα στο σύστημα.
/run	Περιέχει αρχεία και καταλόγους που πρέπει να είναι διαθέσιμα κατά τη διάρκεια της εκκίνησης του συστήματος, όπως προσωρινές πληροφορίες για τις εκκινούμενες υπηρεσίες.
/srv	Συνήθως χρησιμοποιείται για την αποθήκευση δεδομένων

	που σχετίζονται με υπηρεσίες που παρέχονται από το σύστημα. Αυτά τα δεδομένα μπορεί να περιλαμβάνουν αρχεία που απαιτούνται για τη λειτουργία των υπηρεσιών, όπως ιστοσελίδες, FTP δεδομένα, αποθετήρια ελέγχου εκδόσεων (π.χ. git) κ.α.
/sys	Παρέχει πρόσβαση στις παραμέτρους του πυρήνα του Linux μέσω του συστήματος αρχείων, και χρησιμοποιείται συχνά για τη διαμόρφωση των παραμέτρων του συστήματος.
/tmp	Εδώ αποθηκεύονται προσωρινά αρχεία που αφορούν τρέχουσες διεργασίες.
/usr	Περιέχει τα περισσότερα αρχεία και προγράμματα του συστήματος, συμπεριλαμβανομένων των εκτελέσιμων αρχείων, των βιβλιοθηκών και πολλών άλλων.
/var	Περιλαμβάνει δεδομένα του συστήματος που μεταβάλλονται, όπως αρχεία καταγραφής (logs), προσωρινά αρχεία, δεδομένα βάσεων δεδομένων κ.α.

2.2 Προσδιορισμός της τοποθεσίας των αρχείων

Για να προσδιορίσουμε τη θέση ενός αρχείου ή καταλόγου στο σύστημα διαχείρισης αρχείων του Linux χρησιμοποιούμε το μονοπάτι (path). Κάθε μονοπάτι αποτελείται από μια σειρά καταλόγων που χωρίζονται από τον χαρακτήρα / και οδηγούν στον προορισμό. Για παράδειγμα για να φτάσουμε

Στο Linux, υπάρχουν δύο τρόποι να καθορίσουμε την τοποθεσία ενός αρχείου ή καταλόγου στο σύστημα αρχείων: τα απόλυτα (absolute) μονοπάτια και τα σχετικά (relative) μονοπάτια.

Απόλυτα μονοπάτια

Ένα απόλυτο μονοπάτι (absolute path) στο Linux είναι μια διαδρομή προς ένα αρχείο ή έναν κατάλογο που ξεκινά από τη ρίζα του συστήματος αρχείων (/). Αυτή η διαδρομή προσδιορίζει ακριβώς πού βρίσκεται το στοιχείο στο σύστημα. Τα απόλυτα μονοπάτια είναι ανεξάρτητα από τον τρέχοντα κατάλογο και μπορούν να χρησιμοποιηθούν για αναφορά σε στοιχεία σε οποιονδήποτε κατάλογο στο σύστημα αρχείων. Ακολουθούν τρία παραδείγματα απόλυτων μονοπατιών:

1. Το αρχείο **`/etc/passwd`:** Αυτό το απόλυτο μονοπάτι αναφέρεται στο αρχείο **`passwd`** που περιέχει πληροφορίες για τους χρήστες του συστήματος. Ξεκινά από τη ρίζα του συστήματος αρχείων (/) και συνεχίζει με το όνομα του καταλόγου (**`etc`**) και το όνομα του αρχείου (**`passwd`**).

Απόλυτο μονοπάτι: **/etc/passwd**

2. Ο κατάλογος χρήστη **/home/georgedemo/Documents:** Αυτό το απόλυτο μονοπάτι αναφέρεται σε έναν κατάλογο που βρίσκεται στον φάκελο χρήστη "georgedemo". Ξεκινά από τη ρίζα του συστήματος αρχείων (/) και συνεχίζει με τα ονόματα των καταλόγων για να φτάσει στον κατάλογο "Documents" εντός του φακέλου χρήστη "georgedemo".

Απόλυτο μονοπάτι: **/home/georgedemo/Documents**

3. Ο κατάλογος εγκατάστασης εφαρμογών **/usr/local/bin:** Αυτό το απόλυτο μονοπάτι αναφέρεται σε έναν κατάλογο όπου συνήθως εγκαθίστανται τοπικές εφαρμογές. Ξεκινά από τη ρίζα του συστήματος αρχείων (/) και συνεχίζει με τα ονόματα των καταλόγων για να φτάσει στον κατάλογο "bin" εντός του "/usr/local".

Απόλυτο μονοπάτι: **/usr/local/bin**

Σχετικά (relative) μονοπάτια.

Ένα σχετικό (relative) μονοπάτι στο Linux είναι μια διαδρομή προς ένα αρχείο ή έναν κατάλογο η οποία βασίζεται στον τρέχοντα κατάλογο. Με άλλα λόγια, δεν αρχίζει από τη ρίζα του συστήματος αρχείων (/), αλλά από τον κατάλογο στον οποίο βρισκόμαστε (τρέχων κατάλογος). Τα σχετικά μονοπάτια είναι χρήσιμα για αναφορά σε στοιχεία που βρίσκονται στον τρέχοντα κατάλογο ή σε υποκαταλόγους του. Ας δούμε όμως τέσσερα παραδείγματα σχετικών μονοπατιών:

1. **Αρχείο στον τρέχοντα κατάλογο:** Αν βρισκόμαστε στον κατάλογο **/home/georgedemo/Documents** και θέλουμε να αναφερθούμε σε ένα αρχείο με το όνομα **reportSeptember.txt** που βρίσκεται στον ίδιο κατάλογο, το σχετικό μονοπάτι είναι απλά το όνομα του αρχείου.

Παράδειγμα: **reportSeptember.txt**

2. **Υποκατάλογος σε σχέση με τον τρέχοντα κατάλογο:** Αν βρισκόμαστε στον κατάλογο **/home/georgedemo** και θέλουμε να αναφερθούμε σε έναν υποκατάλογο με το όνομα **photos** που βρίσκεται εντός του τρέχοντος καταλόγου, το σχετικό μονοπάτι είναι απλά το όνομα του υποκαταλόγου.

Παράδειγμα: **photos**

3. **Γονικός κατάλογος:** Αν βρισκόμαστε στον κατάλογο **/home/georgedemo/Documents/reports** και θέλουμε να αναφερθούμε στο γονικό κατάλογο **/home/ georgedemo/Documents**, το σχετικό μονοπάτι είναι **..** (δύο τελείες). Αυτό αναφέρεται στον κατάλογο που βρίσκεται ένα επίπεδο από τον τρέχοντα κατάλογο.

Παράδειγμα: `..`

4. **Προ-Γονικός κατάλογος:** Αν βρισκόμαστε στον κατάλογο **`/home/georgedemo/Documents/reports`** και θέλουμε να αναφερθούμε στο γονικό κατάλογο **`/home/georgedemo/`**, το σχετικό μονοπάτι είναι **`../..`** (δύο τελείες). Αυτό αναφέρεται στον κατάλογο που βρίσκεται δύο επίπεδα από τον τρέχοντα κατάλογο.

Παράδειγμα: **`../..`**

Ο τρέχων κατάλογος ("working directory" ή "current working directory") στο Linux είναι ο κατάλογος στον οποίο βρισκόμαστε σε μια δεδομένη στιγμή καθώς εκτελούμε εντολές στο τερματικό μας. Αυτός ο κατάλογος καθορίζει τη θέση στο σύστημα αρχείων μας και επηρεάζει τον τρόπο με τον οποίο αναφερόμαστε σε αρχεία και καταλόγους.

Μπορούμε να ελέγξουμε το τρέχοντα κατάλογο μας χρησιμοποιώντας την εντολή **`pwd`** (print working directory). Εκτελώντας αυτήν την εντολή, το σύστημά μας θα εμφανίσει το απόλυτο μονοπάτι (absolute path) προς τον τρέχοντα κατάλογο.

Ενδεικτικά, αν εκτελέσουμε την εντολή **`pwd`** και λάβουμε ως απάντηση: **`/home/georgedemo/Documents`**, αυτό σημαίνει ότι ο τρέχων κατάλογος είναι ο **`/home/user/Documents`**.

Ο τρέχων κατάλογος είναι σημαντικός γιατί όταν αναφερόμαστε σε αρχεία ή καταλόγους χρησιμοποιώντας σχετικά μονοπάτια, αυτά τα μονοπάτια θεωρούνται ως σχετικά με τον τρέχοντα κατάλογο. Ανάλογα με τον τρέχοντα κατάλογο, τα ίδια σχετικά μονοπάτια μπορούν να αναφέρονται σε διαφορετικά αρχεία ή καταλόγους.

Προβολή καταλόγων και αρχείων

Η εντολή **`ls`** χρησιμοποιείται στο Linux και σε άλλα UNIX παρόμοια λειτουργικά συστήματα για την εμφάνιση του περιεχομένου ενός καταλόγου. Είναι ένα από τα πιο βασικά εργαλεία για τη διαχείριση αρχείων και καταλόγων. Ορισμένες από τις κύριες παραμέτρους και επιλογές της **`ls`** περιλαμβάνουν:

1. **`-l` (long format):** Αυτή η παράμετρος εμφανίζει τις λεπτομέρειες για κάθε αρχείο ή κατάλογο, συμπεριλαμβανομένων δικαιωμάτων πρόσβασης, ιδιοκτησίας, μεγέθους, χρόνου τροποποίησης και ονόματος.

Παράδειγμα: **`ls -l`**

2. **`-a` (all):** Αυτή η παράμετρος εμφανίζει όλα τα αρχεία και τους καταλόγους, συμπεριλαμβανομένων των κρυφών αρχείων που ξεκινούν με την τελεία (**.**).

Παράδειγμα: **`ls -a`**

3. **`-h` (human-readable):** Αυτή η παράμετρος εμφανίζει το μέγεθος των αρχείων σε μια ανθρώπινα αναγνώσιμη μορφή, όπως "K" για Kilobyte ή "M" για Megabyte.

Παράδειγμα: **`ls -h`**

4. **`-t` (sort by modification time):** Αυτή η παράμετρος ταξινομεί τα αρχεία κατά τον χρόνο τροποποίησης τους, όπου τα πιο πρόσφατα αρχεία εμφανίζονται πρώτα.

Παράδειγμα: **`ls -t`**

5. **`-R` (recursive):** Αυτή η παράμετρος εμφανίζει το περιεχόμενο των υποκαταλόγων αναδρομικά, δηλαδή εμφανίζει τα αρχεία και τους υποκαταλόγους σε όλα τα επίπεδα καταλόγων.

Παράδειγμα: **`ls -R`**

6. **`ll` (ή `ls -l`):** Η εντολή ll δεν είναι προεπιλεγμένη στα περισσότερα συστήματα, αλλά συνήθως χρησιμοποιείται ως συντόμευση (alias) για την εκτέλεση της **`ls -l`**. Αυτή η εντολή εμφανίζει τη λίστα των αρχείων και καταλόγων με λεπτομερείς πληροφορίες, όπως δικαιώματα πρόσβασης, ιδιοκτησία, μέγεθος, χρόνο τροποποίησης και άλλα.

Παράδειγμα: **`ll`** ή **`ls -l`**

7. **`ls -F`:** Η εντολή **`ls -F`** εμφανίζει τη λίστα των αρχείων και καταλόγων, όμως προσθέτει ειδικούς χαρακτήρες στα ονόματα των αρχείων για να δείξει τον τύπο του κάθε στοιχείου. Για παράδειγμα, οι κατάλογοι τελειώνουν με την κάθετο /, τα εκτελέσιμα αρχεία επισημαίνονται με *, και τα συμβολικά συνδέσμου (symlinks) με @.

Παράδειγμα: **`ls -F`**

Μετακίνηση μεταξύ των καταλόγων

Η εντολή **`cd`** (Change Directory) χρησιμοποιείται για την αλλαγή του τρέχοντος καταλόγου (working directory) στο Linux και σε άλλα UNIX-παρόμοια λειτουργικά συστήματα. Μπορεί να χρησιμοποιηθεί με διάφορες παραμέτρους και επιλογές. Εδώ είναι μερικές από τις βασικές παραμέτρους και επιλογές της **`cd`**:

1. **Χωρίς παραμέτρους:** Χωρίς παραμέτρους, η εντολή **`cd`** απλά επιστρέφει στον αρχικό κατάλογο του χρήστη. Αυτό είναι χρήσιμο όταν θέλουμε να επιστρέψουμε στον αρχικό κατάλογο από οποιονδήποτε κατάλογο βρισκόμαστε.

Παράδειγμα: **`cd`**

2. **Με απόλυτο μονοπάτι:** Μπορούμε να χρησιμοποιήσουμε ένα απόλυτο μονοπάτι για να μεταβούμε απευθείας σε έναν συγκεκριμένο κατάλογο.

Παράδειγμα: **`cd /home/georgedemo/Documents`**

3. **Με σχετικό μονοπάτι:** Μπορείτε να χρησιμοποιήσετε ένα σχετικό μονοπάτι για να μετακινηθείτε σε έναν κατάλογο βάσει της τρέχουσας θέσης σας.

Παράδειγμα: **`cd Documents`**

4. **Με την παράμετρο -:** Χρησιμοποιώντας την παράμετρο -, μπορούμε να επιστρέψουμε στον προηγούμενο κατάλογο που βρισκόμασταν.

Παράδειγμα: **`cd -`**

5. **Με την παράμετρο `..`:** Η παράμετρος **..** αναφέρεται στον γονικό κατάλογο (parent directory) του τρέχοντος καταλόγου. Χρησιμοποιώντας αυτήν την επιλογή, μπορείτε να μεταβούμε ένα επίπεδο πίσω στον κατάλογο.

Παράδειγμα: **`cd ..`**

Δημιουργία αρχείων

Για να δημιουργήσουμε κενά αρχεία ή να ανανεώσουμε (ενημερώσουμε) τις χρονοσφραγίδες ενός αρχείου (χρόνος πρόσβασης, χρόνος τροποποίησης) μπορούμε να χρησιμοποιήσουμε την εντολή touch. Μερικές συνηθισμένες χρήσεις της είναι:

1. **Δημιουργία κενών αρχείων:** Για να δημιουργήσουμε ένα νέο κενό αρχείο, μπορούμε να εκτελέσουμε την εντολή **`touch`** ακολουθούμενη από το όνομα του επιθυμητού αρχείου. Αν το αρχείο δεν υπάρχει, το **`touch`** θα το δημιουργήσει.

Παράδειγμα: **`touch myfile.txt`**

2. **Ενημέρωση σφραγίδων χρόνου:** Αν το αρχείο υπάρχει ήδη, το touch μπορεί να χρησιμοποιηθεί για να ενημερώσει τις σφραγίδες χρόνου πρόσβασης και τροποποίησης του αρχείου χωρίς να αλλάξει το περιεχόμενό του. Αυτό μπορεί να είναι χρήσιμο για τον προσδιορισμό πότε χρησιμοποιήθηκε ή τροποποιήθηκε τελευταία ένα αρχείο.

Παράδειγμα: **`touch existingfile.txt`**

3. **Δημιουργία πολλαπλών αρχείων:** Μπορούμε να δημιουργήσουμε πολλά αρχεία ταυτόχρονα προσθέτοντας πολλά ονόματα αρχείων ως ορίσματα στην εντολή touch.

Παράδειγμα: **touch athens.txt rome.txt madrid.txt**

Παραδείγματα χρήσης

Αρχικά εκτελούμε την εντολή **pwd** για να εξετάσουμε τον τρέχοντα κατάλογο. Εκτελώντας την εντολή **ls** βλέπουμε συνοπτικά τα περιεχόμενα του καταλόγου και στη συνέχεια εκτελούμε την **ls -l** βλέποντας έτσι περισσότερες λεπτομέρειες για τα αρχεία και τους φακέλους που περιέχονται. Με την εντολή **ls -lh** βλέπουμε τα μεγέθη των αρχείων σε πιο κατανοητή μορφή ενώ με την εντολή **ls -a** εμφανίζονται και τυχόν κρυφά αρχεία.

```
georgedemo@localhost ~]$ pwd
/home/georgedemo
[georgedemo@localhost ~]$ ls
date.pdf  Documents  Music     Public     Videos
Desktop   Downloads  Pictures  Templates
[georgedemo@localhost ~]$ ls -l
total 20
-rw-r--r--. 1 georgedemo georgedemo 16759 Sep 20 11:35 date.pdf
drwxr-xr-x. 2 georgedemo georgedemo     6 Jul 14 08:56 Desktop
drwxr-xr-x. 2 georgedemo georgedemo     6 Jul 14 08:56 Documents
drwxr-xr-x. 2 georgedemo georgedemo    69 Sep 11 13:21 Downloads
drwxr-xr-x. 2 georgedemo georgedemo     6 Jul 14 08:56 Music
drwxr-xr-x. 2 georgedemo georgedemo     6 Jul 14 08:56 Pictures
drwxr-xr-x. 2 georgedemo georgedemo     6 Jul 14 08:56 Public
drwxr-xr-x. 2 georgedemo georgedemo     6 Jul 14 08:56 Templates
drwxr-xr-x. 2 georgedemo georgedemo     6 Jul 14 08:56 Videos

[georgedemo@localhost ~]$ ls -lh
total 20K
-rw-r--r--. 1 georgedemo georgedemo 17K Sep 20 11:35 date.pdf
drwxr-xr-x. 2 georgedemo georgedemo   6 Jul 14 08:56 Desktop
drwxr-xr-x. 2 georgedemo georgedemo   6 Jul 14 08:56 Documents
drwxr-xr-x. 2 georgedemo georgedemo  69 Sep 11 13:21 Downloads
drwxr-xr-x. 2 georgedemo georgedemo   6 Jul 14 08:56 Music
drwxr-xr-x. 2 georgedemo georgedemo   6 Jul 14 08:56 Pictures
drwxr-xr-x. 2 georgedemo georgedemo   6 Jul 14 08:56 Public
drwxr-xr-x. 2 georgedemo georgedemo   6 Jul 14 08:56 Templates
drwxr-xr-x. 2 georgedemo georgedemo   6 Jul 14 08:56 Videos

georgedemo@localhost ~]$ ls -a
.               .cache     .lesshst  Templates
..              .config    .local    .vboxclient-clipboard.pid
.bash_history   date.pdf   .mozilla  .vboxclient-draganddrop.pid
.bash_logout    Desktop    Music     .vboxclient-seamless.pid
.bash_profile     Documents    Pictures     .vboxclient-vmsvga-session-
tty2.pid
.bashrc         Downloads  Public    Videos
```

Εισάγοντας την εντολή **cd Music** μετακινούμαστε στον φάκελο **Music** και εμφανίζονται το τρέχων μονοπάτι. Δημιουργούμε τρία αρχεία με χρήση της εντολής touch και εισάγοντας την εντολή **cd** επιστρέφουμε στον μητρικό μας κατάλογο. Τέλος εκτελούμε την εντολή **ls -R** με διαφορετικά ορίσματα για να δούμε αναδρομικά τι περιέχουν οι κατάλογοι που προσδιορίζουμε ως παραμέτρους.

```
georgedemo@localhost ~]$ cd Music
[georgedemo@localhost Music]$ pwd
/home/georgedemo/Music
[georgedemo@localhost Music]$ touch Xarxakos_Mana_Mou_Ellas.mp3
[georgedemo@localhost Music]$ touch Xarxakos_Lola.mp3
[georgedemo@localhost Music]$ touch ../Documents/syllabus.pdf
[georgedemo@localhost Music]$ cd
[georgedemo@localhost ~]$ pwd
/home/georgedemo
[georgedemo@localhost ~]$ ls -R Music
Music:
Xarxakos_Lola.mp3  Xarxakos_Mana_Mou_Ellas.mp3
[georgedemo@localhost ~]$ ls -lR Music
Music:
total 0
-rw-r--r--. 1 georgedemo georgedemo 0 Oct  2 16:29 Xarxakos_Lola.mp3
-rw-r--r--.   1   georgedemo   georgedemo   0   Oct      2   16:29
Xarxakos_Mana_Mou_Ellas.mp3
[georgedemo@localhost ~]$ ls -lR Documents
Documents:
total 0
-rw-r--r--. 1 georgedemo georgedemo 0 Oct  2 16:30 syllabus.pdf

[georgedemo@localhost ~]$ ls -R
.:
date.pdf  Documents  Music     Public     Videos
Desktop   Downloads  Pictures  Templates

./Desktop:

./Documents:
syllabus.pdf

./Downloads:
date.pdf  date.ps  regulations-msc-2017.pdf

./Music:
Xarxakos_Lola.mp3  Xarxakos_Mana_Mou_Ellas.mp3

./Pictures:

./Public:

./Templates:

./Videos:
```

Προσοχή: Όταν δίνουμε την εντολή **`ls -R`** για να εμφανίσουμε τα περιεχόμενα ενός καταλόγου και όλων των υποκαταλόγων του ο όγκος των πληροφοριών μπορεί να είναι αρκετά μεγάλος ανάλογα με το βάθος που έχουν οι υποκατάλογοι. Για αυτό το λόγο θα χρειαστεί να φιλτράρουμε την παραγόμενη έξοδο π.χ. μέσω της εντολής **`grep`**.

2.3 Δημιουργία, αντιγραφή, μετακίνηση και διαγραφή αρχείων και καταλόγων.

Για να εργαστούμε αποτελεσματικά με τα αρχεία και τους καταλόγους σε ένα σύστημα Linux θα χρειαστεί να μπορούμε να δημιουργούμε και να διαγράφουμε αρχεία και καταλόγους αλλά και να αντιγράφουμε ή/και μετακινούμε αλλάζοντας τη δενδρική δομή τους.

Δημιουργία καταλόγων

Η εντολή mkdir στο Linux χρησιμοποιείται για τη δημιουργία νέων καταλόγων (φακέλων). Η εντολή μπορεί να δημιουργήσει πολλούς καταλόγους ταυτόχρονα και να ορίσει τα δικαιώματα πρόσβασης σε αυτούς τους καταλόγους. Είναι σημαντικό να σημειωθεί ότι ο χρήστης που εκτελεί αυτήν την εντολή πρέπει να έχει επαρκή δικαιώματα για να δημιουργήσει έναν κατάλογο στον γονικό κατάλογο, διαφορετικά μπορεί να λάβει ένα μήνυμα σφάλματος "άρνησης πρόσβασης" (permission denied).

Μερικοί τρόποι χρήσης της εντολής mkdir είναι οι παρακάτω:

Δημιουργία ενός απλού καταλόγου:

- Για να δημιουργήσουμε έναν νέο κατάλογο στον τρέχοντα κατάλογο, απλώς χρησιμοποιούμε την εντολή **mkdir** ακολουθούμενη από το όνομα του καταλόγου που επιθυμούμε να δημιουργήσουμε.

  Παράδειγμα: **`mkdir mystuff`**

  Ο φάκελος mystuff δημιουργείται μέσα στον τρέχοντα κατάλογο. Αν επιθυμούμε να δημιουργήσουμε φάκελο εντός κάποιου άλλου φακέλου μπορούμε να τον δημιουργήσουμε χρησιμοποιώντας το απόλυτο μονοπάτι:

  π.χ. **`mkdir /home/georgedemo/Music/mystuff`**

  ή το σχετικό μονοπάτι (ανάλογα που βρισκόμαστε):

  π.χ. **`mkdir ../Music/mystuff`**

Δημιουργία πολλαπλών φακέλων ταυτόχρονα:

- Μπορούμε να δημιουργήσουμε πολλούς φακέλους ταυτόχρονα παρέχοντας πολλά ονόματα φακέλων ως ορίσματα στην εντολή mkdir.

  Παράδειγμα: **`mkdir MyStuff MyVacations MyDreams`**

Δημιουργία φακέλου με πολλαπλά επίπεδα:

- Η δημιουργία ενός καταλόγου (ή πολλών καταλόγων) με υποκαταλόγους (nested directories) σε μία μόνο εντολή, χωρίς να απαιτείται η προϋπόθεση ύπαρξης των ενδιάμεσων καταλόγων μπορεί να επιτευχθεί με χρήση της

εντολής **mkdir -p**. Με άλλα λόγια, με αυτό τον τρόπο μπορεί να δημιουργηθεί μια δομή καταλόγων με πολλά επίπεδα ακόμη και αν αυτά τα επίπεδα δεν υπάρχουν ακόμη.

Παράδειγμα: **mkdir –p MyStuff/MyVacations/Rodos**

Έτσι, αν θέλουμε να δημιουργήσουμε το φάκελο **Rodos** και οι δύο άλλοι φάκελοι (**MyStuff** και MyVacations) δεν υπάρχουν εκ των προτέρων, αυτοί θα δημιουργηθούν χωρίς να εμφανιστεί στην οθόνη κάποιο λάθος.

Αντιγραφή αρχείων

Με την εντολή cp (από το "copy") στο Linux μπορούμε να δημιουργήσουμε αντίγραφα αρχείων και καταλόγων από μια τοποθεσία σε μια άλλη. Η χρήση της εντολής cp μας δίνει τη δυνατότητα να ορίσουμε τα αρχεία που θέλουμε να αντιγράψουμε καθώς και τον προορισμό όπου αυτά θα αντιγραφούν.

Έτσι μερικές χρήσεις της είναι οι παρακάτω:

1. **Αντίγραφο με διαφορετικό όνομα:** Μπορείτε να αντιγράψουμε ένα αρχείο σε μια νέα τοποθεσία και να του δώσουμε διαφορετικό όνομα:

   cp Jazz1.mp3 album1.mp3

   σε αυτή την περίπτωση δημιουργείται ένα αντίγραφο του αρχείου Jazz1.mp3 με όνομα album1.mp3

   cp Jazz1.mp3 myJazzAlbum/

   ενώ σε αυτή την περίπτωση δημιουργείται ένα αντίγραφο του αρχείου **Jazz1.mp3** αλλά τοποθετείται μέσα στον κατάλογο **myJazzAlbum/**

   Προσοχή: Αν κατά την αντιγραφή ενός αρχείου σε έναν κατάλογο ο κατάλογος δεν υπάρχει, τότε θα δημιουργηθεί ένα αρχείο με το όνομα του προτεινόμενου καταλόγου. Για αυτό το λόγο χρησιμοποιούμε το σύμβολο / στο τέλος του ονόματος του προτεινόμενου καταλόγου, δηλαδή **cp Jazz1.mp3 myJazzAlbum/** ώστε αν δεν υπάρχει ο φάκελος myJazzAlbum να λάβουμε αντίστοιχο μήνυμα λάθους.

2. **Αντίγραφο πολλών αρχείων σε έναν κατάλογο:** Μπορούμε να αντιγράψουμε πολλά αρχεία σε έναν κατάλογο:

   cp Jazz1.mp3 Jazz2.mp3 Jazz3.mp3 /home/George/myJazzAlbum/

με αυτή την εντολή τα αρχεία **Jazz1.mp3**, **Jazz2.mp3**, και **Jazz3.mp3** αντιγράφονται στον κατάλογο **/home/George/myJazzAlbum/**

3. **Αντίγραφο αρχείων και καταλόγων, συμπεριλαμβανομένου του περιεχομένου όλων των υποκαταλόγων:** Με χρήση της παραμέτρου '-r' ή '-R' (recursive – αναδρομικά) μπορούμε να πραγματοποιήσουμε αναδρομική αντιγραφή, δηλαδή να συμπεριληφθούν όλα τα αρχεία και οι υποκατάλογοι που βρίσκονται μέσα στον αρχικό κατάλογο.

   Έτσι αν έχουμε τον αρχικό κατάλογο "**Music**" με τα ακόλουθα αρχεία και υποκαταλόγους:

```
Music/
    ├── EltonJohn.mp3
    ├── BobbyDarin.mp3
    └── Jazz/
        ├── StanGetz.mp3
        └── AlbertoMontoya.mp3
```

   Και επιθυμούμε να αντιγράψουμε τα περιεχόμενα του "**Music**" σε έναν νέο κατάλογο "**BackupMusic**" τότε θα χρησιμοποιήσουμε την εντολή '**cp –r**' ως εξής:

```
cp –r Music/ BackupMusic/
```

Μετακίνηση αρχείων

Η εντολή **mv** στο Linux χρησιμοποιείται για τη μετακίνηση αρχείων και καταλόγων από μια τοποθεσία σε μια άλλη, αλλά δεν περιορίζεται μόνο στη μετακίνηση. Μπορεί επίσης να χρησιμοποιηθεί για τη μετονομασία αρχείων και καταλόγων, αλλάζοντας το όνομά τους χωρίς να αλλάξει η τοποθεσία τους.

Κατά τη χρήση της εντολής **mv**, υπάρχουν δύο βασικές λειτουργίες:

1. **Μετακίνηση αρχείων/καταλόγων**: Χρησιμοποιώντας την εντολή **mv** μπορούμε να μετακινήσουμε ένα αρχείο ή έναν κατάλογο από μια τοποθεσία σε μια άλλη. Αυτό σημαίνει ότι το αρχείο ή ο κατάλογος διατηρεί το όνομά του, αλλά αλλάζει την τοποθεσία του στο σύστημα αρχείων.

   ```
   mv Jazz1.mp3 MyAlbums/
   ```

σε αυτή την περίπτωση το αρχείο **`Jazz1.mp3`** μετακινείται στον κατάλογο **`MyAlbums`**.

2. **Μετονομασία αρχείων/καταλόγων**: Εκτός από τη μετακίνηση, η εντολή **mv** μπορεί να χρησιμοποιηθεί για τη μετονομασία αρχείων και καταλόγων στην ίδια τοποθεσία. Αυτό σημαίνει ότι το αρχείο ή ο κατάλογος διατηρεί την τοποθεσία του, αλλά αλλάζει το όνομά του.

   ```
   mv Jazz1.mp3 album1.mp3
   ```

   σε αυτή την περίπτωση το αρχείο **`Jazz1.mp3`** μετονομάζεται σε **`album1.mp3`**.

Διαγραφή αρχείων και καταλόγων

Η εντολή **rm** στο Linux χρησιμοποιείται για τη διαγραφή αρχείων και καταλόγων από το σύστημα αρχείων. Είναι μια ισχυρή εντολή, και πρέπει να χρησιμοποιείται με προσοχή, καθώς διαγράφει τα αρχεία οριστικά, χωρίς δυνατότητα ανάκτησης. Μερικές χρήσεις της είναι οι παρακάτω:

1. **Διαγραφή ενός αρχείου**: Για να διαγράψουμε ένα αρχείο με το όνομα "**`jazz.mp3`**", εκτελούμε την εντολή:

   ```
   rm jazz.mp3
   ```

2. **Διαγραφή πολλών αρχείων:** Μπορούμε να διαγράψουμε πολλά αρχεία ταυτόχρονα χρησιμοποιώντας χαρακτήρες αναπλήρωσης (wildcards). Για παράδειγμα, για να διαγράψουμε όλα τα αρχεία με κατάληξη **`".txt`**", εκτελούμε:

   ```
   rm *.txt
   ```

3. **Διαγραφή ενός καταλόγου με το περιεχόμενό του:** Για να διαγράψουμε έναν κατάλογο και όλα τα αρχεία και υποκατάλογους του, μπορούμε να χρησιμοποιήσουμε την επιλογή **`-r`** (αναδρομικά):

   ```
   rm -r MyAlbums/
   ```

4. **Προειδοποίηση πριν από τη διαγραφή:** Για να εμφανίσουμε ένα μήνυμα προειδοποίησης πριν από τη διαγραφή ενός αρχείου, μπορούμε να χρησιμοποιήσουμε την επιλογή -i:

   ```
   rm –i *.txt
   ```

   Αυτό θα ζητήσει επιβεβαίωση πριν από τη διαγραφή του αρχείου.

Προσοχή: Χρησιμοποιούμε την εντολή **rm** με προσοχή, καθώς δεν παρέχει αναίρεση και τα διαγεγραμμένα αρχεία δεν μπορούν να ανακτηθούν. Επιπλέον, αν διαγράψουμε κατά λάθος έναν κατάλογο, θα χαθούν όλα τα αρχεία και οι υποκατάλογοι που περιέχονται σε αυτόν τον κατάλογο.

2.4 Δημιουργία σκληρών και συμβολικών συνδέσμων σε αρχεία (hard and symbolic links)

Στο Linux υπάρχει διαφορετικοί τρόποι για να δημιουργήσουμε ονόματα που αντιστοιχούν (οδηγούν) σε ένα αρχείο. Αυτοί είναι τα "hard links" και τα "soft links" (ή "symbolic links") που επιτρέπουν τη σύνδεση αρχείων και καταλόγων στο σύστημα αρχείων. Και τα δύο είδη συνδέσεων επιτρέπουν την αναφορά σε ένα αρχείο ή κατάλογο από διαφορετικές τοποθεσίες, αλλά λειτουργούν με διαφορετικό τρόπο.

Hard Links

Όταν δημιουργούμε ένα αρχείο στο Linux, παράγεται μια δομή δεδομένων (inode – index node) στο σύστημα αρχείων, που αποθηκεύει ορισμένα χαρακτηριστικά του αρχείου. Τέτοια είναι το όνομα του αρχείου, τα δικαιώματα, ο ιδιοκτήτης του αρχείου καθώς και το σημείο στο δίσκο που αποθηκεύεται το αρχείο. Αυτό το τελευταίο, που συνδέει το inode με το σημείο του δίσκου που ξεκινάνε τα περιεχόμενα του αρχείου, είναι το hard link. Άρα κάθε αρχείο έχει τουλάχιστον ένα hard link.

Τα "hard links" στο Linux είναι ένας τρόπος σύνδεσης αρχείων σε διάφορες θέσεις του συστήματος αρχείων, επιτρέποντας την κοινή χρήση του ίδιου περιεχομένου από πολλές διαδρομές χωρίς να απαιτεί ούτε πολυπλοκότερη αποθήκευση δεδομένων, ούτε περισσότερο χώρο. Όταν δημιουργούμε ένα hard link προς ένα αρχείο, δημιουργούμε ένα νέο όνομα που δείχνει στο ίδιο αρχείο. Αν κάποιος διαγράψει ένα από τα "hard links," το περιεχόμενο δεν χάνεται, αλλά παραμένει προσβάσιμο μέσω των υπολοίπων "hard links" προς το ίδιο αρχείο.

Έτσι με τη χρήση των “hard links” επιτυγχάνουμε τα εξής:

Διαμοιρασμό περιεχομένου: Όλα τα “hard links” προς το ίδιο αρχείο διαμοιράζονται το ίδιο inode (αναγνωριστικό αρχείου). Αυτό σημαίνει ότι το περιεχόμενο των αρχείων είναι ακριβώς το ίδιο.

Κοινή χρήση αποθηκευτικού χώρου: Καθώς τα “hard links” μοιράζονται το ίδιο περιεχόμενο, δεν απαιτείται πρόσθετος αποθηκευτικός χώρος για τα αρχεία. Αυτό τα καθιστά χρήσιμα για την εξοικονόμηση χώρου στο δίσκο.

Διαγραφή αρχείων: Όταν διαγράφουμε ένα από τα “hard links” το περιεχόμενο παραμένει προσβάσιμο μέσω των υπολοίπων “hard links.” Το περιεχόμενο διαγράφεται από το σύστημα αρχείων μόνο όταν δεν υπάρχει κανένα “hard link” προς αυτό.

Μετονομασία: Εφόσον όλα τα “hard links” μοιράζονται το ίδιο inode, μπορούμε να αλλάξουμε το όνομα ενός αρχείου χρησιμοποιώντας ένα “hard link” και αυτή η αλλαγή θα εφαρμοστεί και στα υπόλοιπα “hard links”.

Παράδειγμα

```
ln jazz.mp3 /home/georgedemo/Music/myFavoriteAlbums/ChetBaker.mp3
```

με αυτή την εντολή δημιουργούμε ένα νέο "hard link", δηλαδή ένα νέο όνομα, για το ήδη υπάρχον αρχείο **jazz.mp3**. Το νέο hard link ονομάζεται **ChetBaker.mp3** και βρίσκεται στον κατάλογο **/home/georgedemo/Music/myFavoriteAlbums**

Εκτελώντας στη συνέχεια την εντολή ls –l αμέσως μετά τα δικαιώματα στο αρχείο εμφανίζεται ο αριθμός των hard links που έχει ένα αρχείο (στην περίπτωση μας είναι 2).

```
[georgedemo@localhost ~]$ ls -l jazz.mp3
-rw-r--r--. 2 georgedemo georgedemo 0 Oct 11 13:46 jazz.mp3
```

Επιπλέον για να εξετάσουμε αν δύο αρχεία είναι οδηγούν στο ίδιο σημείο στη μονάδα αποθήκευσης (στο ίδιο σύστημα αρχείων), μπορούμε να χρησιμοποιήσουμε την επιλογή –i που μας δείχνει το inode κάθε ενός. Έτσι έχουμε:

```
[georgedemo@localhost ~]$ ls -il jazz.mp3
34361093 -rw-r--r--. 2 georgedemo georgedemo 0 Oct 11 13:46 jazz.mp3
```

```
[georgedemo@localhost ~]$ ls -il Music/myFavoriteAlbums/ChetBaker.mp3
34361093 -rw-r--r--. 2 georgedemo georgedemo 0 Oct 11 13:46
Music/myFavoriteAlbums/ChetBaker.mp3
```

Και στις δύο περιπτώσεις ο αριθμός Inode είναι ο ίδιος.

Σημείωση (διαγραφή και μετακίνηση)

Τα "hard links" είναι χρήσιμα για τη δημιουργία αντιγράφων αρχείων χωρίς την κατανάλωση πρόσθετου χώρου στο δίσκο, καθώς και για την οργάνωση των αρχείων χωρίς να αλλάζουμε την τοποθεσία τους. Πρέπει, ωστόσο, να προσέχουμε ότι η διαγραφή ενός "hard link" δεν ελευθερώνει τον αποθηκευτικό χώρο παρά μόνο όταν διαγράφονται όλα τα "hard links" προς το ίδιο αρχείο.

```
[georgedemo@localhost ~]$ rm Music/myFavoriteAlbums/ChetBaker.mp3
[georgedemo@localhost ~]$ ls -l jazz.mp3
-rw-r--r--. 1 georgedemo georgedemo 0 Oct 11 13:46 jazz.mp3
```

Εκτελώντας τη διαγραφή του αρχείου **ChetBaker.mp3** και στη συνέχεια εκτελώντας την εντολή **ls –l jazz.mp3** παρατηρούμε ότι αφενός διατηρήθηκε το αρχείο **jazz.mp3** αλλά ότι παράλληλα μειώθηκε ο αριθμός των "hard links" σε **1**.

Τέλος τα hard links μπορούν να μετακινηθούν στο σύστημα αρχείων με χρήση της εντολής mv χωρίς να επηρεαστεί η σύνδεση με το αρχικό inode.

Σημείωση: το γεγονός ότι τα hard links βασίζονται στα i-nodes σημαίνει ότι μπορούν να γίνουν μόνο εντός του ίδιου μέσου ή διαμέρισης (ουσιαστικά του ίδιου υποσυστήματος αρχείων), μια και τα i-nodes είναι μοναδικά μόνο στο ίδιο μέσο αποθήκευσης ή στην ίδια διαμέριση και δεν υπάρχει τρόπος άμεσης αναφοράς σε i-node άλλου μέσου. Επίσης, μπορούν να χρησιμοποιηθούν μόνο για απλά αρχεία. Δεν μπορεί να φτιαχτεί ένα hard link που να δείχνει σε έναν κατάλογο ή σε ένα ειδικό αρχείο. Για αυτές τις περιπτώσεις υπάρχουν τα soft links, που αναλύονται αμέσως παρακάτω.

Soft Links (Symbolic links)

Για να δημιουργήσουμε ένα 'soft link' θα χρησιμοποιήσουμε την εντολή **`ln -s`**. Έτσι δημιουργούμε έναν ειδικό τύπο αρχείου που δείχνει προς ένα άλλο αρχείο ή κατάλογο που ήδη υπάρχει.

π.χ. **`ln -s mynotes.txt ~/myShortcuts/mynotes_link.txt`**

σε αυτή την περίπτωση δημιουργούμε ένα 'soft link' για το αρχείο **`mynotes.txt`** που ήδη υπάρχει και που το ονομάζουμε **`mynotes_links.txt`**

εκτελώντας στη συνέχεια την εντολή:

`ls -l mynotes.txt /~/myShortcuts/mynotes_link.txt`

θα έχουμε την παρακάτω έξοδο:

```
lrwxrwxrwx.  1  georgedemo  georgedemo  11  Oct  13  15:42
/home/georgedemo/myShortcuts/mynotes_link.txt -> mynotes.txt
-rw-r--r--. 1 georgedemo georgedemo  0 Oct 13 15:37 mynotes.txt
```

Εδώ παρατηρούμε ότι ο πρώτος χαρακτήρας είναι το σύμβολο **`l`** που υποδηλώνει ότι το αρχείο είναι ένα 'soft link' και όχι ένα κανονικό αρχείο. Όταν στη συνέχεια εμφανίζεται το όνομα του 'soft link' ακολουθεί το σύμβολο **`->`** που δείχνει στο αρχείο-στόχο **`mynotes.txt`**.

Ιδιαίτερη προσοχή θα πρέπει να δοθεί όταν διαγραφεί το αρχείο **`mynotes.txt`** οπότε και το 'soft link' θα συνεχίσει να υπάρχει αλλά δεν θα δείχνει κάπου. Σε αυτή την περίπτωση αν τυχόν δημιουργηθεί ένα αρχείο με το ίδιο όνομα (**`mynotes.txt`**) τότε το 'soft link' θα δείξει προς το νέο αρχείο.

Τα 'soft links' έχουν μερικά πλεονεκτήματα σε σχέση με τα 'hard links'. Αυτά είναι:

- Μπορούν να οδηγούν σε ένα ειδικό αρχείο ή σε ένα φάκελο
- Μπορούν να οδηγούν σε ένα αρχείο που βρίσκεται σε διαφορετικό σύστημα αρχείων (file system), ακόμα και αφαιρούμενο, π.χ. εξωτερικό δίσκο ή USB flash.

Τα μειονεκτήματά τους είναι τα εξής:

- Αν πάψει να υπάρχει το αντικείμενο στο οποίο δείχνουν (π.χ. το σβήσουμε, το μετακινήσουμε ή αφαιρέσουμε τη συσκευή USB που το περιέχει) το link παραμένει, αλλά δεν ισχύει, τουλάχιστο μέχρι π.χ. να ξανασυνδέσουμε τη συσκευή USB. Αυτό ονομάζεται “dangling link”.
- Μπορεί εύκολα κανείς να δημιουργήσει βρόχους στο σύστημα αρχείων. Αν φτιάξουμε ένα soft link A που να δείχνει στο soft link B, αυτό στο soft link C και το C να δείχνει ξανά στο A, έχουμε έναν τέτοιο βρόχο. Αυτοί δεν είναι καλοί γενικά για το σύστημα αρχείων επειδή μπορεί να οδηγήσουν σε εντολές που θα μπορούσαν να μη τερματίζουν ποτέ (π.χ. μια αναδρομική αναζήτηση για αρχεία) και να προκαλέσουν μεγάλη κατανάλωση πόρων του συστήματος. Αν και το Linux έχει μηχανισμούς που καταλαβαίνουν ότι βρίσκονται σε τέτοιο βρόχο, καλό είναι να αποφεύγονται.

2.5 Αποδοτική εκτέλεση εντολών σε πολλαπλά αρχεία με χρήση ταιριάσματος προτύπων (pattern matching) στον φλοιό Bash

Ένας από τους τρόπους που είναι διαθέσιμοι στο φλοιό Bash για να επεκτείνουμε την εμβέλεια μιας γραμμής εντολής είναι η χρήση *προτύπων* για ταίριασμα ονομάτων, γνωστό και σαν *globbing* ή *wildcards* (wildcard στα αγγλικά είναι ο μπαλαντέρ των παιχνιδιών τράπουλας, που μπορεί να αντικαταστήσει διάφορα φύλλα). Αυτή η δυνατότητα επιτρέπει την ευκολότερη διαχείριση μεγάλου αριθμού αρχείων. Με τη χρήση *μεταχαρακτήρων* που «ανοίγουν» για να ταιριάξουν με ονόματα αρχείων και μονοπατιών που αναζητούμε, οι εντολές μπορούν να εκτελούνται άμεσα πάνω σε ένα προσχεδιασμένο σύνολο αρχείων.

Ταίριασμα προτύπων

Το “globbing” είναι μια λειτουργία που εκτελεί ο φλοιός κατά την ανάλυση στης εντολής και που αναπτύσσει ένα πρότυπο με wildcards σε μια λίστα με ονόματα μονοπατιών που ταιριάζουν σε αυτό. Οι μεταχαρακτήρες αντικαθίστανται από τη λίστα με τα ταιριάσματα πριν την εκτέλεση της εντολής. Στον ακόλουθο πίνακα παρουσιάζονται συνηθισμένοι μεταχαρακτήρες και κλάσεις προτύπων.

Πρότυπο	Ταιριάσματα
*****	Οποιαδήποτε συμβολοσειρά με μηδέν ή περισσότερους χαρακτήρες.
?	Ένας οποιοσδήποτε χαρακτήρας.
[abc…]	Ένας οποιοσδήποτε χαρακτήρας από την κλάση μέσα στις αγκύλες.

[!abc…]	Ένας οποιοσδήποτε χαρακτήρας που δεν περιλαμβάνεται στην κλάση μέσα στις αγκύλες.
[^abc…]	Ένας οποιοσδήποτε χαρακτήρας που δεν περιλαμβάνεται στην κλάση μέσα στις αγκύλες.
[[:alpha:]]	Ένας οποιοσδήποτε αλφαβητικός χαρακτήρας. Ισοδύναμο με το [a-zA-Z].
[[:lower:]]	Ένας οποιοσδήποτε πεζός χαρακτήρας. Ισοδύναμο με το [a-z].
[[:upper:]]	Ένας οποιοσδήποτε κεφαλαίος χαρακτήρας. Ισοδύναμο με το [A-Z].
[[:alnum:]]	Ένας οποιοσδήποτε αλφαβητικός χαρακτήρας ή ψηφίο, ένας αλφαριθμητικός χαρακτήρας. Ισοδύναμο με το [a-zA-Z0-9].
[[:punct:]]	Ένας οποιαδήποτε εκτυπώσιμος χαρακτήρας που δεν είναι χαρακτήρας κενού ή αλφαριθμητικός, ένας χαρακτήρας στίξης.
[[:digit:]]	Ένα οποιοδήποτε ψηφίο, 0-9. Ισοδύναμο με το [0-9].
[[:space:]]	Ένας χαρακτήρας κενού, Tab, Line Feed, Carriage Return, Form Feed

Ας δούμε μερικά παραδείγματα, ώστε να καταλάβουμε καλύτερο τον τρόπο λειτουργίας του globbing.

```
georgedemo@localhost:~$ mkdir xrc

georgedemo@localhost:~$ cd xrc

georgedemo@localhost:~/xrc$ touch alpha beta gamma delta
epsilon abc bread gear done e9

georgedemo@localhost:~/xrc$ ls

abc  alpha  beta  bread  delta  done  e9  epsilon  gamma
gear

georgedemo@localhost:~/xrc$ ls a*

abc  alpha

georgedemo@localhost:~/xrc$ ls *a*

abc  alpha  beta  bread  delta  gamma  gear
```

```
georgedemo@localhost:~/xrc$ ls [ad]*
abc  alpha  delta  done
georgedemo@localhost:~/xrc$ ls ?????
alpha  bread  delta  gamma
georgedemo@localhost:~/xrc$ ls -Al *[[:digit:]]*
-rw-rw-r-- 1 agio agio 0 Νοε  2 15:17 e9
georgedemo@localhost:~/xrc$ ls -1 *{ea,am}*
bread
gamma
gear
georgedemo@localhost:~/xrc$
```

Επέκταση της «~» (tilde - κυματοειδής γραμμή)

Ο χαρακτήρας «~» (κυματοειδής γραμμή, ισπανική περισπωμένη) χρησιμοποιείται για να ταιριάξει το τρέχοντα αρχικό κατάλογο του χρήστη.

Αν ακολουθείται αμέσως από το χαρακτήρα της καθέτου («/»), τότε ταιριάζει με το δικό μας αρχικό κατάλογο.

Αν ακολουθείται από μια συμβολοσειρά χαρακτήρων η οποία ταυτίζεται με το όνομα κάποιου χρήστη, τότε το όλο σύμπλεγμα χαρακτήρων αντικαθίσταται από το απόλυτο μονοπάτι προς τον αρχικό κατάλογο αυτού του χρήστη. Αν δε βρεθεί ταίριασμα με όνομα χρήστη, τότε το σύμπλεγμα παραμένει έτσι.

Επέκταση αγκίστρων

Μπορούμε να τοποθετήσουμε μέσα σε άγκιστρα μια λίστα από αντικείμενα, από τα οποία θα επιλέγεται ένα κάθε φορά.

Μία μορφή αυτής της λίστας είναι συμβολοσειρές χωρισμένες μεταξύ τους με κόμματα χωρίς κενά μεταξύ τους, π.χ. `{ph,eta,amm}`. Από αυτές χρησιμοποιείται κάθε στιγμή είτε η `ph`, είτε η `eta`, είτε η `amm`.

Η άλλη μορφή της λίστας είναι μια έκφραση με διπλή τελεία όπως η `{c..g}`. Αυτή επεκτείνεται στην ακολουθία `c d e f g`, όπου και πάλι χρησιμοποιείται κάθε φορά ένας από τους χαρακτήρες της ακολουθίας.

Οι επεκτάσεις αγκίστρων μπορούν να εγκιβωτιστούν η μία μέσα στην άλλη, δημιουργώντας πιο περίπλοκες εκφράσεις.

Όλα αυτά συνδυάζονται με τις τυχόν συμβολοσειρές που προηγούνται ή ακολουθούν τις επεκτάσεις.

Για να γίνει εμφανές το είδος της επέκτασης που γίνεται, θα χρησιμοποιήσουμε την εντολή `echo` για να δούμε όλους τους συνδυασμούς που παράγονται:

```
John.txt Paul.txt George.txt Ringo.txt
```

```
agio@localhost:~$ echo Beatle_{1..4}.txt
```

```
Beatle_1.txt Beatle_2.txt Beatle_3.txt Beatle_4.txt
```

```
agio@localhost:~$ echo foo_{alpha{1,2,3},beta{4..6},gamma{A..C}}_bar.txt
```

```
foo_alpha1_bar.txt foo_alpha2_bar.txt foo_alpha3_bar.txt foo_beta4_bar.txt
foo_beta5_bar.txt foo_beta6_bar.txt foo_gammaA_bar.txt foo_gammaB_bar.txt
foo_gammaC_bar.txt
```

```
agio@localhost:~$
```

Αυτή η επέκταση μας δίνει και έναν τρόπο εύκολης δημιουργίας αρχείων ή καταλόγων που το όνομά τους ακολουθεί κάποιους επαναληπτικούς κανόνες:

```
agio@localhost:~/xrc$ touch file_{05..10}.txt
```

```
agio@localhost:~/xrc$
```

Επέκταση μεταβλητών

Οι *μεταβλητές φλοιού* (θα τις δούμε αναλυτικά παρακάτω) είναι ένας τρόπος για να αποθηκεύσουμε μια τιμή σε ένα χώρο αποθήκευσης με όνομα. Μας παρέχουν έναν εύκολο τρόπο προσπέλασης και τροποποίησης της τιμής δεδομένων, είτε από τη γραμμή εντολής, είτε σε ένα πρόγραμμα σεναρίου φλοιού.

Η ανάθεση τιμής σε μια μεταβλητή φλοιού μπορεί να γίνει με τον παρακάτω τρόπο (παρατηρήστε την έλλειψη κενών και στις δύο πλευρές του ίσον):

```
agio@localhost:~$ AVARNAME=a_value
```

Η επέκταση της μεταβλητής επιστρέφει την τιμή της και μπορούμε να την εφαρμόσουμε για να χρησιμοποιήσουμε την τιμή της. Η επέκταση της μεταβλητής γίνεται βάζοντας το χαρακτήρα του δολλαρίου ($) αμέσως πριν το όνομα της μεταβλητής. Ο φλοιός βλέποντας το $ ψάχνει να βρει μια μεταβλητή με όνομα το υπόλοιπο της συμβολοσειράς και να την αντικαταστήσει με την τιμή της:

```
agio@localhost:~$ MONTH=June
```

```
agio@localhost:~/xrc$ echo $MONTH
```

```
June
```

```
agio@localhost:~/xrc$ echo ${MONTH}
```

```
June
```

Ο δεύτερος τρόπος (με τις αγκύλες) χρησιμοποιείται για αποφυγή λαθών λόγω άλλων επεκτάσεων φλοιού που μπορεί να υπάρχουν σε μια έκφραση.

Με τις μεταβλητές φλοιού θα ασχοληθούμε αναλυτικά σε παρακάτω κεφάλαιο.

Αντικατάσταση εντολών

Με την αντικατάσταση εντολής μπορεί να πάρει τη θέση μιας εντολής η έξοδος που παράγεται από την εκτέλεσή της. Για να γίνει αυτό, κλείνουμε την εντολή σε παρανθέσεις και βάζουμε σαν πρόθεμα ένα χαρακτήρα δολαρίου ($). Όπως και με τα άγκιστρα, οι αντικαταστάσεις εντολών μπορούν να εγκιβωτιστούν η μία μέσα στην άλλη πολλές φορές.

```
agio@localhost:~/xrc$ echo I am $(whoami).

I am agio.

agio@localhost:~/xrc$ echo I am $(whoami) and I am at $(pwd).

I am agio and I am at /home/agio/xrc.
```

Υπάρχει και μιά παλιότερη εκδοχή της αντικατάστασης εντολής, που χρησιμοποιεί τον χαρακτήρα backtick (`), που βρίσκεται ακριβώς αριστερά από το πλήκτρο «1». Η μορφή της είναι `` `command` `` και δεν προτιμάται πια η χρήση της επειδή ο χαρακτήρας «`» μπορεί εύκολα να μπερδευτεί με το μονό εισαγωγικό, αλλά και επειδή δεν μπορούν να εγκιβωτιστούν η μία μέσα στην άλλη.

Προστασία των ορισμάτων από επέκταση

Υπάρχουν περιπτώσεις που θέλουμε να προστατέψουμε ένα τμήμα της γραμμής εντολής από την επέκταση του φλοιού, π.χ. αν περιέχει χαρακτήρες που έχουν ειδική σημασία για το φλοιό, αλλά θέλουμε να λειτουργήσουν σαν απλοί χαρακτήρες και να μην τους επεκτείνει ο φλοιός. Σε μια τέτοια περίπτωση μπορούμε να χρησιμοποιήσουμε είτε τα *εισαγωγικά*, είτε *χαρακτήρες και συμβολοσειρές διαφυγής*.

Ένας χαρακτήρας διαφυγής στον Bash είναι η ανάποδη κάθετος (\), η οποία προστατεύει τον αμέσως επόμενό της χαρακτήρα από επέκταση. Στη δεύτερη εντολή βλέπουμε το αποτέλεσμα της εφαρμογής του χαρακτήρα διαφυγής στα δύο άγκιστρα:

```
agio@localhost:~$ echo Beatle_{1..4}.txt

Beatle_1.txt Beatle_2.txt Beatle_3.txt Beatle_4.txt

agio@localhost:~$ echo Beatle_\{1..4\}.txt

Beatle_{1..4}.txt
```

Τα μονά και διπλά εισαγωγικά χρησιμοποιούνται για την προστασία από επέκταση των συμβολοσειρών που περικλείουν.

Με τα διπλά εισαγωγικά σταματάει το globbing και η επέκταση του φλοιού, αλλά επιτρέπεται η αντικατάσταση εντολών και η επέκταση μεταβλητών:

```
agio@localhost:~$ myname=$(whoami); echo $myname
```

```
agio
```

```
agio@localhost:~$ echo "My name is ${myname}."
```

```
My name is agio.
```

Με τα μονά εισαγωγικά σταματάει κάθε είδους επέκταση του φλοιού και όλο το κείμενο ερμηνεύεται κυριολεκτικά, χωρίς επεκτάσεις και αντικαταστάσεις.

```
agio@localhost:~$ echo "My name is $(whoami) or ${myname}."
```

```
My name is agio or agio.
```

```
agio@localhost:~$ echo 'My name is $(whoami) or ${myname}.'
```

```
My name is $(whoami) or ${myname}.
```

2.6 Λήψη βοήθειας

Όπως είναι γνωστό το Linux δημιουργήθηκε με βάση το λειτουργικό σύστημα Unix. Οι προγραμματιστές όμως που ανάπτυξαν το Unix δημιούργησαν ένα σύστημα λήψης βοήθειας που ονομάζεται *man pages* (συντομογραφία για το *manual page*).

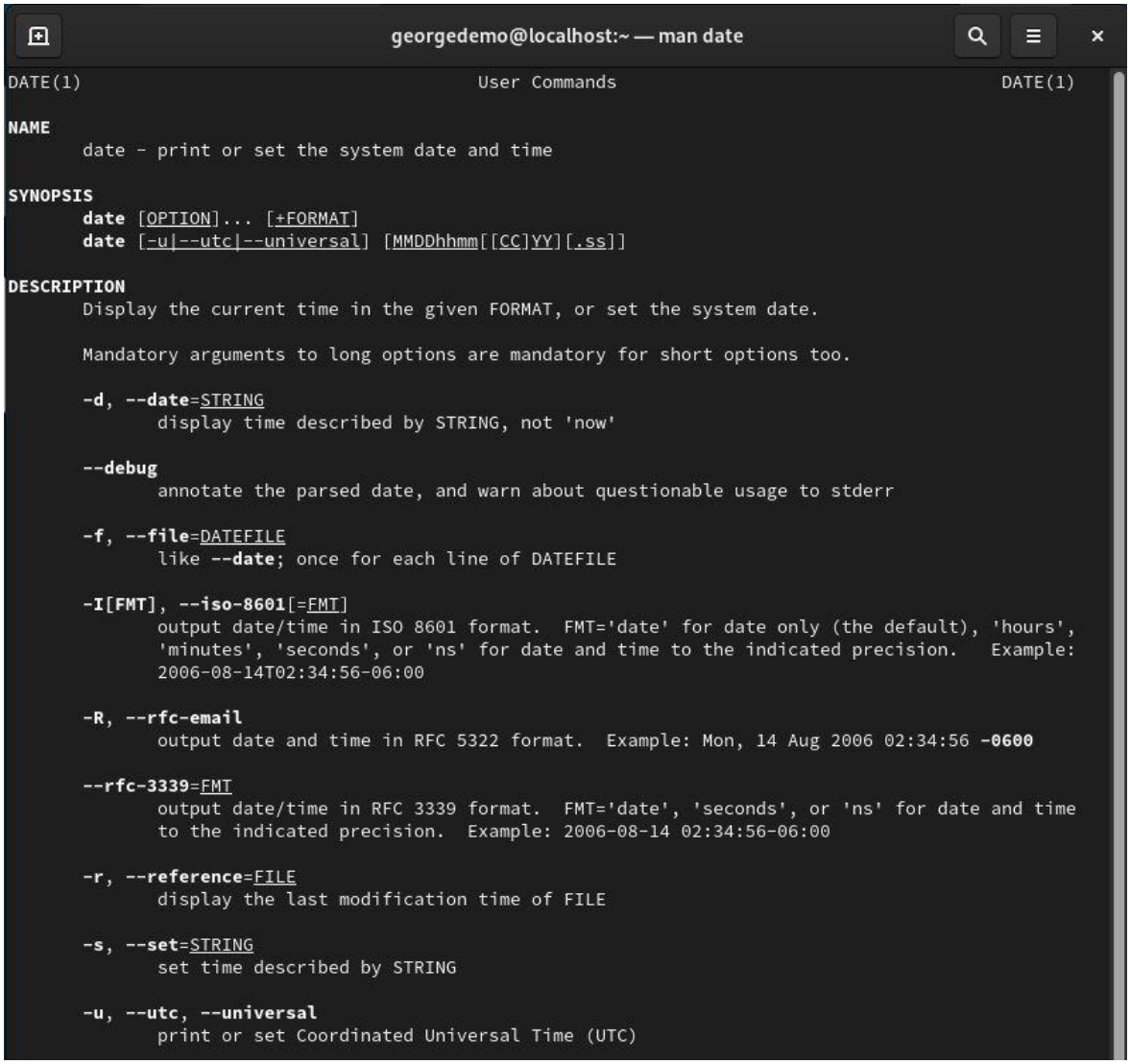
Ο σκοπός των σελίδων *man* είναι να δώσουν τόσο μια σύνοψη της κάθε εντολής όσο και λεπτομέρειες αναφορικά με τις επιλογές που τη συνοδεύουν καθώς και μια περιγραφή των χαρακτηριστικών της. Παράλληλα οι σελίδες man δίνονται και ως μέρος των πακέτων λογισμικού παρέχοντας έτσι την αντίστοιχη τεκμηρίωση.

Ενώ όμως οι σελίδες man έχουν μια πιο αυστηρή μορφή και είναι δομημένες ως μια συλλογή ανεξάρτητων αρχείων κειμένου, στο Linux υπάρχει και μια διαφορετική μορφή τεκμηρίωσης, τα κείμενα Info, που εστιάζουν στην υποστήριξη ολόκληρων πακέτων λογισμικού. Το GNU Project έχει αναπτύξει και υποστηρίζει το σύστημα τεκμηρίωσης GNU Info παρέχοντας παράλληλα παραδείγματα χρήσης των πακέτων λογισμικού που αναπτύσσει.

2.6.1 Εύρεση πληροφορίας στις τοπικές σελίδες του εγχειριδίου (man pages)

Προκειμένου να δούμε την αντίστοιχη σελίδα man για μια εντολή, θα πρέπει να εκτελέσουμε την εντολή man σε ένα παράθυρο τερματικού ακολουθούμενη από το όνομα της εντολής, π.χ.

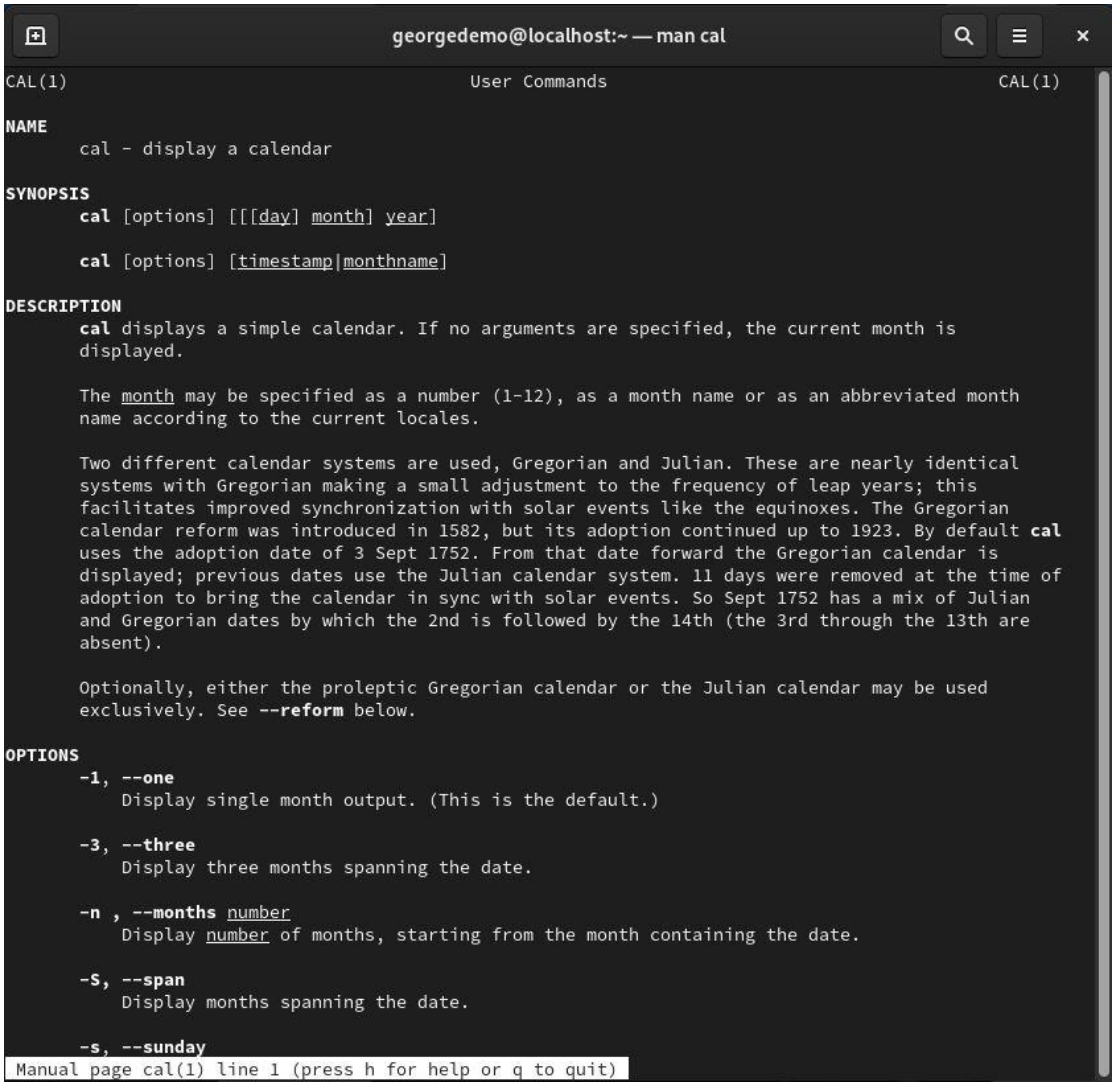
```
man date
```



Εικόνα 35 Η εντολή man (cal)

ή

```
man cal
```



Εικόνα 36 Η εντολή man (cal)

Όπως βλέπουμε και στο αποτέλεσμα της εκτέλεσης των δύο παραπάνω εντολών, η έξοδος της εντολής man περιλαμβάνει τα παρακάτω πεδία:

Πεδίο	**Περιγραφή**
NAME	Περιλαμβάνει το όνομα της εντολής ή ενός αρχείου και μια πολύ σύντομη περιγραφή
SYNOPSIS	Μια περίληψη του συντακτικού της εντολής
DESCRIPTION	Είναι μια αναλυτική περιγραφή της εντολής που εξηγεί τις λειτουργίες που επιτελούνται
OPTIONS	Παρουσιάζονται αναλυτικά οι διαφορετικές επιλογές και μια σύντομη επεξήγηση κάθε μίας
EXAMPLES	Παραδείγματα χρήσης της εντολής
FILES	Παρουσιάζεται μια λίστα των φακέλων και αρχείων που

	σχετίζονται με τη σελίδα man
SEE ALSO	Είναι όλη η σχετιζόμενη πληροφορία, ενώ τις περισσότερες φορές οδηγεί σε άλλα σημεία των σελίδων man
BUGS	Γνωστά προβλήματα στο λογισμικό
AUTHOR	Περιλαμβάνει πληροφορίες σχετικά με όλους τους συντελεστές της ανάπτυξης αυτής της ενότητας

Σημείωση: Αν διαβάσουμε προσεκτικά την έξοδο που έχει παράξει η εκτέλεση της εντολής **`man date`** αλλά και της **`man cal`**, μπορούμε να δούμε ότι στην πρώτη γραμμή εμφανίζεται **`DATE(1)`** και **`CAL(1)`**. Αυτό σημαίνει ότι εμφανίζεται η τεκμηρίωση από την ενότητα **`1`** των σελίδων **`man`**. Θα δούμε όμως αναλυτικά τις ενότητες man στις επόμενες σελίδες.

Πλοήγηση στις σελίδες man με χρήση του πληκτρολογίου

Προκειμένου να πλοηγηθούμε στις σελίδες man θα χρειαστεί να χρησιμοποιήσουμε το πληκτρολόγιο αποτελεσματικά. Παρακάτω εμφανίζονται μερικές από τις επιλογές που έχουμε:

Εντολή-πλήκτρο	**Χρήση**
Κάτω βελάκι (*Down Arrow*)	Κινεί τη σελίδα προς τα κάτω κατά μία γραμμή.
Πάνω βελάκι (*Up Arrow*)	Κινεί τη σελίδα προς τα πάνω κατά μία γραμμή.
Σελίδα προς τα κάτω (*Page Down*) ή το πλήκτρο **εισαγωγής κενού** (*Backspace*):	Κινεί τη σελίδα προς τα κάτω κατά μία σελίδα.
Σελίδα προς τα πάνω (*Page Up*)	Κινεί τη σελίδα προς τα επάνω κατά μία σελίδα.
D	Κινεί τη σελίδα προς τα κάτω κατά μισή σελίδα.
U	Κινεί τη σελίδα προς τα επάνω κατά μισή σελίδα.
Κλείσιμο (*Q*)	Κλείνει το παράθυρο του εγχειριδίου man και επιστρέφει στο τερματικό.
/<*κείμενο*>	Αναζητά το <κείμενο> προς τα εμπρός στο εγχειρίδιο man. Για παράδειγμα, /date θα αναζητήσει τη λέξη "date" στο κείμενο.
N	Μεταβαίνει στο επόμενο αποτέλεσμα αναζήτησης.
?<*κείμενο*>	Αναζητά το <κείμενο> προς τα πίσω στο κείμενο. Για

	παράδειγμα, ?λέξη θα αναζητήσει τη λέξη "λέξη" προς τα πίσω στο κείμενο.
Shift+N	Μεταβαίνει στο προηγούμενο αποτέλεσμα αναζήτησης προς τα πίσω.
G	Μεταφέρει στην αρχή του εγχειριδίου man
Shift+G	Μεταφέρει στο τέλος του εγχειριδίου man
G – εισαγωγή αριθμού σελίδας - Enter	Σας επιτρέπει να εισάγετε τον αριθμό της σελίδας προς την οποία θέλετε να μεταβείτε.

Ενότητες στις σελίδες Man (Man Sections)

Από την αρχική δημιουργία του εγχειριδίου τεκμηρίωσης man, αυτό περιλαμβάνει έναν αριθμό από διαφορετικές ενότητες. Έτσι μπορεί να περιλαμβάνει τεκμηρίωση όχι μόνο για τις εντολές αλλά και για αρχεία ρύθμισης του συστήματος (system files) που περιέχουν πληροφορίες σχετικά με τη ρύθμιση του λειτουργικού συστήματος ή/και των υπηρεσιών.

Έτσι, για την καλύτερη οργάνωση των σελίδων man σε όλες τις διανομές, αυτές κατηγοριοποιούνται σε εννέα (9) ενότητες. Αυτές είναι:

Ενότητα	**Περιεχόμενο**
1	Προγράμματα που μπορούν να εκτελεστούν ή εντολές κελύφους (executable and shell programs)
2	Κλήσεις συστήματος (kernel routines)
3	Συναρτήσεις βιβλιοθηκών (Library functions)
4	Ειδικά αρχεία (π.χ. device files)
5	Μορφές αρχείων (για αρχεία ρυθμίσεων συστήματος - configuration files)
6	Παιχνίδια
7	Διάφορα (περιλαμβάνει μακρο-πακέτα και πρότυπα)
8	Εντολές διαχείρισης συστήματος και διαβαθμισμένες εντολές
9	Ρουτίνες του Linux kernel (internal kernel calls)

Προκειμένου να εξετάσουμε σε ποια ενότητα βρίσκεται η τεκμηρίωση για μία εντολή, μπορούμε να πραγματοποιήσουμε μια αναζήτηση χρησιμοποιώντας το όνομα της εντολής. Έτσι αν πληκτρολογήσουμε:

man –f passwd

Θα λάβουμε την παρακάτω έξοδο:

```
georgedemo@localhost:~$ man -f passwd
passwd (5)          - password file
passwd (1)          - update user's authentication tokens
passwd (1ossl)      - OpenSSL application commands
```

Εδώ βλέπουμε τρεις διαφορετικές σελίδες εγχειριδίου που σχετίζονται με τη λέξη-κλειδί "passwd". Η περιγραφή που ακολουθεί τον τίτλο εξηγεί σύντομα τον σκοπό της κάθε σελίδας του εγχειριδίου.

Έτσι το passwd (1) περιγράφει την εντολή που χρησιμοποιούμε για την αλλαγή των κωδικών, ενώ το passwd (5) περιγράφει τη δομή του αρχείου /etc/passwd στο οποίο αποθηκεύονται στοιχεία των τοπικών λογαριασμών των χρηστών.

Το ίδιο αποτέλεσμα με την εντολή man –f επιτυγχάνουμε και με τη χρήση της εντολής whatis. Έτσι αν πληκτρολογήσουμε:

whatis passwd

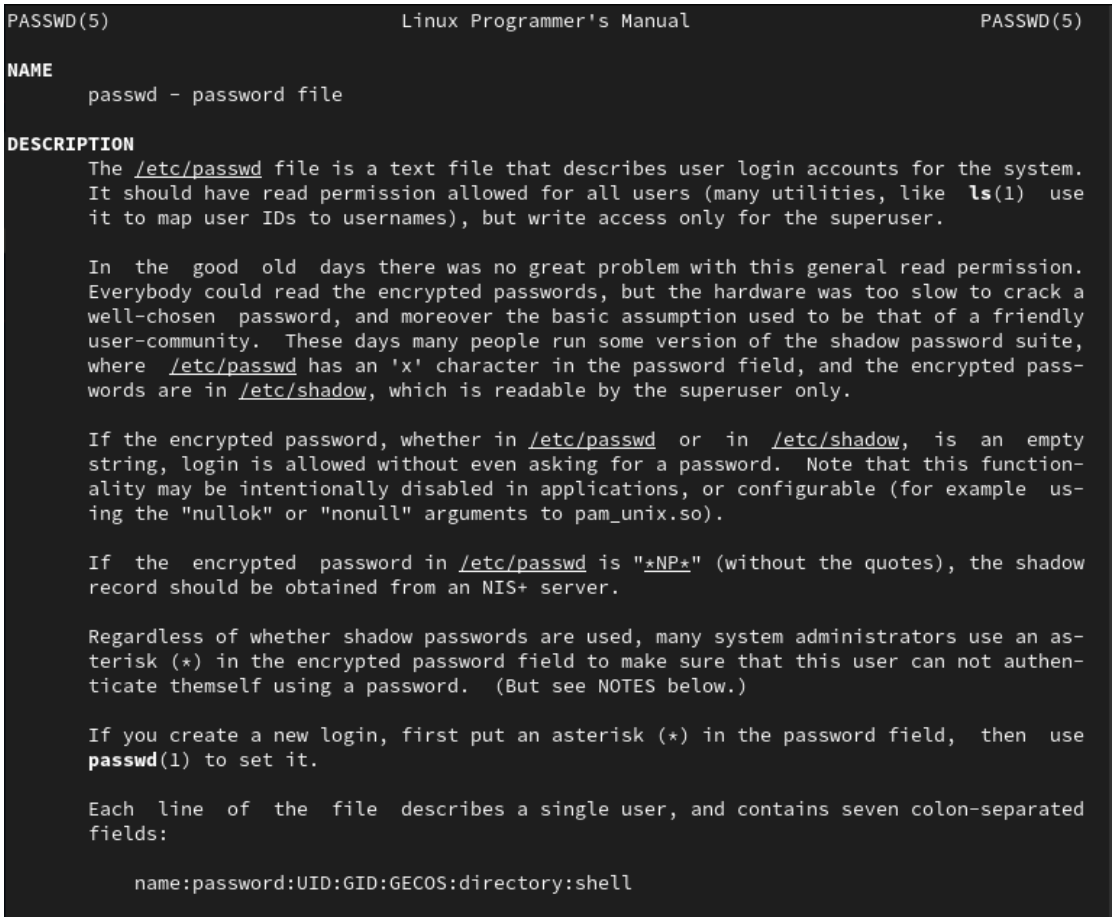
Θα λάβουμε την παρακάτω έξοδο:

```
georgedemo@localhost:~$ whatis passwd
passwd (5)          - password file
passwd (1)          - update user's authentication tokens
passwd (1ossl)      - OpenSSL application commands
```

Επομένως αν θέλουμε να διαβάσουμε τις σελίδες του εγχειριδίου man που αφορούν στο αρχείο passwd θα πρέπει να εισάγουμε την εντολή:

man 5 passwd

και θα λάβουμε την παρακάτω έξοδο:



```
PASSWD(5)                    Linux Programmer's Manual                    PASSWD(5)

NAME
       passwd - password file

DESCRIPTION
       The /etc/passwd file is a text file that describes user login accounts for the system.
       It should have read permission allowed for all users (many utilities, like ls(1) use
       it to map user IDs to usernames), but write access only for the superuser.

       In the good old days there was no great problem with this general read permission.
       Everybody could read the encrypted passwords, but the hardware was too slow to crack a
       well-chosen password, and moreover the basic assumption used to be that of a friendly
       user-community. These days many people run some version of the shadow password suite,
       where /etc/passwd has an 'x' character in the password field, and the encrypted pass-
       words are in /etc/shadow, which is readable by the superuser only.

       If the encrypted password, whether in /etc/passwd or in /etc/shadow, is an empty
       string, login is allowed without even asking for a password. Note that this function-
       ality may be intentionally disabled in applications, or configurable (for example us-
       ing the "nullok" or "nonull" arguments to pam_unix.so).

       If the encrypted password in /etc/passwd is "*NP*" (without the quotes), the shadow
       record should be obtained from an NIS+ server.

       Regardless of whether shadow passwords are used, many system administrators use an as-
       terisk (*) in the encrypted password field to make sure that this user can not authen-
       ticate themself using a password. (But see NOTES below.)

       If you create a new login, first put an asterisk (*) in the password field, then use
       passwd(1) to set it.

       Each line of the file describes a single user, and contains seven colon-separated
       fields:

           name:password:UID:GID:GECOS:directory:shell
```

Εικόνα 37 man passwd

Στην πρώτη γραμμή μπορούμε να δούμε το **PASSWD(5)** που σημαίνει ότι εμφανίζεται η τεκμηρίωση από την ενότητα **5** των σελίδων **man**.

Αναζήτηση σελίδων man με λέξη κλειδί

Στην περίπτωση που δε γνωρίζουμε ή δεν θυμόμαστε το όνομα της εντολής που αναζητούμε στις σελίδες man, τότε μπορούμε να αναζητήσουμε με χρήση μιας λέξης κλειδί με χρήση της επιλογής **–k** όπως παρακάτω:

```
[georgedemo@localhost ~]$ man -k cal
rpc (3)              - library routines for remote procedure calls
Env (3pm)            - perl module that imports environment variables as scalars or arrays
_Exit (2)            - terminate the calling process
_exit (2)            - terminate the calling process
_longjmp (3p)        - non-local goto
_syscall (2)         - invoking a system call without library support (OBSOLETE)
afs_syscall (2)      - unimplemented system calls
alloca (3)           - allocate memory that is automatically freed
anacron (8)          - runs commands periodically
```

Εικόνα 38 Αναζήτηση σελίδων εγχειριδίου με λέξη κλειδί

Προσοχή: Η έξοδος αυτής της εντολής μπορεί να είναι πολύ εκτεταμένη.

Παράλληλα με την εντολή **man –k** μπορούμε να χρησιμοποιήσουμε την εντολή **apropos** που στις περισσότερες διανομές Linux παράγει το ίδιο αποτέλεσμα.

Σημείωση: Η αναζήτηση χρησιμοποιώντας λέξη κλειδί βασίζεται σε ένα λεξικό που δημιουργείται με την εντολή mandb. Αυτή θα πρέπει να εκτελεστεί με χρήση του λογαριασμού root και εκτελείται καθημερινά από το σύστημα. Στην περίπτωση που θέλουμε να επικαιροποιήσουμε το εγχειρίδιο man αρκεί να εκτελέσουμε την εντολή sudo mandb.

Παρουσιάζοντας τις σελίδες man σε ένα περιηγητή ιστού

Επειδή οι σελίδες man έχουν μεγάλη έκταση και μπορεί να είναι σχετικά δύσκολο να περιπλανηθούμε σε αυτές υπάρχει η δυνατότητα εξαγωγής τους σε μορφή html και ανάγνωσης τους σε έναν περιηγητή ιστού.

Προκειμένου να το κάνουμε αυτό αρχικά εγκαθιστούμε το πακέτο groff ακολουθώντας τα ακόλουθα βήματα:

1. Εγκατάσταση του πακέτου:

```
sudo yum install groff
```

2. Επιλογή περιηγητή ιστού:

```
export BROWSER=firefox
```

Αν θέλουμε να επιλέξουμε έναν άλλο περιηγητή ιστού αντικαθιστούμε τη λέξη firefox με google-chrome ή chromium-browser ή όποιον άλλο περιηγητή επιθυμούμε.

3. Χρησιμοποιούμε την επιλογή -H για να ανοίξουμε τη σελίδα man στον περιηγητή ιστού:

```
man -Hfirefox date
```

Η επιλογή -H option προκαλεί την εφαρμογή groff να παράγει την έξοδο σε μορφή HTML και ανοίγει αυτόματα τον αντίστοιχο περιηγητή ιστού.

4. Χρησιμοποιούμε την επιλογή –t προκειμένου να παράγουμε έξοδο σε μορφή Postscript και με χρήση ανακατεύθυνσης να την οδηγήσουμε σε ένα αρχείο. Στη συνέχεια μπορούμε να εκτυπώσουμε αυτό το αρχείο.

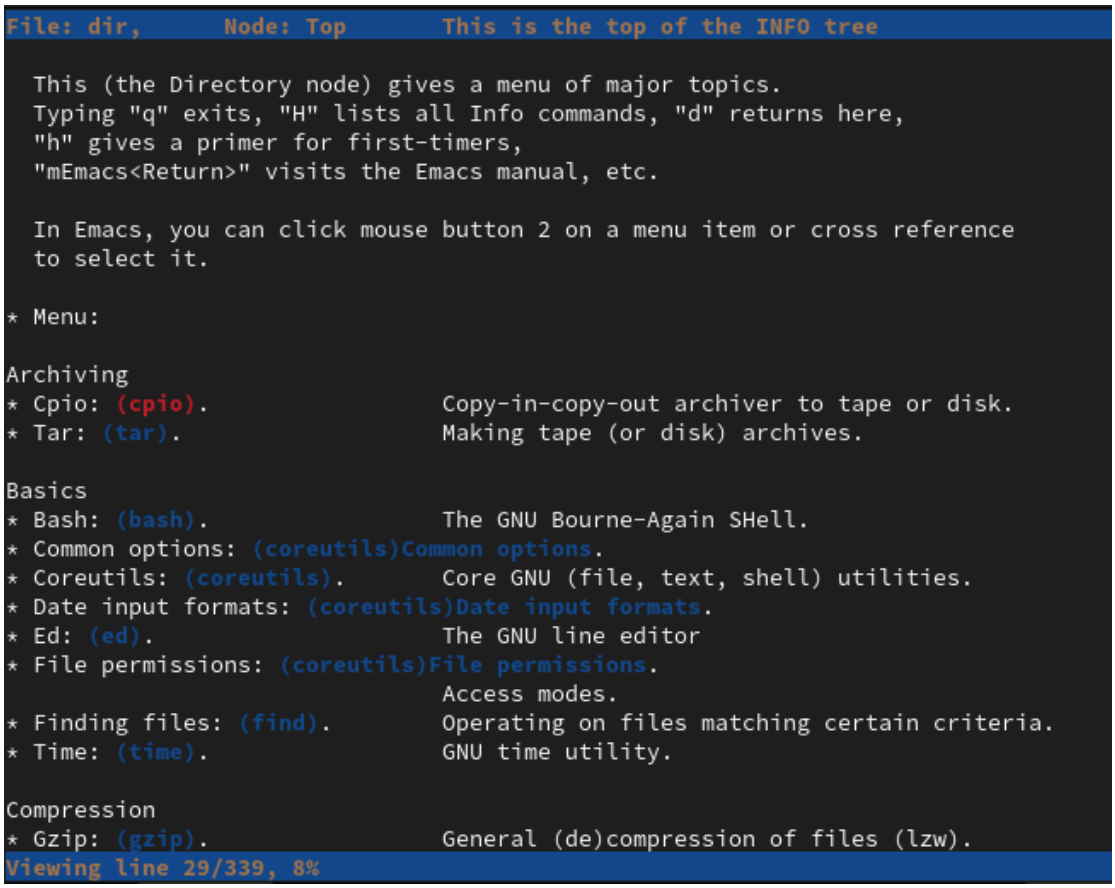
```
man –t date > date.ps (εξαγωγή σε αρχείο Postscript)

man –t date > date.pdf (εξαγωγή σε αρχείο pdf)

man –t date | lp (εκτύπωση της σελίδας man)
```

2.6.2 Εύρεση πληροφορίας από τοπική τεκμηρίωση μέσω του GNU Info.

Προκειμένου να διαβάσουμε τις σελίδες Info, μπορούμε να χρησιμοποιήσουμε την εντολή **`pinfo`** οπότε και θα εμφανίσει τον αρχικό κατάλογο της τεκμηρίωσης.



Εικόνα 39 Η εντολή pinfo

Αν και το **`Pinfo`** είναι εγκατεστημένο εκ των προτέρων, στην περίπτωση που η εντολή δεν επιστρέψει αποτελέσματα, μπορούμε να το εγκαταστήσουμε άμεσα με την εντολή:

```
sudo yum install pinfo
```

Σε αντίθεση με την τεκμηρίωση man, η τεκμηρίωση **Info** είναι εκτενής ενώ χρησιμοποιεί υπερσυνδέσμους για να δομήσει την πληροφορία. Επιπλέον η εξαγωγή σελίδων info είναι δυνατή σε πολλές μορφές. Αντίθετα, οι σελίδες man είναι βελτιστοποιημένες για εκτυπωμένη έξοδο. Η μορφή Info είναι πιο ευέλικτη από τις σελίδες man, επιτρέποντας την σε βάθος συζήτηση σύνθετων εντολών και εννοιών.

Μια τυπική σελίδα Info είναι ένα ολοκληρωμένο έγγραφο παρέχοντας τις ακόλουθες βελτιώσεις:

- Έναν πίνακα περιεχομένου προς όλα τα κείμενα που αφορούν σε ένα θεματικό αντικείμενο
- Πλήρη αναζήτηση κειμένου σε ολόκληρο το έγγραφο

- Ένα ενιαίο έγγραφο που περιέχει όλες τις πληροφορίες που αφορούν σε ένα θεματικό αντικείμενο
- Υπερσυνδέσμους που συνδέουν όλα τα κείμενα που ανήκουν στην ίδια θεματική κατηγορία

Παρότι αρκετές εντολές διαθέτουν τόσο σελίδες man αλλά και τεκμηρίωση σε μορφή info, η τελευταία παρέχει περισσότερο αναλυτικές πληροφορίες. Επιπλέον το **pinfo** είναι η πιο βελτιωμένη έκδοση **info**, ενώ η τεκμηρίωση που παρέχει επεκτείνεται συνεχώς με την εγκατάσταση νέων πακέτων λογισμικού.

Παρότι τόσο το **man** όσο και το **info** (**pinfo**) παρουσιάζουν τεκμηρίωση για εντολές και λειτουργίες, διαφέρουν ως προς τα πλήκτρα που χρησιμοποιούν για την κίνηση εντός της τεκμηρίωσης.

Χρήση	**MAN (Εντολή-πλήκτρο)**	**INFO (Εντολή-πλήκτρο)**
Παρουσιάζει τον κατάλογο των αντικειμένων τεκμηρίωσης	--	**D**
Κινεί τη σελίδα προς τα κάτω κατά μία γραμμή.	**Κάτω βελάκι** (*Down Arrow*)	**Κάτω βελάκι** (*Down Arrow*)
Κινεί τη σελίδα προς τα πάνω κατά μία γραμμή.	**Πάνω βελάκι** (*Up Arrow*)	**Πάνω βελάκι** (*Up Arrow*)
Κινεί τη σελίδα προς τα κάτω κατά μία σελίδα.	**Σελίδα προς τα κάτω** (*Page Down*) ή το πλήκτρο **εισαγωγής κενού** (*Backspace*):	**Σελίδα προς τα κάτω** (*Page Down*) ή το πλήκτρο **εισαγωγής κενού** (*Backspace*):
Κινεί τη σελίδα προς τα επάνω κατά μία σελίδα.	**Σελίδα προς τα πάνω** (*Page Up*)	**Σελίδα προς τα πάνω** (*Page Up*)
Κινεί τη σελίδα προς τα κάτω κατά μισή σελίδα.	**D**	--
Κινεί τη σελίδα προς τα επάνω κατά μισή σελίδα.	**U**	--
Κλείνει το παράθυρο του εγχειριδίου man και επιστρέφει στο τερματικό.	**Κλείσιμο** (*Q*)	**Κλείσιμο** (*Q*)
Αναζητά το <κείμενο> προς τα εμπρός στο εγχειρίδιο	*/<κείμενο>*	*/<κείμενο>*

man. Για παράδειγμα, /date θα αναζητήσει τη λέξη "date" στο κείμενο.		
Μεταβαίνει στο επόμενο αποτέλεσμα αναζήτησης.	**N**	/ και μετά **Enter**

3. Δημιουργία, επισκόπηση και διόρθωση αρχείων κειμένου

Μαθησιακοί στόχοι

Οι επιμορφούμενοι θα πρέπει να είναι σε θέση να:

- Αποθηκεύουν τα αποτελέσματα ή τα μηνύματα λάθους από εκτέλεση εντολών σε αρχείο με χρήση της ανακατεύθυνσης ροής
- Δημιουργούν και επεξεργάζονται αρχεία από τη γραμμή εντολών ή από έναν κειμενογράφο, όπως ο **vim**.
- Να χρησιμοποιούν μεταβλητές φλοιού βοηθητικά για το τρέξιμο εντολών και να διορθώνουν αρχεία σεναρίου εκκίνησης του Bash, ώστε να ορίζουν μεταβλητές φλοιού και περιβάλλοντος, ώστε να αλλάζουν τη συμπεριφορά του φλοιού και προγραμμάτων που τρέχουν από το φλοιό.

3.1 Αποθήκευση του αποτελέσματος εντολών σε αρχείο με χρήση ροής

Μια *διεργασία* (πρόγραμμα που εκτελείται) διαβάζει κάποια είσοδο και παράγει κάποια έξοδο. Για εντολές που τρέχουν από το φλοιό, κανονικά η είσοδος γίνεται από το πληκτρολόγιο και η έξοδος εμφανίζεται στο παράθυρο του τερματικού.

Για τις διεργασίες χρησιμοποιούνται αριθμημένοι δίαυλοι (κανάλια), που ονομάζονται *περιγραφείς αρχείων* (file dedcriptors) και για κάθε διεργασία υπάρχουν τουλάχιστον τρεις:

- Η *καθιερωμένη είδοδος* (standard input, δίαυλος 0, όνομα διαύλου **stdin**) διαβάζει από το πληκτρολόγιο.
- Η *καθιερωμένη έξοδος* (standard output, δίαυλος 1, όνομα διαύλου **stdout**) στέλνει την κανονική έξοδο στο τερματικό.
- Η *καθιερωμένη έξοδος λαθών* (standard error, δίαυλος 2, όνομα διαύλου **stderr**) στέλνει μηνύματα λαθών στο τερματικό.

Αν κάποιο πρόγραμμα ανοίγει χωριστές συνδέσεις προς άλλα αρχεία, χρησιμοποιούνται μεγαλύτεροι (αριθμητικά) περιγραφείς αρχείων.

Η *ανακατεύθυνση εισόδου/εξόδου* (I/O redirection) αλλάζει τη ροή εισόδου ή εξόδου μιας διεργασίας. Αντί να λαβαίνει είσοδο από το πληκτρολόγιο και να στέλνει έξοδο και λάθη στο τερματικό, η διεργασία διαβάζει από και γράφει σε αρχεία. Με αυτό τον τρόπο μπορούμε να σώσουμε μηνύματα που κανονικά στέλνονται στο παράθυρο του

τερματικού, σε ένα αρχείο, ή με την ανακατεύθυνση να απορρίψουμε ανεπιθύμητη έξοδο ή λάθη, ώστε να μην εμφανίζονται στο τερματικό.

Ας δούμε τα διάφορα είδη ανακατεύθυνσης.

- `> file` ανακατευθύνει την έξοδο σε αρχείο. Αν το αρχείο προϋπήρχε, χάνονται τα προηγούμενα περιεχόμενά του και η έξοδος της εντολής γράφει από πάνω τους, αλλιώς το αρχείο δημιουργείται. Η stderr εξακολουθεί να εμφανίζει τυχόν μηνύματα λάθους στο τερματικό.

  Παράδειγμα:
  ```
  $ ls > lsout.txt
  ```

  (η λίστα των αρχείων του τρέχοντος καταλόγου σώζεται στο αρχείο με το όνομα `lsout.txt`)

- `>> file` ανακατευθύνει την έξοδο σε αρχείο. Αν το αρχείο προϋπήρχε και περιείχε δεδομένα, η έξοδος της εντολής προστίθεται μετά το τέλος τους, αλλιώς το αρχείο δημιουργείται. Η stderr εξακολουθεί να εμφανίζει τυχόν μηνύματα λάθους στο τερματικό.

  Παράδειγμα:
  ```
  $ ls >> lsout.txt
  ```

  (η λίστα των αρχείων του τρέχοντος καταλόγου προστίθεται στο τέλος των περιεχομένων του αρχείου με το όνομα lsout.txt)

3.1.1 Αποθήκευση των μηνυμάτων λάθους που παρουσιάζονται κατά την εκτέλεση εντολών σε αρχείο.

Η αποθήκευση των μηνυμάτων λάθους που εμφανίζονται κατά την εκτέλεση εντολών γίνεται με την ανακατεύθυνση της καθιερωμένης εξόδου λαθών (stderr) σε ένα αρχείο.

Ας δούμε τους τρόπους που μπορεί να γίνει αυτό:

- `2> file` ανακατευθύνει την έξοδο λαθών σε αρχείο. Αν το αρχείο προϋπήρχε, χάνονται τα προηγούμενα περιεχόμενά του και η έξοδος της εντολής γράφει από πάνω τους, αλλιώς το αρχείο δημιουργείται. Η stdout ξακολουθεί να εμφανίζει την έξοδό της στο τερματικό.

- `2> /dev/null` απορρίπτει την έξοδο λαθών, με το να την ανακατευθύνει στο αρχείο /dev/null. Η stdout ξακολουθεί να εμφανίζει την έξοδό της στο τερματικό.

- `> file 2>&1` ανακατευθύνει την stdout και την stderr στο ίδιο αρχείο. Αν το αρχείο προϋπήρχε, χάνονται τα προηγούμενα περιεχόμενά του. (το ίδιο αποτέλεσμα έχει και το: `&> file`)

- `>> file 2>&1` ανακατευθύνει την stdout και την stderr στο ίδιο αρχείο. Αν το αρχείο προϋπήρχε, ό,τι παράγουν η καθιερωμένη έξοδος και η έξοδος λαθών προστίθενται μετά τα προϋπάρχοντα περιεχόμενά του. (το ίδιο αποτέλεσμα έχει και το: `&>> file`)

Εδώ πρέπει να επισημάνουμε τη σημασία που έχει η σειρά των λειτουργιών ανακατεύθυνσης. Η ακολουθία:

```
> file 2>&1
```

ανακατευθύνει την stdout στο αρχείο **file** και μετά ανακατευθύνει την stderr στο ίδιο μέρος με την stdout (το αρχείο **file**).

Η ακολουθία:

```
2>&1 > file
```

κάνει τις ανακατευθύνσεις με την αντίστροφη σειρά. Πρώτα ανακατευθύνει την stderr στο καθιερωμένο μέρος για την stdout (το παράθυρο του τερματικού, που έτσι κι αλλιώς είναι το σύνηθες μέρος) και *μετά* ανακατευθύνει **μόνο** την stdout στο αρχείο **file** (άρα τελικά το αρχείο **file** δεν περιέχει την έξοδο λαθών)**.**

Οι εναλλακτικοί τρόποι που αναφέρθηκαν στις δύο τελευταίες περιπτώσεις ενώνουν τις ανακατευθύνσεις, οπότε λύνεται το ζήτημα της σωστής σειράς. Έχουν το μειονέκτημα να μην υποστηρίζονται από όλα τα κελύφη επειδή είναι νεότεροι και αυτός είναι ένας λόγος που τους αποφεύγουν κάποιοι διαχειριστές συστημάτων.

3.1.2 Χρήση Pipes για τροφοδότηση της εξόδου μιας εντολής ως είσοδο σε μια άλλη

Στα λειτουργικά συστήματα τύπου unix, είναι συχνή η χρήση ακολουθιών εντολών, οι οποίες χωρίζονται μεταξύ τους με το χαρακτήρα «|». Οι ακολουθίες αυτές ονομάζονται pipelines (διοχετεύσεις) και ο χαρακτήρας «|» ονομάζεται pipe character (χαρακτήρας διοχέτευσης).

Μια διοχέτευση (pipe) συνδέει την καθιερωμένη έξοδο της πρώτης εντολής με την καθιερωμένη είσοδο της δεύτερης εντολής. Έτσι, η δεύτερη εντολή δέχεται σαν

δεδομένα εισόδου τα δεδομένα που παράγει η έξοδος της πρώτης εντολής και τα επεξεργάζεται.

Αυτό μπορεί να γίνει με περισσότερες από δύο εντολές, οδηγώντας σε μια σειρά ανακατευθύνσεων, όπου η αρχική έξοδος της πρώτης εντολής περνάει από διαδοχικά στάδια επεξεργασίας.

Ένα παράδειγμα με δύο εντολές είναι το παρακάτω:

```
$ ls -1 /usr/bin | less
```

Η πρώτη εντολή παράγει μια λίστα των περιεχομένων του καταλόγου /usr/bin (το -1 κάνει την εντολή ls να παράγει την έξοδο με ένα μόνο όνομα ανά γραμμή). Επειδή τα περιεχόμενα του καταλόγου /usr/bin είναι συνήθως πολλά και δε χωράνε σε μια οθόνη, τα διοχετεύουμε στην εντολή less, που μας επιτρέπει να τα δούμε οθόνη-οθόνη και να πάμε εμπρός-πίσω.

Στο παρακάτω παράδειγμα, η εντολή ls διοχετεύει την έξοδό της στην εντολή wc -l, η οποία τυπώνει τον αριθμό των γραμμών που δέχτηκε από την ls:

```
$ ls | wc -l
```

Να και ένα παράδειγμα με συνδυασμό διοχέτευσης και ανακατεύθυνσης, όπου η ls -Sr δίνει μια λίστα των περιεχομένων του τρέχοντος καταλόγου, ταξινομημένα από το μεγαλύτερο σε μέγεθος στο μικρότερο, η εντολή head -n 10 κρατάει τις δέκα πρώτες γραμμές από την είσοδο που δέχεται και αυτές σώζονται στο αρχείο ten_largest_files.txt, πάλι στον τρέχοντα κατάλογο:

```
$ ls -S | head -n 10 > ten_largest_files.txt
```

3.1.3 Η εντολή tee, οι ανακατευθύνσεις και οι διοχετεύσεις

Όταν υπάρχει συνδυασμός ανακατευθύνσεων και διοχετεύσεων, ο φλοιός πρώτα ετοιμάζει ολόκληρη τη διοχέτευση και μετά ανακατευθύνει την είσοδο και την έξοδο. Έτσι, αν υπάρχει μια ανακατεύθυνση στη μέση μιας διοχέτευσης, η έξοδος θα πάει στο αρχείο (ανακατεύθυνσης) και όχι στην επόμενη εντολή στη διοχέτευση. Άρα, η εντολή:

```
$ ls > ls_output.txt | less
```

έχει σαν αποτέλεσμα να γραφεί η έξοδος της ls στο αρχείο ls_output.txt και η less δεν εμφανίζει τίποτα στο τερματικό.

Το πρόβλημα αυτό μπορεί να ξεπεραστεί με τη χρήση της εντολής tee. Η tee σε μια pipeline (διοχέτευση) αντιγράφει την καθιερωμένη είσοδό της στην καθιερωμένη της έξοδο , αλλά ταυτόχρονα ανακατευθύνει τη δική της stdout στα αρχεία που της

δίνονται σαν ορίσματα στη γραμμή εντολής. Έτσι, για να έχουμε το επιθυμητό αποτέλεσμα στην προηγούμενη εντολή, πρέπει να τη δώσουμε ως εξής:

```
$ ls | tee ls_output.txt | less
```

(Η tee πήρε το όνομά της από την υδραυλική σύνδεση σε σχήμα «Τ», που δημιουργεί διακλάδωση σε μια παροχή. Μπορείτε να φανταστείτε την οριζόντια γραμμή του «Τ» σαν την κανονική διοχέτευση που καταλήγει στο less και την κάθετη γραμμή σαν τη διακλάδωση που δημιουργεί η tee, ώστε να κατευθύνει τα δεδομένα και στο αρχείο ls_output.txt)

Με τον ίδιο τρόπο, βάζοντας το tee στο τέλος μιας διοχέτευσης, μπορούμε ταυτόχρονα να σώσουμε την τελική έξοδο σε αρχείο και να την εμφανίσουμε στην οθόνη:

```
$ ls -S | head -n 10 | tee ten_largest_files.txt
```

Τέλος, ο σωστός τρόπος για να ανακατευθύνουμε την stderr μέσα από μια διοχέτευση είναι μέσω της ανακατεύθυνσής της στην stdout (υποθέτοντας ότι δεν υπάρχει αρχείο με κατάληξη .zzz στον τρέχοντα κατάλογο):

```
$ ls *.zzz 2>&1 | less
```

Ένα παράδειγμα με περισσότερες εντολές είναι το παρακάτω:

```
$ cat textfile.txt | sort | uniq | wc -w
```

Η cat τυπώνει στο παράθυρο του τερματικού τα περιεχόμενα του (υποθετικού) αρχείου κειμένου textfile.txt, η sort τα ταξινομεί ανά γραμμή, η uniq αφαιρεί τις διπλές ίδιες γραμμές και η wc -w μετράει τον αριθμό των λέξεων που διαβάζει.

3.2 Δημιουργία και διαχείριση αρχείων κειμένου με χρήση του vim

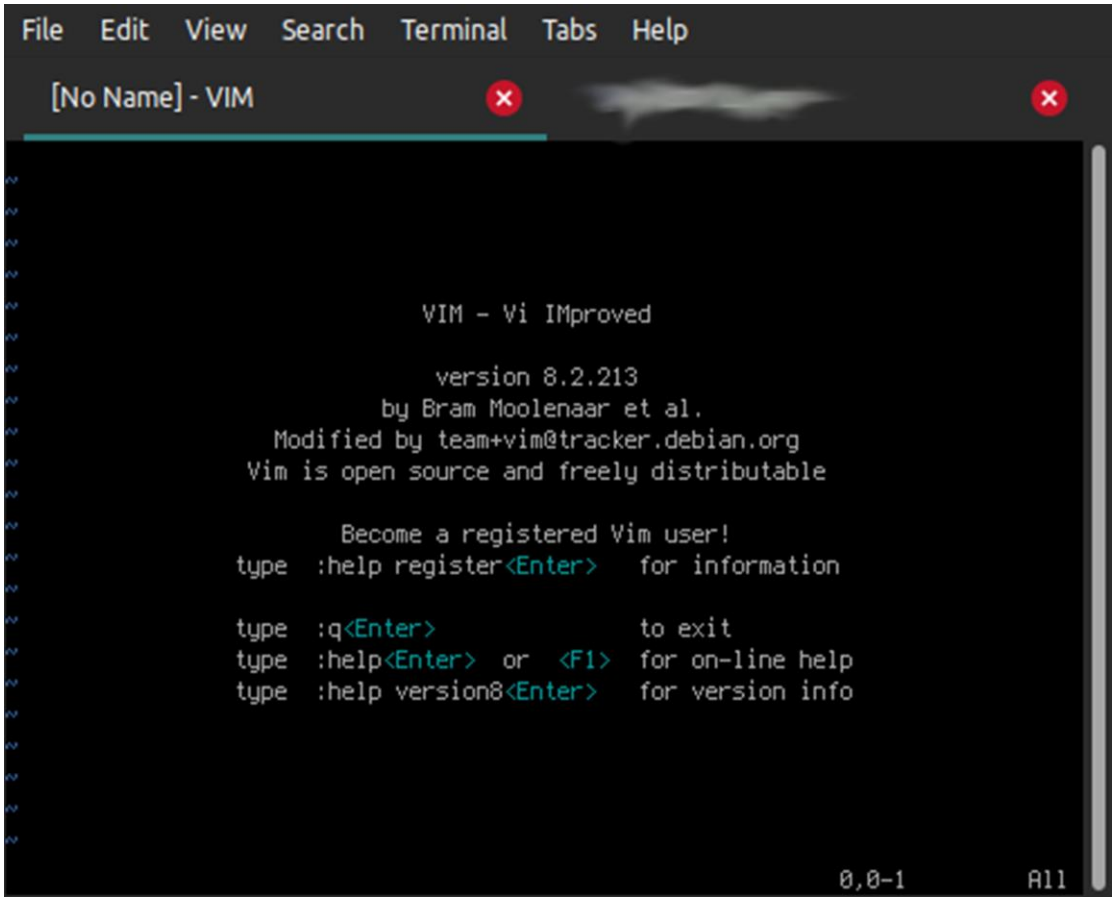
Το Linux, όπως όλα τα λειτουργικά συστήματα βασισμένα στο Unix, για την αποθήκευση των ρυθμίσεων του συστήματος, αλλά και διαφόρων εφαρμογών, χρησιμοποιεί αρχεία ρυθμίσεων που έχουν μορφή αναγνώσιμου κειμένου. Έχουν διάφορες μορφές (όπως αυτή των αρχείων INI ή κάποια νεότερη δομημένη μορφή, όπως YAML ή XML)

Τα αρχεία κειμένου έχουν το πλεονέκτημα να είναι αναγνώσιμα, οπότε είναι πολύ πιο εύκολο να βρεθεί κάποια λάθος ρύθμιση ή να αλλαχτεί/προστεθεί κάτι, χωρίς τη χρήση εξειδικευμένων εργαλείων. Ένα πρόγραμμα κειμενογράφου είναι αρκετό.

Είναι σημαντικό να μάθετε καλά τη χρήση τουλάχιστον ενός κειμενογράφου που να μπορεί να χρησιμοποιηθεί από τερματικό, κονσόλα κειμένου ή σε απομακρυσμένη σύνδεση ssh, ώστε να μπορείτε να κάνετε τις απαιτούμενες εργασίες όταν δεν υπάρχει γραφικό περιβάλλον, είτε επειδή κάποιο πρόβλημα εμποδίζει την εκκίνησή του, είτε

επειδή δεν έχει εγκατασταθεί στο μηχάνημα που πρέπει να επέμβετε, περίπτωση πολύ συνηθισμένη σε μηχανήματα που είναι εξυπηρετητές (servers). Μερικοί τέτοιοι κειμενογράφοι είναι οι pico και nano, που είναι μικροί , απλοί και εύκολοι στη χρήση, αλλά μια καλύτερη επιλογή ίσως είναι ο **vim**.

Ο κειμενογράφος **vim** είναι μια εξελιγμένη έκδοση του **vi** και είναι ένα πρόγραμμα



Εικόνα 40 Η αρχική οθόνη του vim. Κάποιες στοιχειώδεις πληροφορίες και βοήθεια για έξοδο από τον vim. Οι χαρακτήρες «~» αριστερά δείχνουν ότι είμαστε μετά το τέλος του αρχείου που επεξεργαζόμαστε

που έχει μεγάλες δυνατότητες. Όπως και ο πρόγονός του vi, είναι δύσκολος στην εκμάθηση, αλλά αν τον μάθει κανείς, γίνεται ένα παντοδύναμο εργαλείο. Το σημαντικό είναι ότι υπάρχει σε όλες τις διανομές Linux (και τα περισσότερα λειτουργικά τύπου Unix) και αν δεν είναι ήδη εγκατεστημένος, εγκαθίσταται εύκολα.

Σε πολλές διανομές Linux ο vim είναι ήδη εγκατεστημένος σε μια μια περιορισμένη έκδοση (συνήθως λέγεται vim-minimal ή κάτι κοντινό), η οποία έχει περίπου τις δυνατότητες του vi. Σε αυτές τις περιπτώσεις καλό είναι να εγκαθιστούμε την κανονική έκδοση (vim-enhanced ή κάτι παρόμοιο), ώστε να έχουμε όλες τις δυνατότητες του vim. Οι βασικές λειτουργίες που θα περιγράψουμε παρακάτω είναι κοινές στον vim και στον vi.

Για να ξεκινήσουμε να επεξεργαζόμαστε ένα αρχείο κειμένου με τον vim. Δίνουμε απλά την εντολή:

```
$ vim filename
```

(όπου αντικαθιστούμε το *filename* με το όνομα του αρχείου που θέλουμε να επεξεργαστούμε).

3.1.1 Οι τρόποι λειτουργίας του vim

Ο vim έχει το ασυνήθιστο χαρακτηριστικό του να έχει διάφορους τρόπους λειτουργίας (modes): υπάρχουν μεταξύ άλλων η *λειτουργία εντολών (command mode), η εκτεταμένη λειτουργία εντολών (extended command mode)* η *λειτουργία επεξεργασίας (edit mode), και* η *οπτική λειτουργία (visual mode).*

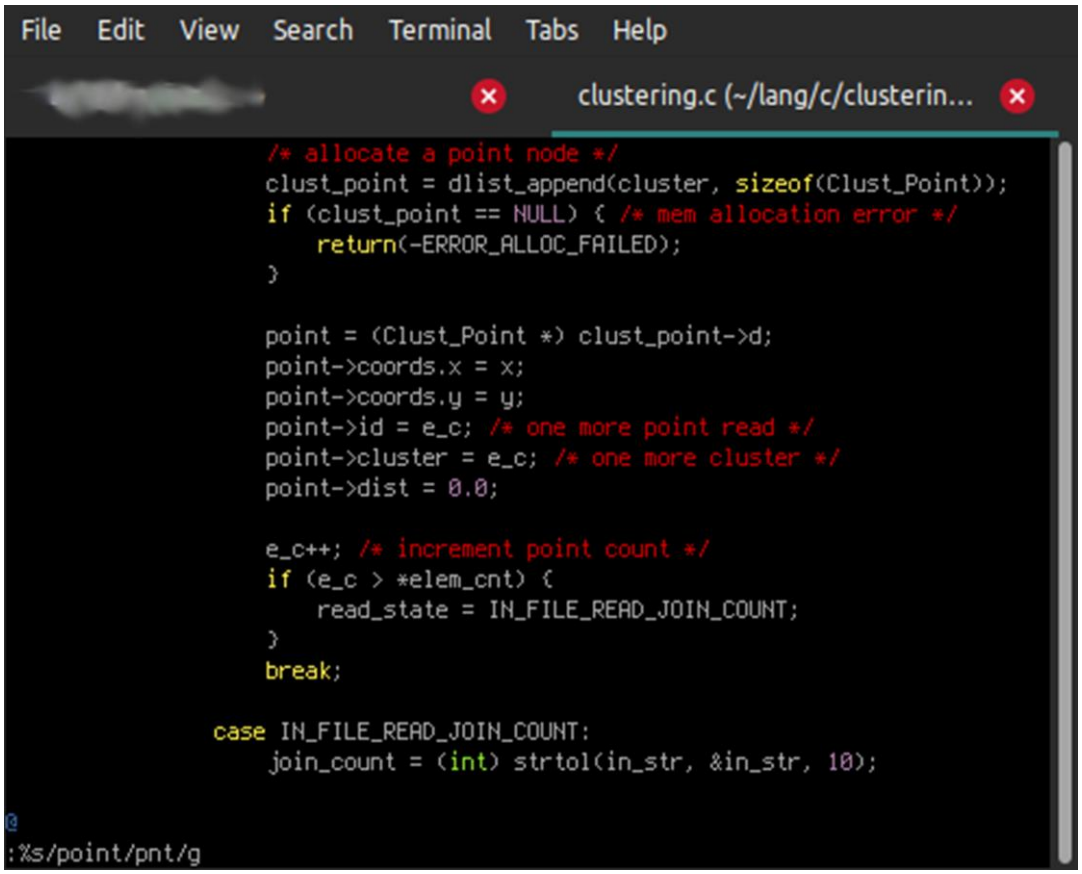
(Αυτή η ιδιοτροπία προέρχεται από το γεγονός ότι ο vim και οι πρόγονοί του κατάγονται από μια εποχή όπου τα πληκτρολόγια των διαφόρων υπολογιστών ήταν πολύ διαφορετικά μεταξύ τους, με διαφορετικά πλήκτρα εκτός των αλφαριθμητικών, τα οποία έδιναν στο λειτουργικό σύστημα εντελώς διαφορετικούς κωδικούς. Τα μόνα σίγουρα ως προς τη λειτουργία τους πλήκτρα ήταν τα βασικά: γράμματα, αριθμοί, και οι συνδυασμοί τους με το πλήκτρο Shift.)

Όταν ξεκινάει ο vim, είναι σε λειτουργία εντολών (command mode). Αυτή τη λειτουργία χρησιμεύει για μετακίνηση στο κείμενο, για κόψιμο και επικόλληση και άλλες λειτουργίες χειρισμού του κειμένου.

Πατώντας το πλήκτρο **i** ο vim μεταβαίνει στη *λειτουργία εισαγωγής* όπου ό,τι γράφουμε γίνεται μέρος του αρχείου κειμένου. Για να επιστρέψουμε στη λειτουργία εντολών, πατάμε το πλήκτρο **Esc**.

Με το πλήκτρο **v** μεταβαίνει στην οπτική λειτουργία (visual mode), όπου μπορούν να επιλεχθούν τμήματα του κειμένου για επεξεργασία. Ο συνδυασμός **Shift + v** χρησιμεύει για την επιλογή πολλαπλών (ολόκληρων) γραμμών, ενώ ο συνδυασμός **Ctrl + v** επιλέγει «ορθογώνιες» περιοχές του κειμένου (π.χ. το κείμενο που περικλείεται από την 8η στήλη της 12ης γραμμής, έως την 57η στήλη της 82ης γραμμής). Για να βγούμε από την οπτική λειτουργία, απλά πατάμε ξανά τον ίδιο συνδυασμό πλήκτρων.

Πατώντας το **:** μεταβαίνουμε στην εκτεταμένη λειτουργία εντολών (extended command mode), για λειτουργίες όπως το σώσιμο το αρχείου και η έξοδος από τον vim.



Εικόνα 41 Ο vim σε εκτεταμένη λειτουργία εντολών. Η συμβολοσειρά μετά το χαρακτήρα της άνω και κάτω τελείας (:) κάτω αριστερά είναι η εντολή. Εδώ μια εντολή εύρεσης και αντικατάστασης κειμένου, κοινή με την αντίστοιχη οδηγία της εντολής sed.

Όταν είμαστε στη λειτουργία εισαγωγής ή στην οπτική λειτουργία, στο κάτω αριστερά μέρος του παράθυρου του τερματικού εμφανίζεται η ένδειξη «-- **INSERT** --» ή «-- **VISUAL** --», αντίστοιχα, ενώ στην εκτεταμένη λειτουργία εντολών εμφανίζεται ο χαρακτήρας «:», μαζί με ό,τι έχουμε πατήσει μετά για να δώσουμε την εντολή.

Αν δεν είμαστε σίγουροι/ες για τη λειτουργία στην οποία βρισκόμαστε, πατώντας μερικές φορές το **Esc** επιστρέφουμε στη λειτουργία εντολών (εκεί το **Esc** δεν επιτελεί καμία λειτουργία, οπότε είναι ασφαλές να το πατήσουμε).

3.1.2 Οι βασικές λειτουργίες διόρθωσης κειμένων με τον vim

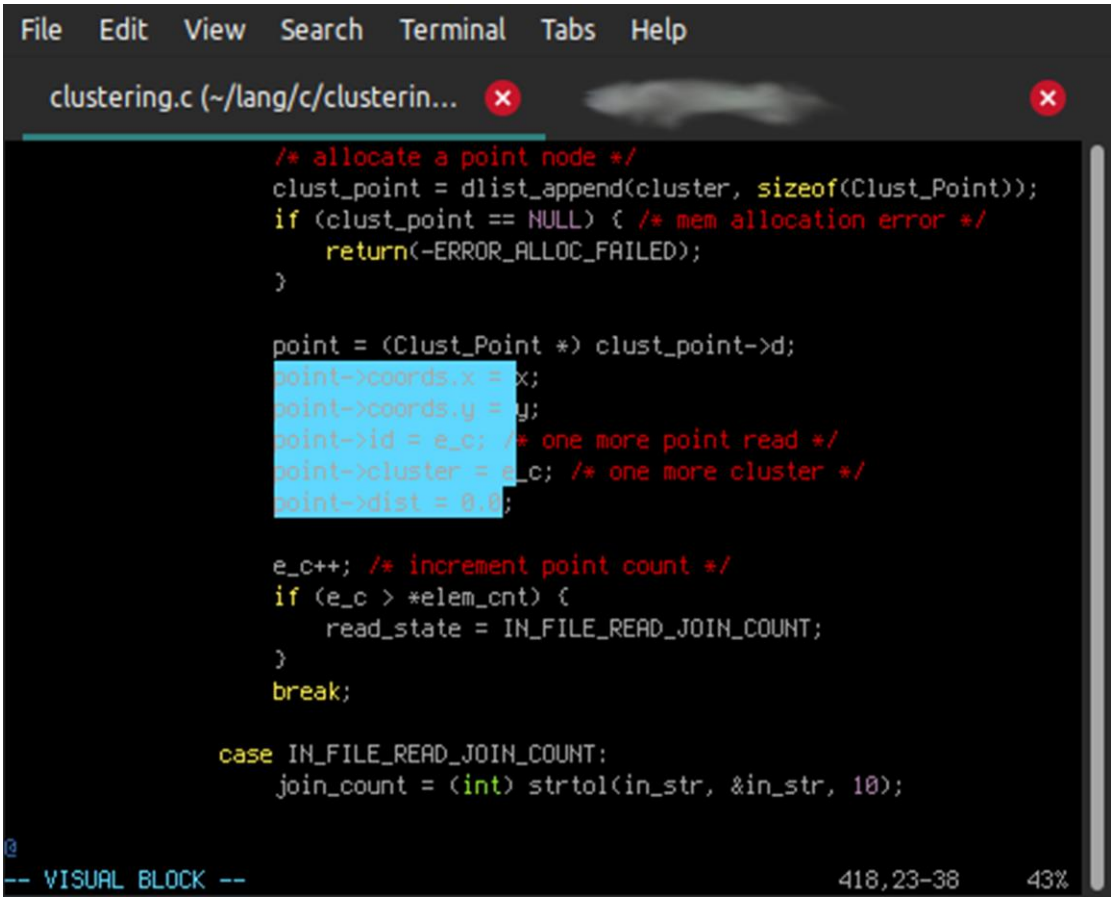
Ο διαφορετικός τρόπος λειτουργίας του vim σε σχέση με τους συνήθεις κειμενογράφους κάνει απαιτητική την εκμάθησή του. Είναι πιο εύκολο να ξεκινήσει

κανείς με τις βασικές λειτουργίες και σταδιακά να επεκταθεί στις πολλές δυνατότητες του vim, αναλόγως και με τις ανάγκες της/του.

Όπως αναφέρθηκε, πατώντας το **i** ο vim μεταβαίνει στη λειτουργία εισαγωγής (insert mode). Πατώντας το **Esc** επιστρέφει στη λειτουργία εντολών. Εκεί, πατώντας το **x** σβήνεται ο χαρακτήρας κάτω από τον δρομέα (cursor). Πατώντας το **u** αναιρείται η πιο πρόσφατη ενέργεια. Η ακολουθία **:w** σώζει το αρχείο, παραμένοντας στη λειτουργία εντολών, ενώ η ακολουθία **:wq** πρώτα σώζει και μετά εξέρχεται από τον vim. Η ακολουθία **:q!** εξέρχεται από τον vim,αλλά χωρίς να σώσει τυχόν αλλαγές μετά το πιο πρόσφατο σώσιμο.

Για να αναδιατάξουμε τα περιεχόμενα του αρχείου που επεξεργαζόμαστε, χρησιμοποιούμε τα πλήκτρα **y** και **p** (από τα yank: τραβώ και put: βάζω, έτσι ονομάζεται στον vim η αντιγραφή και επικόλληση). Πρώτα επιλέγουμε το κείμενο τοποθετώντας το δρομέα κάτω από τον πρώτο προς επιλογή χαρακτήρα και μετά πατώντας τον κατάλληλο συνδυασμό για οπτική λειτουργία, αναλόγως του τρόπου που θέλουμε να επιλέξουμε, π.χ. το **v**, και με τη χρήση των πλήκτρων βελών κάνουμε την επιθυμητή επιλογή. Έχοντας επιλέξει το απόσπασμα που θέλουμε, πατάμε το **y**, που αντιγράφει το απόσπασμα στη μνήμη **χωρίς να το αποκόπτει**. Πάμε με το δρομέα στο σημείο που θέλουμε να επικολλήσουμε την επιλογή και πατάμε το **p**, οπότε αντιγράφεται σε αυτό το σημείο το επιλεγμένο απόσπασμα. Αν θέλουμε να αποκόψουμε και να μεταφέρουμε την επιλογή μας, αντί για το **y** πατάμε το **x.**

Αν θέλουμε να επιλέξουμε ολόκληρες γραμμές (οπτική λειτουργία γραμμής), αντί για το **v** πατάμε το συνδυασμό **Shift+v,** ενώ για ορθογώνιες περιοχές, πατάμε το **Ctrl+v**. Σε αυτές τις περιπτώσεις, στο κάτω μέρος της οθόνης εμφανίζεται η ένδειξη «-- **VISUAL LINE** --» και «-- **VISUAL BLOCK** --», αντίστοιχα. Η επιλογή με visual block είναι ιδιαίτερα χρήσιμη στον προγραμματισμό.



Εικόνα 42 Ο vim στην κατάσταση «visual block», με μια ορθογώνια περιοχή κειμένου επιλεγμένη.

Ένας τρόπος για να μάθει κανείς τον vim είναι με τη χρήση του προγράμματος vimtutor. Αυτό συνήθως περιλαμβάνεται στο πακέτο του vim-enhanced και τρέχοντάς το μπορεί κανείς να ακολουθήσει μια σειρά μαθημάτων, ώστε να μάθει καλά τη βασική χρήση και λειτουργία του vim.

Σημείωση: ο vim, παρά τη δυσκολία που εμφανίζει στην εκμάθησή του, έχει πολλά θετικά στοιχεία. Π.χ. τόσο ο vim, όσο και η «παραθυρική» έκδοσή του (gvim) έχουν πολύ μικρές απαιτήσεις ως προς τη δικτυακή κίνηση. Έτσι, σε περιπτώσεις απομακρυσμένης σύνδεσης (μέσω ssh), ακόμα και με αργές συνδέσεις μπορεί κανείς να επεξεργάζεται κυριολεκτικά δεκάδες αρχεία ταυτόχρονα και χωρίς ενοχλητικές καθυστερήσεις στην απόκρισή του.

3.3 Χρήση μεταβλητών φλοιού

Ο φλοιός Bash δίνει τη δυνατότητα ορισμού *μεταβλητών φλοιού*. Μπορούν να φανούν χρήσιμες τόσο κατά την εκτέλεση εντολών ή προγραμμάτων σεναρίου φλοιού (shell scripts), όσο και για την αλλαγή της συμπεριφοράς του φλοιού. Μπορούν να

υποκαταστήσουν ορίσματα στις εντολές ή να θέσουν μια κοινή ρύθμιση για εντολές που τρέχουν από το φλοιό.

Μια κατηγορία τους είναι οι *μεταβλητές περιβάλλοντος* (environment variables). Αυτές είναι μεταβλητές που παίρνουν τιμή κατά την είσοδο στο λογαριασμό (log in), οπότε και ξεκινάει ο Bash, και είναι διαθέσιμες σε όλα τα προγράμματα που τρέχουν από τη συγκεκριμένη συνεδρία.

Οι τιμές τους μπορούν να αλλάξουν με μια απλή ανάθεση, αλλά η αλλαγή ισχύει μόνο για τη συγκεκριμένη συνεδρία. Αν έχουμε ανοιχτά ταυτόχρονα δύο παράθυρα τερματικού και αλλάξουμε την τιμή μιας μεταβλητής στο ένα παράθυρο (ή δημιουργήσουμε μια νέα μεταβλητή), **η αλλαγή ισχύει μόνο στο τερματικό και τη συνεδρία που το κάναμε**. Η τιμή της ίδιας μεταβλητής στο άλλο τερματικό **δεν** επηρεάζεται.

3.3.1 Ανάθεση τιμών σε μεταβλητές

Η ανάθεση τιμής σε μεταβλητή γίνεται με την εξής σύνταξη:

```
VAR_NAME=value
```

Τα ονόματα μεταβλητών μπορούν να περιέχουν μικρά ή κεφαλαία γράμματα, ψηφία και τον χαρακτήρα υπογράμμισης (κάτω παύλα, «_»). Επισημαίνεται ότι **δεν πρέπει να υπάρχει κενό ούτε αριστερά, ούτε δεξιά του χαρακτήρα «=»**. Μερικά παραδείγματα:

```
$ ANSWER=42

$ server_name=srv123

$ tempfile_1=/tmp/foo

$ _DEST_OS=RHEL8
```

Η εντολή set δίνει μια λίστα από όλες τις μεταβλητές φλοιού που έχουν οριστεί την τρέχουσα στιγμή. Μαζί εμφανίζονται και όλες οι συναρτήσεις φλοιού, που μπορούν να αγνοηθούν:

```
$ set | less
```

Χρησιμοποιούμε την εντολή less επειδή η έξοδος της εντολής μπορεί να είναι μακροσκελής.

Για να δούμε την τιμή της μεταβλητής, πρέπει να βάλουμε μπροστά από το όνομα της μεταβλητής τον χαρακτήρα «$» (ονομάζεται «επέκταση της μεταβλητής»), αλλιώς απλώς τυπώνεται σαν αλφαριθμητικό το όνομα της μεταβλητής:

```
$ echo ANSWER
ANSWER
$ echo $ANSWER
42
```

4. Διαχείριση τοπικών χρηστών και ομάδων

Μαθησιακοί στόχοι

Οι επιμορφούμενοι θα πρέπει να είναι σε θέση να:

- Κατανοούν τη σκοπιμότητα ύπαρξης χρηστών και ομάδων χρηστών σε ένα σύστημα Linux
- Μεταβαίνουν σε λογαριασμό υπερχρήστη (superuser) ώστε να διαχειρίζονται ένα σύστημα Linux, καθώς και να εκχωρούν το δικαίωμα αυτό σε άλλους χρήστες με χρήση της εντολής **`sudo`**
- Δημιουργούν, διαχειρίζονται και να διαγράφουν χρήστες και ομάδες
- Διαχειρίζονται τοπικές πολιτικές συνθηματικών των χρηστών

4.1 Περιγραφή του σκοπού των χρηστών και ομάδων σε ένα σύστημα Linux

Σε ένα οποιοδήποτε λειτουργικό σύστημα, οι λογαριασμοί χρηστών εξυπηρετούν τη σύνδεση χρηστών στο σύστημα ή/και την πρόσβασή τους στους πόρους του συστήματος. Ο λογαριασμός χρήστη χρησιμοποιείται επίσης για τον καθορισμό ορίων ασφαλείας μεταξύ διαφορετικών ατόμων και προγραμμάτων που μπορούν να εκτελέσουν εντολές.

Οι χρήστες έχουν ονόματα χρήστη έτσι ώστε να αντιστοιχίζονται σε πραγματικούς ανθρώπους και να διευκολύνεται έτσι η ταυτοποίησή τους. Εσωτερικά, το σύστημα διακρίνει τους λογαριασμούς χρηστών από τον μοναδικό αριθμό αναγνώρισης που τους έχει εκχωρηθεί. Ο αριθμός αυτός ονομάζεται αναγνωριστικό χρήστη (ή **User ID** ή **UID**). Εάν ένας λογαριασμός χρήστη προορίζεται για χρήση από ανθρώπους, του εκχωρείται ένας μυστικός κωδικός πρόσβασης, ο οποίος χρησιμοποιείται από τον χρήστη κατά τη σύνδεσή του ώστε να αποδεικνύει ότι είναι ο πραγματικός εξουσιοδοτημένος χρήστης.

Οι λογαριασμοί χρηστών είναι θεμελιώδεις για την ασφάλεια του συστήματος. Κάθε διεργασία (πρόγραμμα που εκτελείται) στο σύστημα εκτελείται ως κάποιος συγκεκριμένος χρήστης. Επίσης, κάθε αρχείο έχει έναν συγκεκριμένο χρήστη ως κάτοχό του. Η έννοια της ιδιοκτησίας του αρχείου χρησιμοποιείται από το σύστημα ώστε να ελέγχει την πρόσβαση των χρηστών στους πόρους του συστήματος. Ο χρήστης που σχετίζεται με μια διεργασία που εκτελείται, καθορίζει τα αρχεία και τους καταλόγους στους οποίους έχει πρόσβαση η διεργασία αυτή.

Υπάρχουν τρεις κύριοι τύποι λογαριασμών χρήστη: ο υπερχρήστης (superuser), οι χρήστες συστήματος και οι κανονικοί χρήστες.

- Ο λογαριασμός υπερχρήστη είναι αυτός που έχει πλήρη πρόσβαση στο σύστημα ώστε να μπορεί να το διαχειρίζεται. Το όνομα του υπερχρήστη είναι *root* και ο λογαριασμός έχει *UID 0*.
- Το λειτουργικό σύστημα διαθέτει λογαριασμούς συστήματος που χρησιμοποιούνται από διεργασίες που παρέχουν υποστηρικτικές υπηρεσίες. Αυτές οι διεργασίες, ή daemons (δαίμονες), συνήθως δε χρειάζεται να εκτελούνται με δικαιώματα *superuser*. Επομένως, τους εκχωρούνται τα δικαιώματα εκείνα που είναι απαραίτητα ώστε να εκτελούν τις λειτουργίες για τις οποίες έχουν δημιουργηθεί και που ταυτόχρονα τους επιτρέπουν να προστατεύουν τα αρχεία τους και άλλους πόρους από μη εξουσιοδοτημένη πρόσβαση από άλλες διεργασίες και από απλούς χρήστες του συστήματος. Ένας χρήστης δε μπορεί να συνδεθεί διαδραστικά στο σύστημα (log in) χρησιμοποιώντας έναν λογαριασμό συστήματος.
- Οι περισσότεροι χρήστες έχουν απλούς (regular) λογαριασμούς χρηστών τους οποίους χρησιμοποιούν για την καθημερινή τους εργασία. Όπως και οι λογαριασμοί συστήματος, οι απλοί χρήστες έχουν περιορισμένη πρόσβαση στο σύστημα.

Με χρήση της εντολής **id** χωρίς παραμέτρους, ο χρήστης που είναι συνδεδεμένος εκείνη τη στιγμή στο σύστημα μπορεί να δει πληροφορίες για τον λογαριασμό του όπως για παράδειγμα το UID, το GID και τις ομάδες (groups) στις οποίες ανήκει:

```
[user01@host ~]$ id
uid=1000(user01) gid=1000(user01) groups=1000(user01)
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Για να δούμε αντίστοιχες πληροφορίες για κάποιον άλλον λογαριασμό, μπορούμε στην εντολή **id** να περάσουμε ως παράμετρο το όνομα χρήστη του άλλου λογαριασμού:

```
[user01@host]$ id user02
uid=1002(user02) gid=1001(user02) groups=1001(user02)
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Για να δούμε τον κάτοχο κάποιου **αρχείου** χρησιμοποιούμε την εντολή **ls -l**. Για να δούμε τον κάτοχο ενός **καταλόγου** χρησιμοποιούμε την εντολή **ls -ld**.

Στα ακόλουθα παραδείγματα, το **όνομα χρήστη** φαίνεται στην τρίτη στήλη:

```
[user01@host ~]$ ls -l file1
-rw-rw-r--. 1 user01 user01 0 Feb 5 11:10 file1
[user01@host]$ ls -ld dir1
drwxrwxr-x. 2 user01 user01 6 Feb 5 11:10 dir1
```

Για να δούμε πληροφορίες σχετικές με τις διεργασίες που εκτελούνται, χρησιμοποιούμε την εντολή **ps**. Η προεπιλογή είναι να εμφανίζονται μόνο οι διεργασίες που εκτελούνται στον τρέχοντα φλοιό (*shell*). Με προσθήκη της επιλογής **a** εμφανίζονται όλες οι διεργασίες που εκτελούνται στο συγκεκριμένο τερματικό. Για να δούμε τον χρήστη που σχετίζεται με κάθε διεργασία που εκτελείται, προσθέτουμε την επιλογή **u**. Στο ακόλουθο παράδειγμα, το όνομα χρήστη εμφανίζεται στην πρώτη στήλη:

```
[user01@host]$ ps -au
USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
root 777 0.0 0.0 225752 1496 tty1 Ss+ 11:03 0:00 /sbin/agetty -o -p -- \u --noclear tty1 linux
root 780 0.0 0.1 225392 2064 ttyS0 Ss+ 11:03 0:00 /sbin/agetty -o -p -- \u --keep-baud 115200,38400,9600
user01 1207 0.0 0.2 234044 5104 pts/0 Ss 11:09 0:00 -bash
user01 1319 0.0 0.2 266904 3876 pts/0 R+ 11:33 0:00 ps au
```

Στο παραπάνω παράδειγμα εμφανίζεται το όνομα χρήστη και όχι το *UID*. Το λειτουργικό σύστημα βέβαια αναφέρεται σε κάθε χρήστη με βάση το *UID* του και όχι το όνομά του. Η αντιστοίχιση ονόματος χρήστη → *UID* αποθηκεύεται στο αρχείο **/etc/passwd**.

Κάθε γραμμή του αρχείου **/etc/passwd** περιέχει πληροφορίες σχετικές με τον χρήστη. Αποτελείται από 7 πεδία που χωρίζονται μεταξύ τους με άνω και κάτω τελεία (*colon*). Η πληροφορία που περιέχεται σε κάθε πεδίο παρατίθεται ακολούθως:

```
1user01: 2x: 31000: 41000: 5User One: 6/home/user01: 7/bin/bash
```

1. Το όνομα χρήστη
2. Σε παλιότερες διανομές Linux, σε αυτό το πεδίο αποθηκευόταν ο κωδικός χρήστη σε κρυπτογραφημένη μορφή. Πλέον, ο κωδικός αποθηκεύεται στο αρχείο ***/etc/shadow***. Στο αρχείο ***/etc/passwd*** το πεδίο αυτό έχει πάντα την τιμή **x**.
3. Το αναγνωριστικό χρήστη (***UID***)
4. Το αναγνωριστικό ομάδας για τον χρήστη (***GID***). Οι ομάδες θα συζητηθούν στην παράγραφο 4.4
5. Το πραγματικό ονοματεπώνυμο του χρήστη
6. Ο αρχικός (*home*) **κατάλογος** του χρήστη. Πρόκειται για τον αρχικό κατάλογο κατά την έναρξη του φλοιού. Περιέχει τα προσωπικά αρχεία του χρήστη, καθώς και ρυθμίσεις για τον χρήστη αυτόν.
7. Ο προκαθορισμένος **φλοιός** ο οποίος εκτελείται για τον συγκεκριμένο χρήστη κατά τη σύνδεσή του στο σύστημα. Για τους απλούς χρήστες, πρόκειται συνήθως για τον φλοιό **/bin/bash** και είναι το πρόγραμμα εκείνο που παρέχει το περιβάλλον εκτέλεσης εντολών. Για κάποιον λογαριασμό

συστήματος, για τον οποίο για παράδειγμα δεν επιτρέπεται η σύνδεση με όνομα χρήστη/κωδικό πρόσβασης, το πεδίο αυτό θα μπορούσε να έχει την τιμή **`/sbin/nologin`**.

Η έννοια της ομάδας (group)

Μια ομάδα είναι ένα σύνολο χρηστών οι οποίοι πρέπει από κοινού να έχουν πρόσβαση σε αρχεία, αλλά και σε άλλου είδους πόρους του συστήματος. Αντί λοιπόν να εκχωρείται δικαίωμα πρόσβασης για κάθε πόρο σε κάθε έναν χρήστη ξεχωριστά, το δικαίωμα εκχωρείται σε επίπεδο ομάδας.

Όπως και με τους χρήστες, έτσι και με τις ομάδες, σε κάθε ομάδα αποδίδεται ένα όνομα. Ομοίως, το σύστημα αναγνωρίζει την κάθε ομάδα με τον μοναδικό αναγνωριστικό αριθμό της, **Group ID** ή **GID**.

Το αρχείο στο οποίο το σύστημα αποθηκεύει την αντιστοιχία ομάδα → GID, καθώς και άλλες πληροφορίες για τις τοπικές ομάδες του συστήματος, είναι εξ ορισμού το αρχείο **`/etc/group`**. Κάθε γραμμή του αρχείου περιέχει πληροφορίες για μία ομάδα και αποτελείται από τέσσερα πεδία, διαχωρισμένα μεταξύ τους με άνω και κάτω τελεία:

```
1group01: 2x: 310000: 4user01,user02,user03
```

1. Το **όνομα** της ομάδας
2. Δε χρησιμοποιείται πλέον και στα περισσότερα συστήματα περιέχει την τιμή **`x`**
3. Το **GID** της συγκεκριμένη ομάδας
4. Τα ονόματα χρήστη των **μελών** της ομάδας

Πρωτεύουσες (Primary) και Δευτερεύουσες (Supplementary) ομάδες

Στα συστήματα Linux ένας χρήστης μπορεί να ανήκει σε δύο είδη ομάδων: μία κύρια ομάδα (*primary group*) και, προαιρετικά, σε μία ή περισσότερες δευτερεύουσες (συμπληρωματικές) ομάδες (*supplementary* ή *secondary groups*).

Η **κύρια ομάδα** είναι εκείνη στην οποία ανήκουν τα αρχεία που δημιουργούνται από τον χρήστη. Συνήθως, το όνομα της κύριας ομάδας, είναι το ίδιο με το όνομα του χρήστη. Όλοι οι χρήστες πρέπει υποχρεωτικά να ανήκουν σε (μόνο) μία κύρια ομάδα.

Οι **δευτερεύουσες ομάδες** χρησιμοποιούνται για να δώσουν επιπλέον δικαιώματα στους χρήστες που ανήκουν σε αυτές. Δεν είναι υποχρεωτικό κάποιος χρήστης να ανήκει και σε δευτερεύουσες ομάδες, ενώ μπορεί να ανήκει σε μία ή περισσότερες από αυτές. Οι ομάδες αυτές χρησιμοποιούνται για λόγους ευελιξίας, ώστε οι χρήστες που ανήκουν σε αυτές να αποκτούν επιπλέον δικαιώματα (ανάγνωσης, εγγραφής, εκτέλεσης) σε κάποιον πόρο του συστήματος.

Προκειμένου να αλλάξουμε κύρια ομάδα σε κάποιον χρήστη, ή να τον προσθέσουμε και σε μία ή περισσότερες δευτερεύουσες, θα πρέπει αυτή/ές να υπάρχει/ουν ήδη.

Χρησιμοποιώντας την εντολή **id** μπορούμε να δούμε και τις συμπληρωματικές ομάδες στις οποίες ενδεχομένως είναι μέλος ένας χρήστης:

```
[user01@localhost ~]$ id

uid=1000(user01) gid=1000(user01) groups=1000(user01),10(wheel)
```

Στο παράδειγμα αυτό, ο χρήστης *user01* ανήκει στην κύρια ομάδα ***user01*** (GID 1000) και στη δευτερεύουσα ομάδα ***wheel*** (GID 10).

Ερωτήσεις κατανόησης

1. Η κύρια ομάδα ενός χρήστη στο Linux:
    a. Είναι η πρώτη ομάδα που δημιουργείται στο σύστημα
    b. **Καθορίζει την πρόσβαση και τα δικαιώματά του**
    c. Είναι η ομάδα στην οποία ανήκει ο διαχειριστής του συστήματος

2. Οι δευτερεύουσες ομάδες σε ένα σύστημα Linux αναφέρονται σε:
    a. Ομάδες χρηστών που δημιουργούνται μετά την πρωτεύουσα ομάδα
    b. Ομάδες σχετικές με πόρους του συστήματος
    c. **Ομάδες που παρέχουν στους χρήστες μέλη τους επιπλέον δικαιώματα και πρόσβαση σε αρχεία και πόρους**

3. Τι είναι το UID ενός λογαριασμού στο Linux;
    a. Ένας κωδικός που παρέχεται στον χρήστη από τον διαχειριστή του συστήματος
    b. **Ένας μοναδικός αριθμός που ταυτοποιεί τον χρήστη στο σύστημα**
    c. Το όνομα χρήστη του λογαριασμού

4. Ποια είναι η λειτουργία του αρχείου "/etc/group" στο Linux;
    a. **Αποθηκεύει πληροφορίες σχετικά με τις ομάδες του συστήματος**
    b. Αποθηκεύει τους κωδικούς πρόσβασης των χρηστών
    c. Αποθηκεύει τη λίστα των αρχείων και καταλόγων στα οποία έχει πρόσβαση η κάθε ομάδα του συστήματος

5. Σε ποιο από τα παρακάτω υπάρχουν τα προσωπικά αρχεία του χρήστη;
    a. **Στον αρχικό κατάλογο του χρήστη (home)**
    b. Στο /bin/bash
    c. Στο /etc/passwd
    d. Στο /etc/group

4.2 Μετάβαση στον λογαριασμό διαχειριστή

Στις διάφορες διανομές Linux, όπως και στα περισσότερα λειτουργικά συστήματα, υπάρχει ένας λογαριασμός διαχειριστή ο οποίος διαθέτει πλήρη διαχειριστικά προνόμια στο σύστημα. Ο λογαριασμός διαχειριστή είναι συνήθως ο **`root`**.

Ο λογαριασμός διαχειριστή έχει απεριόριστη πρόσβαση και δικαιώματα πάνω σε όλους τους πόρους του συστήματος. Μπορεί να εκτελέσει οποιαδήποτε ενέργεια, μεταξύ των οποίων η εγκατάσταση λογισμικού, η διαχείριση και μορφοποίηση των αποθηκευτικών μέσων και η διαχείριση χρηστών.

Ο λογαριασμός διαχειριστή αναγνωρίζεται με τον **User ID (UID) 0**. Κανένας άλλος λογαριασμός χρήστη δεν μπορεί να έχει UID 0.

Ο λογαριασμός root στα συστήματα Linux είναι το αντίστοιχο του λογαριασμού Administrator των Microsoft Windows.

Λόγω των προνομίων του ο λογαριασμός διαχειριστή πρέπει να χρησιμοποιείται με προσοχή και υπευθυνότητα. Η κακή χρήση του λογαριασμού διαχειριστή μπορεί να οδηγήσει σε διαφόρων ειδών προβλήματα (π.χ. διαγραφή αρχείων και φακέλων, διαγραφή λογαριασμών χρηστών κ.α.) ή σε παραβίαση της ασφάλειας του συστήματος.

Για τους παραπάνω λόγους, συνιστάται οι διαχειριστές του συστήματος να συνδέονται στο σύστημα με απλούς λογαριασμούς, χωρίς αυξημένα δικαιώματα, για τις καθημερινές δραστηριότητες και με χρήση διαφόρων μηχανισμών να μεταβαίνουν σε ρόλο διαχειριστή προσωρινά και μόνο όταν αυτό είναι απαραίτητο.

Μετάβαση σε λογαριασμό άλλου χρήστη

Η εντολή "**`su`**" (συντομογραφία των λέξεων "***s****witch* ***u****ser*") μας επιτρέπει να εκτελούμε εντολές ως κάποιος άλλος χρήστης. Εάν δε, διαθέτουμε τα απαραίτητα δικαιώματα, μπορούμε να εκτελέσουμε εντολές ως root ή να συνδεθούμε στο σύστημα ως root.

Στο ακόλουθο παράδειγμα, ο χρήστης που είναι συνδεδεμένος στο σύστημα ως *user01*, συνδέεται σε αυτό ως *user02* (με την προϋπόθεση φυσικά ότι γνωρίζει και πληκτρολογεί τον κωδικό του χρήστη *user02* όταν του ζητείται):

```
[user01@host]$ su - user02
Password:
```

```
[user02@host]$
```

Εάν παραλείψουμε το όνομα χρήστη ως παράμετρο, η εντολή **su** ή **su –** επιχειρεί να μας συνδέσει εξ ορισμού ως root:

```
[user01@host]$ su -
Password:
[root@host]#
```

Η διαφορά μεταξύ των εντολών **su** και **su -** στο Linux αφορά την επιλογή των ρυθμίσεων περιβάλλοντος που φορτώνονται κατά την εναλλαγή στον λογαριασμό του νέου χρήστη.

Η εντολή **su** μας επιτρέπει να συνδεόμαστε στο σύστημα ως άλλος χρήστης, αλλά δε μεταβαίνει πλήρως στο περιβάλλον του νέου χρήστη. Αυτό σημαίνει ότι οι μεταβλητές περιβάλλοντος, όπως οι μεταβλητές PATH και οι ρυθμίσεις καταλόγων, δεν αλλάζουν. Με άλλα λόγια, συνεχίζουν να ισχύουν το PATH, οι ρυθμίσεις καταλόγων και άλλες προτιμήσεις που ήταν ενεργές για τον **αρχικό χρήστη**.

Αντίθετα, η εντολή **su -** (με την παύλα) εκτελεί μια πλήρη εναλλαγή στον νέο χρήστη και φορτώνει τις **ρυθμίσεις περιβάλλοντος του νέου χρήστη**. Αυτό σημαίνει ότι οι μεταβλητές περιβάλλοντος, οι ρυθμίσεις καταλόγων και οι άλλες προτιμήσεις φορτώνονται από το προφίλ του νέου χρήστη. Με αυτόν τον τρόπο, έχουμε πρόσβαση σε όλες τις ρυθμίσεις που έχουν οριστεί για τον χρήστη αυτόν, καθιστώντας το περιβάλλον εργασίας πλήρες.

Έτσι, η εντολή **su -** χρησιμοποιείται συχνά για την εκτέλεση εντολών ως άλλος χρήστης με πλήρη διαμόρφωση του περιβάλλοντος, ενώ η εντολή **su** χρησιμοποιείται για γρήγορη εναλλαγή χρήστη χωρίς να αλλάζει το περιβάλλον εργασίας.

Κατά συνέπεια, ο διαχειριστής ενός συστήματος, ενδείκνυται να χρησιμοποιεί κυρίως την εντολή **su –** ώστε να συνδέεται σε έναν φλοιό ο οποίος έχει όλες τις ρυθμίσεις περιβάλλοντος του νέου χρήστη.

Εκτέλεση εντολών με το εργαλείο sudo

Πολλές φορές, για λόγους ασφαλείας, ο χρήστης `root` δεν έχει καν κωδικό πρόσβασης. Κατά συνέπεια, οι χρήστες δε μπορούν να συνδεθούν στο σύστημα απευθείας ως `root`, αλλά ούτε μπορούν να συνδεθούν ως `root` με χρήση της εντολής `su`. Σε αυτή την περίπτωση, προκειμένου ένας χρήστης να αποκτήσει δικαιώματα `root`, μπορεί να χρησιμοποιήσει το εργαλείο **sudo**.

Σε αντίθεση με την εντολή `su`, το `sudo` ζητά από τον χρήστη να εισάγει **τον δικό του κωδικό**, και όχι τον κωδικό του χρήστη στον λογαριασμό του οποίου προσπαθεί να αποκτήσει πρόσβαση. Επίσης, με χρήση ανάλογων ρυθμίσεων, το `sudo` μπορεί να επιτρέπει σε έναν χρήστη να εκτελεί όλες, ή μέρος των εντολών που κάποιος άλλος χρήστης μπορεί να εκτελέσει.

Για παράδειγμα, εάν το **sudo** έχει ρυθμιστεί ώστε να επιτρέπει στον χρήστη *user01* να εκτελεί την εντολή **usermod** ως `root`, ο *user01* θα μπορούσε να εκτελέσει την ακόλουθη εντολή ώστε να κλειδώσει (**usermod -L**) ή να ξεκλειδώσει τον λογαριασμό κάποιου άλλου χρήστη, για παράδειγμα του χρήστη *user02*. Μετά το κλείδωμα του λογαριασμού του *user02* δεν είναι δυνατή η σύνδεσή του στο σύστημα:

```
[user01@host ~]$ sudo usermod -L user02
[sudo] password for user01:
[user01@host ~]$ su - user02
Password:
su: Authentication failure
[user01@host ~]$
```

Στην περίπτωση που κάποιος χρήστης προσπαθήσει να εκτελέσει κάποια εντολή ως άλλος χρήστης ενώ οι ρυθμίσεις του `sudo` δεν το επιτρέπουν, εκτός του ότι η εντολή δεν εκτελείται, η απόπειρα καταγράφεται από το σύστημα και συνήθως αποστέλλεται και σχετικό ενημερωτικό email προς τον `root`:

```
[user02@host ~]$ sudo tail /var/log/secure
[sudo] password for user02:
user02 is not in the sudoers file. This incident will be reported.
[user02@host ~]$
```

Επιπρόσθετα, όλες οι εντολές που εκτελούνται με χρήση του **sudo** καταγράφονται εξ ορισμού στο αρχείο **/var/log/secure**.

Τέλος, στις περισσότερες διανομές Linux, οι λογαριασμοί χρηστών που ανήκουν στην ομάδα **wheel**, μπορούν να εκτελούν εντολές με χρήση του `sudo` ως οποιοσδήποτε χρήστης, συμπεριλαμβανομένου του *root*. Αυτό ισχύει πλέον και για το Red Hat Enterprise Linux (έκδοση 7 ή νεότερη). Έως και την έκδοση 6 του Red Hat Enterprise Linux, η ομάδα `wheel` δεν είχε κάποια ειδικά δικαιώματα.

Σε άλλες διανομές Linux, όπως π.χ. στο *Ubuntu*, η αντίστοιχη ομάδα ονομάζεται **sudo**.

Πρόσβαση σε διαδραστικό φλοιό Root με χρήση του sudo

Εάν ένας μη διαχειριστικός λογαριασμός του συστήματος έχει δικαίωμα να εκτελέσει την εντολή **su** με χρήση του **sudo**, μπορεί να αποκτήσει πρόσβαση στον φλοιό του χρήστη root εκτελώντας την εντολή **sudo su -**. Αυτό συμβαίνει διότι το `sudo` θα εκτελέσει την εντολή `su -` ως root και ο root δε χρειάζεται κωδικό για εκτελέσει την εντολή `su`.

Διαμόρφωση του sudo

Το βασικό αρχείο ρυθμίσεων του `sudo` είναι το **`/etc/sudoers`**. Το αρχείο αυτό θα πρέπει, προς αποφυγή προβλημάτων, να τροποποιείται **μόνο** με χρήση της εντολής **`visudo`**.

Ακολούθως φαίνεται μία γραμμή από το αρχείο `/etc/sudoers`, η οποία επιτρέπει την πρόσβαση στο `sudo` στα μέλη της ομάδας `wheel`:

```
%wheel      ALL=(ALL)    ALL
```

Στο παράδειγμά μας, **%wheel** είναι η ομάδα (ή ο χρήστης) τον οποίο αφορά η ρύθμιση. Το σύμβολο **%** υποδεικνύει ότι αναφερόμαστε σε ομάδα. Το **ALL=(ALL)** σημαίνει ότι τα μέλη της ομάδας *wheel* μπορούν να εκτελούν οποιαδήποτε εντολή, σε οποιοδήποτε σύστημα έχει το συγκεκριμένο αρχείο. Το τελικό **ALL** σημαίνει ότι τα μέλη της ομάδας μπορούν να εκτελούν τις εντολές ως οποιοσδήποτε χρήστης.

Στις περισσότερες διανομές Linux, το αρχείο ***/etc/sudoers*** λαμβάνει υπόψη του και τα αρχεία ρυθμίσεων που ενδεχομένως υπάρχουν στον κατάλογο ***/etc/sudoers.d*** (ο χαρακτήρας **#** στην αρχή της γραμμής είναι μέρος της σύνταξης και δεν υποδηλώνει σχόλιο):

```
## Read drop-in files from /etc/sudoers.d (the # here does not
mean a comment)
#includedir /etc/sudoers.d
```

Έτσι ένας διαχειριστής, μπορεί να επιτρέπει πρόσβαση *sudo* σε κάποιον χρήστη, απλά δημιουργώντας το αντίστοιχο αρχείο στον συγκεκριμένο κατάλογο. Αυτή η πρακτική είναι ιδιαίτερα χρήσιμη και συνιστάται καθώς, εκτός των άλλων, τα αρχεία του καταλόγου αυτού διατηρούνται ακόμα και στις περιπτώσεις αναβάθμισης του συστήματος, κάτι που δεν ισχύει για το αρχείο */etc/sudoers*.
Για παράδειγμα, προκειμένου να δώσουμε πρόσβαση sudo στον χρήστη *user01*, θα μπορούσαμε να δημιουργήσουμε το αρχείο **/etc/sudoers.d/user01** με περιεχόμενα:

```
user01      ALL=(ALL)    ALL
```

Αντίστοιχα, θα μπορούσαμε να δημιουργήσουμε το αρχείο **/etc/sudoers.d/group01** ώστε να δώσουμε πρόσβαση sudo στην ομάδα **group01**:

```
%group01    ALL=(ALL)    ALL
```

Προσοχή θα πρέπει να δίνεται στην ονομασία των αρχείων αυτών, καθώς στις περισσότερες διανομές Linux, αρχεία τα οποία περιέχουν στο όνομά τους τον χαρακτήρα '**.**' ή που το όνομά τους τελειώνει με τον χαρακτήρα '**~**' αγνοούνται.

Τέλος, είναι εφικτό να ρυθμίσουμε το *sudo* ώστε να επιτρέπει σε έναν χρήστη (π.χ. στον *user01*) να εκτελεί εντολές ως οποιοσδήποτε άλλος χρήστης, χωρίς να χρειάζεται να εισάγει τον κωδικό του:

```
user01      ALL=(ALL) NOPASSWD:ALL
```

4.3 Δημιουργία, τροποποίηση και διαγραφή τοπικά ορισμένων λογαριασμών χρηστών

Σε ένα σύστημα Linux υπάρχουν αρκετά εργαλεία της γραμμής εντολών για τη διαχείριση των τοπικών λογαριασμών χρηστών.

Δημιουργία χρηστών από τη γραμμή εντολών

Η εντολή **useradd *username*** δημιουργεί ένα νέο χρήστη με όνομα *username*. Ρυθμίζει τον αρχικό κατάλογο (home directory) του χρήστη και τις πληροφορίες λογαριασμού και δημιουργεί μια ιδιωτική ομάδα για τον χρήστη με όνομα το όνομα χρήστη. Σε αυτό το σημείο ο λογαριασμός δεν έχει κάποιον έγκυρο κωδικό πρόσβασης και ο χρήστης δεν μπορεί να συνδεθεί στο σύστημα μέχρι να οριστεί κάποιος κωδικός πρόσβασης.

Η εντολή **useradd --help** εμφανίζει τις βασικές επιλογές που μπορούν να χρησιμοποιηθούν για την παράκαμψη των προεπιλογών. Στις περισσότερες περιπτώσεις, οι ίδιες επιλογές μπορούν να χρησιμοποιηθούν με την εντολή **usermod** για την τροποποίηση ενός υπάρχοντος χρήστη.

Ορισμένες προεπιλογές, όπως το εύρος των έγκυρων αριθμών UID και οι προεπιλεγμένοι κανόνες γήρανσης κωδικού πρόσβασης, ορίζονται στο αρχείο **/etc/login.defs**. Οι τιμές σε αυτό το αρχείο χρησιμοποιούνται μόνο κατά τη δημιουργία νέων χρηστών. Οποιαδήποτε αλλαγή σε αυτό το αρχείο δεν επηρεάζει τους υπάρχοντες χρήστες.

Τροποποίηση υπάρχοντος χρήστη από τη γραμμή εντολών

Εκτελώντας την εντολή **usermod --help**, μπορούμε να δούμε τις διαθέσιμες επιλογές που μπορούμε να χρησιμοποιήσουμε με τη συγκεκριμένη εντολή. Μερικές από τις πιο συνηθισμένες φαίνονται παρακάτω:

-c, --comment COMMENT	Εισαγωγή σχολίου για τον λογαριασμό χρήστη (π.χ. το πραγματικό του όνομα)
-d, --home HOME_DIR	Ορισμός νέου αρχικού καταλόγου χρήστη
-m, --move-home	Μετακίνηση αρχικού καταλόγου χρήστη σε νέα θέση. Πρέπει να χρησιμοποιείται με την επιλογή **-d**
-g, --gid GROUP	Ορισμός κύριας ομάδας χρήστη
-G, --groups GROUPS	Ορισμός των συμπληρωματικών ομάδων στις οποίες θα ανήκει ο χρήστης
-a, --append	(συνήθως χρησιμοποιείται με την επιλογή **-G**) προσθήκη χρήστη **και** σε άλλες ομάδες, πέρα από εκείνες στις οποίες ανήκει ήδη
-l, --login NEW_LOGIN	Νέο όνομα σύνδεσης χρήστη (το όνομα αρχικού καταλόγου του χρήστη δεν επηρεάζεται)
-s, --shell SHELL	Νέος φλοιός σύνδεσης χρήστη
-u, --uid UID	Νέο UID χρήστη
-L, --lock	Κλείδωμα του λογαριασμού χρήστη
-U, --unlock	Ξεκλείδωμα του λογαριασμού χρήστη

Διαγραφή χρηστών από τη γραμμή εντολών

- Για να διαγράψουμε έναν χρήστη και τις πληροφορίες που περιέχονται για αυτόν στο αρχείο **/etc/passwd**, χωρίς όμως να διαγράψουμε τον αρχικό του κατάλογο (home directory), χρησιμοποιούμε την εντολή **`userdel` *`username`***.
- Εάν θέλουμε να διαγράψουμε και τον αρχικό κατάλογο του χρήστη, χρησιμοποιούμε την εντολή **`userdel -r` *`username`***

Ορισμός κωδικού χρήστη από τη γραμμή εντολών

- Για να ορίσουμε νέο κωδικό πρόσβασης, ή να αλλάξουμε τον υπάρχοντα κωδικό πρόσβασης για κάποιον χρήστη, χρησιμοποιούμε την εντολή **`passwd` *`username`***.
- Ο χρήστης **root** μπορεί να θέσει οποιονδήποτε κωδικό πρόσβασης για οποιονδήποτε χρήστη, ακόμα και αν ο κωδικός αυτός δεν πληροί τις προϋποθέσεις του συστήματος σχετικά με το μήκος και την πολυπλοκότητα των

κωδικών. Σε μια τέτοια περίπτωση, εμφανίζεται ένα σχετικό προειδοποιητικό μήνυμα, όμως ο ορισμός του κωδικού πραγματοποιείται κανονικά.

```
[root@host ~]# passwd user01
Changing password for user user01.
New password:
BAD PASSWORD: The password fails the dictionary check - it is based on a dictionary word
Retype new password:
Passwd: all authentication tokens updated successfully.
[root@host ~]#
```

Ένας απλός χρήστης μπορεί να αλλάξει κωδικό πρόσβασης μόνο για τον δικό του λογαριασμό.

4.4 Δημιουργία, τροποποίηση και διαγραφή τοπικά ορισμένων λογαριασμών ομάδων

Όπως αναφέρθηκε και νωρίτερα, προκειμένου να αλλάξουμε κύρια ομάδα σε κάποιον χρήστη, ή να τον προσθέσουμε και σε μία ή περισσότερες δευτερεύουσες, θα πρέπει αυτή/ές να υπάρχει/ουν ήδη.

Ας δούμε μερικές από τις εντολές που μπορούμε να χρησιμοποιήσουμε για τη διαχείριση ομάδων χρηστών.

Δημιουργία ομάδων από τη γραμμή εντολών

- Μπορούμε να δημιουργήσουμε μια νέα ομάδα με την εντολή **groupadd**. Εάν δεν χρησιμοποιήσουμε κάποιο όρισμα, αποδίδεται στην ομάδα το επόμενο διαθέσιμο GID (GroupID), από το εύρος που ορίζεται στο αρχείο **/etc/login.defs**.
- Χρησιμοποιώντας την επιλογή **-g** μπορούμε να αποδώσουμε στην ομάδα ένα συγκεκριμένο GID.

```
[user01@localhost ~]$ sudo groupadd -g 10000 group01
[sudo] password for user01:
[user01@localhost ~]$ tail /etc/group
……
user01:x:1000:
vboxsf:x:977:
vboxdrmipc:x:976:
group01:x:10000:
[user01@localhost ~]$
```

Τροποποίηση ομάδων από τη γραμμή εντολών

Μπορούμε να τροποποιήσουμε τις ιδιότητες μιας υπάρχουσας ομάδας με χρήση της εντολής **groupmod**.

- Με την επιλογή **-n** για παράδειγμα, μπορούμε να ορίσουμε νέο **όνομα** για την ομάδα:

```
[user01@localhost ~]$ sudo groupmod -n group0011 group01
[user01@localhost ~]$ tail /etc/group
……
user01:x:1000:
vboxsf:x:977:
vboxdrmipc:x:976:
group0011:x:10000:
[user01@localhost ~]$
```

  Όπως παρατηρούμε, η ομάδα group01 μετονομάστηκε σε group0011, ενώ ο αναγνωριστικός αριθμός της (GID) παρέμεινε ο ίδιος (10000).

- Με χρήση της επιλογής **-g** μπορούμε να ορίσουμε νέο **GID** για την ομάδα:

```
[user01@localhost ~]$ sudo groupmod -g 11000 group0011
[user01@localhost ~]$ tail /etc/group
…….
user01:x:1000:
vboxsf:x:977:
vboxdrmipc:x:976:
group0011:x:11000:
[user01@localhost ~]$
```

  Όπως παρατηρούμε, το GID της ομάδας group0011 άλλαξε από **10000** σε **11000**, ενώ το όνομα της ομάδας (group0011) παρέμεινε το ίδιο.

Διαγραφή ομάδων από τη γραμμή εντολών

Η διαγραφή ομάδας γίνεται με χρήση της εντολής **groupdel**. Σημειώνεται ότι δεν μπορούμε να διαγράψουμε μια ομάδα, εάν αυτή τυγχάνει να είναι **κύρια** ομάδα για οποιονδήποτε χρήστη του συστήματος.

```
[user01@localhost ~]$ sudo groupdel group0011
[user01@localhost ~]$ tail /etc/group
……
user01:x:1000:
vboxsf:x:977:
vboxdrmipc:x:976:
[user01@localhost ~]$
```

Αλλαγή κύριας ομάδας χρήστη από τη γραμμή εντολών, προσθήκη χρήστη και σε συμπληρωματικές ομάδες

- Για να αλλάξουμε την κύρια ομάδα ενός χρήστη χρησιμοποιούμε την εντολή **`usermod -g`**. Στο ακόλουθο παράδειγμα, δημιουργούμε τον χρήστη ***user02***. Μετά τη δημιουργία του, όπως μπορούμε να δούμε και με χρήση της εντολής **`id user02`**, δημιουργείται και αποδίδεται στον χρήστη αυτόματα ως κύρια ομάδα, η **ομάδα *user02 (GID 1001)***. Στη συνέχεια προσθέτουμε την **ομάδα *group02*** και την ορίζουμε ως κύρια ομάδα του χρήστη ***user02***:

```
[user01@localhost ~]$ sudo useradd user02
[user01@localhost ~]$ tail /etc/group
user01:x:1000:
vboxsf:x:977:
vboxdrmipc:x:976:
user02:x:1001:
[user01@localhost ~]$ id user02
uid=1001(user02) gid=1001(user02) groups=1001(user02)
[user01@localhost ~]$ sudo groupadd group02
[user01@localhost ~]$ sudo usermod -g group02 user02
[user01@localhost ~]$ id user02
uid=1001(user02) gid=1002(group02) groups=1002(group02)
[user01@localhost ~]$
```

- Για να προσθέσουμε τον χρήστη και σε συμπληρωματικές ομάδες πέραν της κύριας, χρησιμοποιούμε την εντολή **`usermod -aG`**. Με την παρακάτω εντολή προσθέτουμε τον χρήστη *user02* και στην ομάδα *group03*, πέραν της ομάδας *group02* στην οποία ανήκει ήδη.

```
[user01@localhost ~]$ sudo usermod -aG group03 user02
[user01@localhost ~]$ id user02
uid=1001(user02)gid=1002(group02)groups=1002(group02),1003(grou
p03)
[user01@localhost ~]$
```

Ιδιαίτερη προσοχή χρειάζεται στην χρήση της επιλογής **`-a`** ή οποία έχει την έννοια του "*append*", ή αλλιώς της προσθήκης ή προσάρτησης. Πρακτικά αυτό σημαίνει ότι, με χρήση της επιλογής **`-a,`** ο χρήστης παραμένει ως μέλος στην κύρια και σε όσες δευτερεύουσες ομάδες ενδεχομένως ανήκε, και προστίθεται επιπλέον και σε άλλες ομάδες, όπως ορίζονται από την εντολή. Εάν χρησιμοποιήσουμε την εντολή **χωρίς** την επιλογή **`-a`** (π.χ. ***`usermod -G group03 user02`***), τότε ο χρήστης θα αφαιρεθεί από τυχόν **δευτερεύουσες** ομάδες στις οποίες ανήκει και θα του αποδοθεί ως ***μοναδική* δευτερεύουσα** ομάδα, η ομάδα ***group03***.

4.5 Ρύθμιση πολιτικής διαχείρισης συνθηματικών για χρήστες

Μια καλή πολιτική κωδικού πρόσβασης για ένα σύστημα Linux θα πρέπει να περιλαμβάνει κανόνες για το μήκος, την πολυπλοκότητα και την περίοδο ισχύος / ημερομηνία λήξης του κωδικού πρόσβασης. Για παράδειγμα, συνιστάται ελάχιστο μήκος κωδικού πρόσβασης 8 χαρακτήρων, καθώς παρέχει σχετικά καλό επίπεδο

ασφάλειας, ενώ εξακολουθεί να είναι εύκολο για κάποιον να το θυμάται. Ως προς την πολυπλοκότητα, συνιστάται για έναν κωδικό πρόσβασης να απαιτείται συνδυασμός κεφαλαίων, πεζών, αριθμητικών και ειδικών χαρακτήρων. Οι πολιτικές λήξης κωδικού πρόσβασης μπορούν επίσης να ρυθμιστούν ώστε να απαιτούν από τους χρήστες να ενημερώνουν τους κωδικούς πρόσβασής τους ανά τακτά χρονικά διαστήματα, για παράδειγμα ανά 60 ή 90 ημέρες.

Όπως αναφέρθηκε νωρίτερα, στα συστήματα Linux, οι πληροφορίες λογαριασμού χρήστη, συμπεριλαμβανομένων των κωδικών πρόσβασης, αποθηκεύονται στα αρχεία ***/etc/passwd*** και ***/etc/shadow***. Το αρχείο ***/etc/passwd*** περιέχει βασικές πληροφορίες χρήστη, όπως το όνομα χρήστη, το αναγνωριστικό χρήστη (UID), το αναγνωριστικό ομάδας (GID), τον αρχικό κατάλογο (home directory) και τον φλοιό σύνδεσης (shell). Ωστόσο, οι κωδικοί πρόσβασης σε αυτό το αρχείο δεν αποθηκεύονται σε απλό κείμενο για λόγους ασφαλείας. Αντίθετα, στο πεδίο κωδικού πρόσβασης τοποθετείται ένα '**x**' για να υποδείξει ότι ο κωδικός είναι αποθηκευμένος σε κρυπτογραφημένη μορφή στο αρχείο ***/etc/shadow***.

Το αρχείο ***/etc/shadow*** περιέχει τους κρυπτογραφημένους κωδικούς πρόσβασης για κάθε λογαριασμό χρήστη, μαζί με πρόσθετες πληροφορίες όπως η ημερομηνία της τελευταίας αλλαγής κωδικού πρόσβασης, η ελάχιστη και μέγιστη ηλικία κωδικού πρόσβασης και ο αριθμός των ημερών πριν από τη λήξη ενός κωδικού πρόσβασης. Αυτό το αρχείο είναι αναγνώσιμο μόνο από τον χρήστη *root* για να αποτρέψει τη μη εξουσιοδοτημένη πρόσβαση στους κρυπτογραφημένους κωδικούς πρόσβασης.

Όπως και το αρχείο */etc/passwd*, έτσι και το αρχείο ***/etc/shadow*** περιέχει μία γραμμή για κάθε χρήστη του συστήματος, η οποία αποτελείται από εννέα πεδία, χωρισμένα μεταξύ τους με άνω και κάτω τελεία. Ακολουθεί ένα παράδειγμα μιας γραμμής του αρχείου:

```
1user02:2$6$lJ6O7…:319557:40:599999:67:72:8:9
```

1. Το όνομα του χρήστη στον οποίο ανήκει ο κωδικός πρόσβασης.
2. Ο κωδικός πρόσβασης του χρήστη σε κρυπτογραφημένη μορφή (στο παράδειγμα εμφανίζονται μόνο οι πρώτοι οκτώ χαρακτήρες).
3. Η ημέρα τελευταίας τροποποίησης του κωδικού. Πρόκειται για το πλήθος των ημερών που έχουν μεσολαβήσει μεταξύ της 01/01/1970 και της ημερομηνίας όπου τροποποιήθηκε για τελευταία φορά ο κωδικός.
4. Το ελάχιστο πλήθος ημερών που θα πρέπει να μεσολαβήσουν από την τελευταία αλλαγή κωδικού, ωσότου ο χρήστης να μπορεί να αλλάξει εκ νέου τον κωδικό του.
5. Το μέγιστο πλήθος ημερών που μπορούν να μεσολαβήσουν χωρίς αλλαγή κωδικού, μέχρι ο κωδικός αυτός να λήξει. Εάν το πεδίο είναι κενό, σημαίνει ότι η ημερομηνία τελευταίας αλλαγής δε λαμβάνεται υπόψη.
6. Περίοδος προειδοποίησης (σε ημέρες) πριν τη λήξη του κωδικού.

7. Περίοδος αδράνειας (σε ημέρες). Εάν ένας κωδικός λήξει, μπορεί ακόμα να γίνει αποδεκτός για σύνδεση στο σύστημα για αυτόν τον αριθμό ημερών. Εάν παρέλθει και αυτή η περίοδος, ο λογαριασμός κλειδώνεται.
8. Η ημέρα κατά την οποία ο κωδικός λήγει, μετρώντας ξανά από την 01/01/1970. Εάν το πεδίο είναι κενό, σημαίνει ότι ο κωδικός δε λήγει σε κάποια συγκεκριμένη ημερομηνία.
9. Το τελευταίο πεδίο της γραμμής είναι συνήθως κενό και είναι δεσμευμένο για πιθανή μελλοντική χρήση.

Μορφή ενός κρυπτογραφημένου κωδικού

Ο κρυπτογραφημένος κωδικός πρόσβασης χωρίζεται σε 3 πεδία, χωρισμένα μεταξύ τους με το σύμβολο **$**, τα οποία εμπεριέχουν τις εξής πληροφορίες:

1. τον τύπο του αλγόριθμου που χρησιμοποιήθηκε για την κρυπτογράφηση (MD5, SHA-256, SHA-512 κ.λπ). Στο παράδειγμά μας, η τιμή **6** υποδεικνύει ότι έχει χρησιμοποιηθεί ο αλγόριθμος **SHA-512**.
2. Μια τυχαία ακολουθία χαρακτήρων που ονομάζεται ***salt*** και χρησιμοποιείται για την κρυπτογράφηση.
3. Την συμβολοσειρά που προκύπτει μετά την κρυπτογράφηση του συνδυασμού του μη κρυπτογραφημένου κωδικού του χρήστη και του salt.

```
1$6 2$ lJ6O74MbfBOrAMk8 3$ sTe5nSUp9Uyo/c……
```

Η χρήση του salt εξασφαλίζει ότι ακόμη και αν δύο χρήστες έχουν τον ίδιο κωδικό πρόσβασης, το κρυπτογραφημένο αποτέλεσμα που θα παραχθεί και τελικά θα αποθηκευτεί στο αρχείο **/etc/shadow** θα είναι διαφορετικό για τον κάθε χρήστη, αφού το *salt* για κάθε χρήστη είναι διαφορετικό.

Επαλήθευση του κωδικού χρήστη

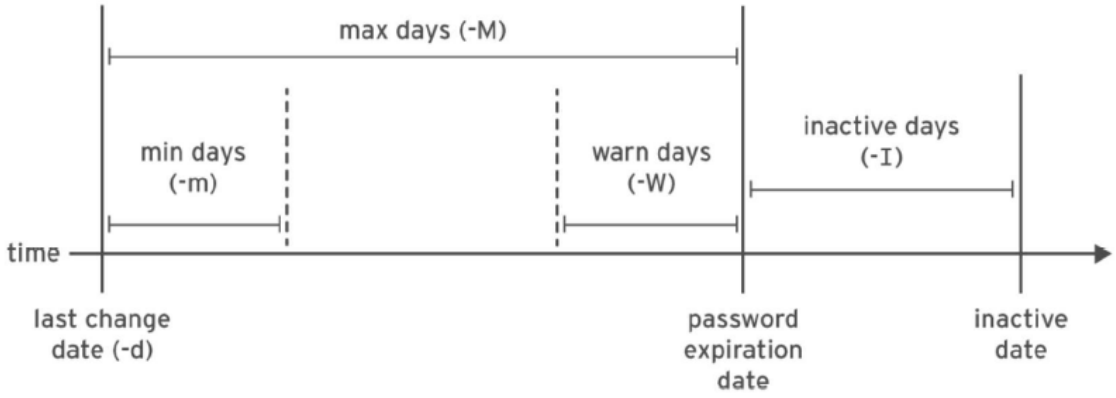
Κάθε φορά που κάποιος χρήστης προσπαθεί να συνδεθεί, το σύστημα ελέγχει την εγγραφή που υπάρχει για τον χρήστη στο αρχείο **/etc/shadow**, συνδυάζει το *salt* του χρήστη με τον κωδικό που πληκτρολογήθηκε και παράγει την κρυπτογραφημένη συμβολοσειρά με χρήση του αλγορίθμου που ορίζεται στο αρχείο **/etc/shadow**. Εάν το αποτέλεσμα συμπίπτει με την αποθηκευμένη στο αρχείο συμβολοσειρά, σημαίνει ότι ο χρήστης πληκτρολόγησε τον σωστό κωδικό και του επιτρέπεται η σύνδεση στο σύστημα.

Με τη μέθοδο αυτή, καθίσταται εφικτός ο έλεγχος χωρίς να απαιτείται κάπου στο σύστημα η αποθήκευση του μη κρυπτογραφημένου κωδικού που πληκτρολογεί ο χρήστης.

Πολιτική γήρανσης των κωδικών πρόσβασης

Όπως αναφέρθηκε και νωρίτερα, είναι σκόπιμο μια πολιτική διαχείρισης κωδικών πρόσβασης να περιλαμβάνει πρόβλεψη και σε ό,τι αφορά τη γήρανση ή την περίοδο ισχύος των κωδικών πρόσβασης των χρηστών. Πρακτικά αυτό σημαίνει ότι ο χρήστης του συστήματος θα είναι υποχρεωμένος, σε τακτά χρονικά διαστήματα τα οποία ορίζονται από τον διαχειριστή, να αλλάζει τον κωδικό πρόσβασής του στο σύστημα, αλλιώς ο λογαριασμός θα κλειδώνεται. Η λογική πίσω από αυτήν την τεχνική είναι ότι ακόμα και αν κάποιος κωδικός χρήστη παραβιαστεί ή υποκλαπεί, αυτός θα είναι διαθέσιμος στον «εισβολέα» για περιορισμένο χρονικό διάστημα.

Στο παρακάτω διάγραμμα φαίνονται οι παράμετροι οι σχετικές με τη γήρανση των κωδικών πρόσβασης. Οι παράμετροι αυτές ορίζονται από τον διαχειριστή με χρήση της εντολής **chage**.



Εικόνα 43 Παράμετροι σχετικές με τη γήρανση των κωδικών πρόσβασης

```
[user01@localhost ~]$ sudo chage -m 0 -M 90 -W 7 -I 14 user02
```

Στο παράδειγμα αυτό, ορίζονται για τον χρήστη ***user02*** τα παρακάτω:

- **-m 0** → ο ελάχιστος αριθμός ημερών που θα πρέπει να μεσολαβήσει από την τελευταία αλλαγή κωδικού του χρήστη ως την επόμενη
- **-M 90** → η μέγιστη ηλικία του κωδικού (σε ημέρες)
- **-W 7** → ο αριθμός των ημερών πριν την ημερομηνία λήξης του κωδικού κατά την οποία ο χρήστης θα ειδοποιείται να προβεί αλλαγή του κωδικού του
- **-I 14** → η περίοδος αδράνειας (σε ημέρες), ή αλλιώς, το διάστημα κατά το οποίο ο χρήστης επιτρέπεται να συνδέεται στο σύστημα, ακόμα και αν ο κωδικός του έχει λήξει. Εάν παρέλθει και η περίοδος αδράνειας και ο χρήστης δεν έχει προβεί σε αλλαγή κωδικού πρόσβασης, ο λογαριασμός του κλειδώνεται

Επιπλέον παράμετροι της εντολής ***chage***:

- **chage -d 0 user02** → ο χρήστης *user02* υποχρεώνεται, στην επόμενη σύνδεσή του, να αλλάξει κωδικό πρόσβασης
- **chage -l user02** → εμφανίζει την πολιτική γήρανσης που ισχύει για τον κωδικό του χρήστη *user02*:

```
[user01@localhost ~]$ sudo chage -l user02
Last password change                                    : Jul 20, 2023
Password expires                                        : Oct 18, 2023
Password inactive                                       : Nov 01, 2023
Account expires                                         : never
Minimum number of days between password change          : 0
Maximum number of days between password change          : 90
Number of days of warning before password expires       : 7
```

- **chage -E 2023-12-31 user02** → ορίζεται ως ημερομηνία λήξης του κωδικού πρόσβασης του χρήστη *user02* η 31 Δεκεμβρίου 2023.

Όπως αναφέρθηκε και στην παράγραφο 4.3, οι προεπιλεγμένοι κανόνες γήρανσης κωδικού πρόσβασης ορίζονται στο αρχείο **/etc/login.defs**. Οι τιμές σε αυτό το αρχείο εφαρμόζονται κατά τη δημιουργία νέων χρηστών. Οποιαδήποτε αλλαγή σε αυτό το αρχείο δεν επηρεάζει τις ρυθμίσεις που έχουν εφαρμοστεί και ισχύουν σε υπάρχοντες χρήστες.

4.6 Περιορισμοί στην πρόσβαση – Κλείδωμα και ξεκλείδωμα λογαριασμών χρηστών

Εκτός από την απενεργοποίηση ενός λογαριασμού χρήστη όταν παρέλθει η περίοδος ισχύος του, ένας διαχειριστής μπορεί να κλειδώσει τον λογαριασμό ενός χρήστη με χρήση της εντολής ***usermod -L***:

```
[user01@localhost ~]$ sudo usermod -L user02
user01@localhost ~]$ su - user02
Password:
su: Authentication failure
```

Ο λογαριασμός μπορεί να ξεκλειδωθεί από κάποιον διαχειριστή ανά πάσα στιγμή, με χρήση της εντολής **usermod -U**.

Στην περίπτωση που ο λογαριασμός εκτός από κλειδωμένος, είναι ταυτόχρονα και απενεργοποιημένος λόγω του ότι έχει λήξει ο κωδικός πρόσβασής του, θα πρέπει, εκτός από το ξεκλείδωμα, να γίνει και επέκταση της ισχύος του.

Ο φλοιός nologin

Κάποιοι λογαριασμοί δεν έχουν λόγο να συνδέονται διαδραστικά με το σύστημα. Για παράδειγμα, ο λογαριασμός που έχει δημιουργηθεί για έναν mail server και ο οποίος χρειάζεται ώστε να αποθηκεύει την αλληλογραφία των χρηστών, δε χρειάζεται να έχει δικαίωμα σύνδεσης στον φλοιό του συστήματος. Αντιθέτως είναι σκόπιμο, για λόγους ασφαλείας, σε τέτοιου είδους λογαριασμούς να απαγορεύεται η σύνδεση στο σύστημα μέσω του φλοιού.

Σε αυτές τις περιπτώσεις συνιστάται να τίθεται ως φλοιός σύνδεσης του χρήστη ο φλοιός ***/sbin/nologin***. Έτσι, στην περίπτωση που ο χρήστης προσπαθήσει να συνδεθεί στο σύστημα, ο φλοιός ***nologin*** θα απαγορέψει τη σύνδεση:

```
[user01@localhost ~]$ sudo usermod -s /sbin/nologin user02
[user01@localhost ~]$ su - user02
Password:
This account is currently not available.
```

Να σημειωθεί ότι παρότι ο φλοιός *nologin* απαγορεύει τη διαδραστική σύνδεση ενός λογαριασμού στο σύστημα, δεν αποτρέπει όλων των ειδών την πρόσβαση. Ο χρήστης δηλαδή εξακολουθεί να μπορεί να ταυτοποιηθεί στο σύστημα με τον κωδικό πρόσβασής του και να προσπελάσει αρχεία μέσω web εφαρμογών, προγράμματα ανάγνωσης αλληλογραφίας, FTP κ.λπ.

Άσκηση – Σύνδεση στο λογαριασμό διαχειριστή, διαχείριση τοπικών λογαριασμών χρηστών, διαχείριση τοπικών λογαριασμών ομάδων, διαχείριση των συνθηματικών των χρηστών

Στην άσκηση αυτή θα:

- Χρησιμοποιήσουμε την εντολή **sudo** για να μεταβούμε στον λογαριασμό του χρήστη root (βήματα 1-4)
- Χρησιμοποιήσουμε την εντολή **sudo** για να εκτελέσουμε εντολές ως **root** (βήματα 5-9)
- Δημιουργήσουμε λογαριασμούς χρηστών - Ορίσουμε κωδικούς πρόσβασης για τους χρήστες αυτούς - Διαχειριστούμε τους λογαριασμούς (τροποποίηση, διαγραφή) (βήματα 10-16)
- Δημιουργήσουμε λογαριασμούς ομάδων - Ορίσουμε κάποιες από αυτές τις ομάδες ως συμπληρωματικές για κάποιους χρήστες - Δώσουμε δικαιώματα πρόσβασης sudo σε λογαριασμούς ομάδων (βήματα 17-22)
- Ορίσουμε πολιτικές σχετικές με τους κωδικούς πρόσβασης των χρηστών (βήματα 23-30) και πιο συγκεκριμένα:
    - Κλείδωμα και ξεκλείδωμα λογαριασμού
    - Επιβολή αλλαγής κωδικού κατά την πρώτη σύνδεση στο σύστημα
    - Επιβολή αλλαγής κωδικού κάθε 90 ημέρες
    - Ορισμός ημερομηνίας λήξης λογαριασμού
    - Ορισμός μέγιστης περιόδου ισχύος του κωδικού πρόσβασης όλων των χρηστών

1. Συνδεόμαστε στο σύστημά μας με κάποιον λογαριασμό χρήστη (π.χ. *user01*) που έχουμε ήδη δημιουργήσει κατά την εγκατάσταση του λειτουργικού συστήματος. Εκτελούμε την εντολή **id** ώστε να δούμε τις πληροφορίες για τον τρέχοντα χρήστη:

```
[user01@localhost ~]$ id
uid=1000(user01)gid=1000(user01)groups=1000(user01),10(wheel)
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

2. Μεταβαίνουμε στον λογαριασμό του root με χρήση της εντολής **sudo su** και εκτελούμε ξανά την εντολή **id**:

```
[user01@localhost ~]$ sudo su
[sudo] password for user01:
[root@localhost user01]# id
uid=0(root) gid=0(root) groups=0(root)
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

3. Εκτελούμε την εντολή **pwd** (**p**rint **w**orking **d**irectory) ώστε να δούμε τον τρέχοντα κατάλογο:

```
[root@localhost user01]# pwd
/home/user01
```

Παρατηρούμε ότι παρότι έχουμε μεταβεί στον λογαριασμό του χρήστη ***root***, ο τρέχων κατάλογος είναι εκείνος του χρήστη ***user01***. Επίσης, παρατηρούμε ότι το prompt είναι [root@localhost **user01**]# (*βλ. Παρ. 4.2 Μετάβαση στον λογαριασμό Διαχειριστή*)

4. Με την εντολή **`exit`** βγαίνουμε από τον φλοιό του *root* και μεταβαίνουμε ξανά σε αυτόν, αυτήν τη φορά όμως με την εντολή **`sudo su -`**

```
[user01@localhost ~]$ sudo su -
[root@localhost ~]# pwd
/root
```

Παρατηρούμε ότι ο τρέχων κατάλογος είναι αυτή τη φορά ο κατάλογος του χρήστη *root* (/root), όπως και ότι το prompt είναι αυτή τη φορά το `[root@localhost ~]#` και όχι το `[root@localhost user01]#` όπως όταν χρησιμοποιήσαμε την εντολή **`sudo su`**.

Κάτι άλλο που παρατηρούμε όταν μεταβαίνουμε στον λογαριασμό του **root** με χρήση του **`sudo`** είναι ότι κάποιες φορές το σύστημα μας ζητά να εισάγουμε τον κωδικό του χρήστη (π.χ. του *user01*) και κάποιες όχι. Αυτό συμβαίνει διότι το `sudo` έχει μια περίοδο ισχύος από την τελευταία φορά που ο χρήστης ταυτοποιήθηκε στο σύστημα. Η περίοδος αυτή είναι συνήθως τα πέντε λεπτά. Εάν ο χρήστης χρησιμοποιήσει το `sudo` ξανά σε λιγότερο από πέντε λεπτά σε σχέση με την τελευταία φορά, το σύστημα δε θα του ζητήσει να ταυτοποιηθεί ξανά.

5. Ελέγχουμε ότι ο χρήστης ***user02*** είναι μέλος της ομάδας ***wheel*** (και επομένως μπορεί να εκτελεί οποιαδήποτε εντολή με χρήση του sudo):

```
user01@localhost ~]$ id user02
uid=1001(user02)gid=1002(group02)groups=1002(group02),10(wheel)
```

6. Συνδεόμαστε ως ***user02***:

```
[user01@localhost ~]$ su - user02
Password:
[user02@localhost ~]$
```

7. Επιχειρούμε (**χωρίς τη χρήση του `sudo`**) να δούμε τα περιεχόμενα ενός αρχείου που είναι αναγνώσιμο μόνο από τον ***root***, για παράδειγμα του αρχείου ***/var/log/messages***. Όπως είναι λογικό, η πρόσβαση απαγορεύεται, παρόλο που ο χρήστης ***user02*** ανήκει στην ομάδα ***wheel***:

```
[user02@localhost ~]$ tail /var/log/messages
tail: cannot open '/var/log/messages' for reading: Permission
 denied
```

8. Για να δούμε τα περιεχόμενα του αρχείου θα πρέπει να κάνουμε χρήση του ***sudo***:

```
[user02@localhost ~]$ sudo tail /var/log/messages
[sudo] password for user02:
Jul 24 23:04:30 localhost systemd[1]: Starting Hostname Service...
Jul 24 23:17:29 localhost systemd[1]: Starting Fingerprint
Authentication Daemon...
```

9. Επιστρέφουμε στον φλοιό του χρήστη ***user01*** με χρήση της εντολής ***exit***:

```
[user02@localhost ~]$ exit
logout
[user01@localhost ~]$
```

10. Μεταβαίνουμε στον λογαριασμό ***root*** με χρήση του ***sudo***:

```
[user01@localhost ~]$ sudo su -
[sudo] password for user01:
[root@localhost ~]#
```

11. Δημιουργούμε τον χρήστη *operator1* και επιβεβαιώνουμε τη δημιουργία του λογαριασμού ελέγχοντας το αρχείο **/etc/passwd**:

```
[root@localhost ~]# useradd operator1
[root@localhost ~]# tail /etc/passwd
vboxadd:x:978:1::/var/run/vboxadd:/bin/false
user02:x:1001:1002::/home/user02:/sbin/nologin
operator1:x:1002:1004::/home/operator1:/bin/bash
```

12. Ορίζουμε έναν κωδικό πρόσβασης για τον λογαριασμό που μόλις δημιουργήσαμε:

```
[root@localhost ~]# passwd operator1
Changing password for user operator1.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

13. Με τον ίδιο τρόπο δημιουργούμε τους χρήστες ***operator2*** και ***operator3***:

```
[root@localhost ~]# useradd operator2
[root@localhost ~]# passwd operator2
Changing password for user operator2.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@localhost ~]# useradd operator3
[root@localhost ~]# passwd operator3
Changing password for user operator3.
New password:
```

```
Retype new password:
passwd: all authentication tokens updated successfully.
```

14. Για τους δύο λογαριασμούς προσθέτουμε ως σχόλιο στα στοιχεία τους τα «Χειριστής 1» και «Χειριστής 2» αντίστοιχα και επιβεβαιώνουμε ότι το αρχείο */etc/passwd* έχει ενημερωθεί ανάλογα:

```
[root@localhost ~]# usermod -c "Χειριστής 1" operator1
[root@localhost ~]# usermod -c "Χειριστής 2" operator2
[root@localhost ~]# tail /etc/passwd
tcpdump:x:72:72::/:/sbin/nologin
vboxadd:x:978:1::/var/run/vboxadd:/bin/false
user02:x:1001:1002::/home/user02:/sbin/nologin
operator1:x:1002:1004:Χειριστής 1:/home/operator1:/bin/bash
operator2:x:1003:1005:Χειριστής 2:/home/operator2:/bin/bash
operator3:x:1004:1006::/home/operator3:/bin/bash
```

15. Διαγράφουμε από το σύστημα τον λογαριασμό του χρήστη *operator3*, μαζί με τα προσωπικά του αρχεία. Επιβεβαιώνουμε ότι ο λογαριασμός έχει διαγραφεί ελέγχοντας ότι η εγγραφή που αφορούσε στον χρήστη στο αρχείο */etc/passwd* έχει αφαιρεθεί:

```
[root@localhost ~]# userdel -r operator3
[root@localhost ~]# tail /etc/passwd
tcpdump:x:72:72::/:/sbin/nologin
user02:x:1001:1002::/home/user02:/sbin/nologin
operator1:x:1002:1004:Χρήστης 1:/home/operator1:/bin/bash
operator2:x:1003:1005:Χρήστης 2:/home/operator2:/bin/bash
[root@localhost ~]#
```

16. Με την εντολή **`exit`** βγαίνουμε από τον φλοιό του ***root*** και επιστρέφουμε στον φλοιό του χρήστη ***user01***:

```
[root@localhost ~]# exit
logout
[user01@localhost ~]$
```

17. Δημιουργούμε τον λογαριασμό ομάδας ***operators*** με ***GID 30000*** και επιβεβαιώνουμε τη δημιουργία της ομάδας ελέγχοντας το αρχείο */etc/group*:

```
root@localhost ~]# groupadd -g 30000 operators
[root@localhost ~]# tail /etc/group
group03:x:1003:user02
operator1:x:1004:
operator2:x:1005:
operators:x:30000:
```

18. Προσθέτουμε τους χρήστες ***operator1*** και ***operator2*** στην ομάδα ***operators*** με χρήση της εντολής ***usermod***. Πριν την εκτέλεση της εντολής βλέπουμε, με

χρήση της εντολής ***id***, τις λεπτομέρειες για τους δύο αυτούς λογαριασμούς και τις (κύριες) ομάδες στις οποίες ανήκουν (*operator1* και *operator2* αντίστοιχα):

```
[root@localhost ~]# id operator1 operator2
uid=1002(operator1) gid=1004(operator1) groups=1004(operator1)
uid=1003(operator2) gid=1005(operator2) groups=1005(operator2)
[root@localhost ~]# usermod -aG operators operator1
[root@localhost ~]# usermod -aG operators operator2
```

19. Επιβεβαιώνουμε την προσθήκη των δύο χρηστών στην (συμπληρωματική) ομάδα ***operators***:

```
[root@localhost ~]# id operator1 operator2
uid=1002(operator1)gid=1004(operator1)groups=1004(operator1),30000(operators)
uid=1003(operator2)gid=1005(operator2)groups=1005(operator2),30000(operators)
```

Αυτό μπορεί να διαπιστωθεί και από την αντίστοιχη εγγραφή του αρχείου ***/etc/group***:

```
[root@localhost ~]# tail /etc/group
group03:x:1003:user02
operator1:x:1004:
operator2:x:1005:
operators:x:30000:operator1,operator2
```

20. Για να προσθέσουμε την ομάδα ***operators*** (και επομένως και των χρηστών που ανήκουν σε αυτήν) στις ομάδες που έχουν πλήρη διαχειριστικά δικαιώματα (μέσω χρήσης του ***sudo***) δημιουργούμε το αρχείο ***/etc/sudoers.d/admin*** και προσθέτουμε σε αυτό την εγγραφή *%operators ALL=(ALL) ALL*

```
[root@localhost ~]# echo "%operators ALL=(ALL) ALL" >> /etc/sudoers.d/admin
[root@localhost ~]# cat /etc/sudoers.d/admin
%operators ALL=(ALL) ALL
```

21. Για να επιβεβαιώσουμε ότι τα μέλη της ομάδας ***operators*** έχουν δικαίωμα εκτέλεσης της εντολής ***sudo***, μεταβαίνουμε στον λογαριασμό ενός από τα μέλη της ομάδας (π.χ. ***operator1***) και εκτελούμε μια εντολή που απαιτεί διαχειριστικά δικαιώματα:

```
[root@localhost ~]# su - operator1
[operator1@localhost ~]$ sudo cat /etc/sudoers.d/admin
We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:
    #1) Respect the privacy of others.
    #2) Think before you type.
```

```
    #3) With great power comes great responsibility.
[sudo] password for operator1:
%operators ALL=(ALL) ALL
```

Να σημειώσουμε ότι ο χρήστης ***operator1*** δεν θα μπορούσε να δει τα περιεχόμενα του αρχείου χωρίς τη χρήση της εντολής ***sudo***, παρόλο που ανήκει σε ομάδα η οποία έχει διαχειριστικά δικαιώματα:

```
[operator1@localhost ~]$ cat /etc/sudoers.d/admin
cat: /etc/sudoers.d/admin: Permission denied
```

22. Με διαδοχικές εκτελέσεις της εντολής ***exit*** επιστρέφουμε στον φλοιό του ***root*** και στη συνέχεια στον φλοιό του ***user01***:

```
[operator1@localhost ~]$ exit
logout
[root@localhost ~]# exit
logout
[user01@localhost ~]$
```

23. Χρησιμοποιώντας τα δικαιώματα διαχειριστή που διαθέτει ο *user01* κλειδώνουμε τον λογαριασμό ***operator1***:

```
[user01@localhost ~]$ sudo usermod -L operator1
[sudo] password for user01:
```

Μετά την εκτέλεση της εντολής, ενδεχόμενη προσπάθεια σύνδεσης στο σύστημα ως ***operator1*** αποτυγχάνει:

```
[user01@localhost ~]$ su - operator1
Password:
su: Authentication failure
```

24. Ξεκλείδωμα του λογαριασμού operator1 *και* εναλλαγή στον λογαριασμό του:

```
[user01@localhost ~]$ sudo usermod -U operator1
[user01@localhost ~]$ su - operator1
Password:
[operator1@localhost ~]$
```

25. Επιβολή αλλαγής κωδικού κατά την πρώτη σύνδεση στο σύστημα για τον χρήστη *operator1*:

```
[user01@localhost ~]$ sudo chage -d 0 operator1
```

Κατά την πρώτη προσπάθεια σύνδεσης του χρήση operator1 στο σύστημα ο χρήστης υποχρεώνεται να αλλάξει κωδικό:

```
[user01@localhost ~]$ su - operator1
Password:
```

```
You are required to change your password immediately
(administrator enforced).
Current password:
New password:
Retype new password:
[operator1@localhost ~]$
```

26. Επιβολή αλλαγής κωδικού κάθε **90 ημέρες** για τον χρήστη ***operator1***:

```
[user01@localhost ~]$ sudo chage -M 90 operator1
```

Για να επιβεβαιώσουμε ότι η νέα πολιτική έχει εφαρμοστεί:

```
[user01@localhost ~]$ sudo chage -l operator1
Last password change                                    : Jul 24, 2023
Password expires                                        : Oct 22, 2023
Password inactive                                       : never
Account expires                                         : never
Minimum number of days between password change          : 0
Maximum number of days between password change          : 90
Number of days of warning before password expires       : 7
```

27. Επιβολή λήξης του λογαριασμού ***operator1*** σε **180 ημέρες** από την τρέχουσα ημερομηνία. Με χρήση της εντολής **date** βρίσκουμε σε ποια μελλοντική ημερομηνία αντιστοιχεί το διάστημα των 180 ημερών:

```
[user01@localhost ~]$ date -d "+180 days"
Sat Jan 20 07:34:17 PM EET 2024
```

Με την προσθήκη της επιλογής ***%F*** μας επιστρέφεται η ακριβής τιμή της ημερομηνίας:

```
[user01@localhost ~]$ date -d "+180 days" +%F
2024-01-20
```

28. Ορίζουμε την ημερομηνία λήξης του λογαριασμού:

```
[user01@localhost ~]$ sudo chage -E 2024-01-20 operator1
[sudo] password for user01:
[user01@localhost ~]$
```

29. Επιβεβαιώνουμε ότι η αλλαγή που κάναμε έχει εφαρμοστεί:

```
[user01@localhost ~]$ sudo chage -l operator1
Last password change                                    : Jul 24, 2023
Password expires                                        : Oct 22, 2023
Password inactive                                       : never
Account expires                                         : Jan 20, 2024
Minimum number of days between password change          : 0
Maximum number of days between password change          : 90
```

```
Number of days of warning before password expires       : 7
[user01@localhost ~]$
```

30. Ορισμός μέγιστης περιόδου ισχύος του κωδικού πρόσβασης όλων των χρηστών. Όπως έχει ήδη αναφερθεί, οι πολιτικές γήρανσης των κωδικών των χρηστών, ορίζονται, μαζί με άλλες ρυθμίσεις, στο αρχείο ***/etc/login.defs***. Μέρος των ρυθμίσεων του αρχείου φαίνεται παρακάτω. Βλέπουμε ότι εξ ορισμού, η μέγιστη διάρκεια ισχύος για τους κωδικούς πρόσβασης των νέων χρηστών είναι 99999 ημέρες (περίπου 273 έτη).

```
# Password aging controls:
#
#        PASS_MAX_DAYS   Maximum number of days a password may
be used.
#        PASS_MIN_DAYS   Minimum number of days allowed between
password changes.
#        PASS_MIN_LEN    Minimum acceptable password length.
#        PASS_WARN_AGE   Number of days warning given before a
password expires.
#
PASS_MAX_DAYS   99999
PASS_MIN_DAYS   0
PASS_WARN_AGE   7
```

Με χρήση ενός επεξεργαστή κειμένου (π.χ. του ***vim***) τροποποιούμε την τιμή PASS_MAX_DAYS από 99999 σε **180**. Υπενθυμίζεται ότι η νέα τιμή θα έχει εφαρμογή στους λογαριασμούς που θα δημιουργηθούν από τη στιγμή της τροποποίησης του αρχείου και μετά, και όχι στους υπάρχοντες λογαριασμούς.

```
[user01@localhost ~]$ sudo vim /etc/login.defs
# Password aging controls:
#
#        PASS_MAX_DAYS   Maximum number of days a password may
be used.
#        PASS_MIN_DAYS   Minimum number of days allowed between
password changes.
#        PASS_MIN_LEN    Minimum acceptable password length.
#        PASS_WARN_AGE   Number of days warning given before a
password expires.
#
PASS_MAX_DAYS   180
PASS_MIN_DAYS   0
PASS_WARN_AGE   7
```

5. Διαχείριση της πρόσβασης σε αρχεία

Μαθησιακοί στόχοι

Οι επιμορφούμενοι θα πρέπει να είναι σε θέση να:

- Ρυθμίζουν τις ιδιότητες πρόσβασης των αρχείων στο σύστημα αρχείων
- Ερμηνεύουν τις επιπτώσεις στα χαρακτηριστικά ασφαλείας των αρχείων
- Ρυθμίζουν τις ιδιότητες πρόσβασης σε αρχεία

5.1 Ιδιότητες πρόσβασης αρχείων και καταλόγων

Οι ιδιότητες πρόσβασης σε ένα σύστημα Linux καθορίζουν ποιος χρήστης ή ομάδα χρηστών έχει δικαιώματα πρόσβασης σε αρχεία και καταλόγους.

Τα δικαιώματα πρόσβασης σε ένα αρχείο ορίζονται ως εξής:

- Το αρχείο έχει έναν κάτοχο (owner) που είναι συνήθως ο χρήστης που το δημιούργησε.
- Το αρχείο επίσης ανήκει σε μια και μόνο ομάδα (group) η οποία είναι συνήθως η κύρια ομάδα του χρήστη που δημιούργησε το αρχείο.
- Τέλος, ορίζονται δικαιώματα πρόσβασης για όλους τους άλλους χρήστες (other) που δεν ανήκουν στην ίδια ομάδα με εκείνη στην οποία ανήκει το αρχείο.

Τα δικαιώματα επιπέδου χρήστη υπερισχύουν των δικαιωμάτων της ομάδας, τα οποία με τη σειρά τους υπερισχύουν των δικαιωμάτων που έχουν οριστεί για τους υπόλοιπους χρήστες.

Για κάθε μία από τις τρεις κατηγορίες μπορούν να οριστούν διαφορετικά δικαιώματα πρόσβασης (ανάγνωσης, εγγραφής, εκτέλεσης).

Οι τύποι πρόσβασης σε αρχεία και καταλόγους είναι οι εξής:

1. Ανάγνωση (**read** - **r**): Ο χρήστης μπορεί να διαβάσει το περιεχόμενο ενός αρχείου ή τα αρχεία που περιέχονται σε έναν κατάλογο.

2. Εγγραφή (**write** - **w**): Ο χρήστης μπορεί να τροποποιήσει τα περιεχόμενα ενός αρχείου ή να δημιουργήσει και να διαγράψει αρχεία σε έναν κατάλογο.

3. Εκτέλεση (**execute** - **x**): Ο χρήστης μπορεί να εκτελέσει ένα αρχείο ή να μπει σε έναν κατάλογο, να δει τις ιδιότητες των αρχείων που περιέχονται σε αυτόν και να προσπελάσει τα αρχεία του (εφόσον τα δικαιώματα πρόσβασης που έχουν οριστεί για τα αρχεία το επιτρέπουν).

Τα δικαιώματα πρόσβασης σε ένα σύστημα Linux λειτουργούν αρκετά διαφορετικά σε σχέση με τα δικαιώματα πρόσβασης του συστήματος NTFS που χρησιμοποιούν τα Microsoft Windows. Στο Linux, τα δικαιώματα έχουν ισχύ μόνο στο αρχείο ή στον

κατάλογο για τον οποίον έχουν οριστεί. Τα δικαιώματα για παράδειγμα ενός καταλόγου, δεν κληρονομούνται αυτόματα από τους υποκαταλόγους ή τα αρχεία τα οποία υπάρχουν ή που δημιουργούνται μέσα στον κατάλογο αυτόν.

Κατ' αντιστοιχία, μπορούμε να πούμε ότι το δικαίωμα ***read*** του Linux είναι το αντίστοιχο του "***List folder contents***" των Windows. Το δικαίωμα ***write*** σε έναν κατάλογο του Linux είναι το αντίστοιχο με το "***Modify***" των Windows και υποδηλώνει την δυνατότητα διαγραφής αρχείων και υποκαταλόγων.

Αναγνωρίζοντας τα δικαιώματα πρόσβασης και τον κάτοχο αρχείων και καταλόγων

Με την εντολή **`ls`** και την επιλογή **`-l`** μπορούμε να δούμε λεπτομέρειες σχετικές με τα δικαιώματα πρόσβασης αλλά και τον κάτοχο ενός αρχείου:

```
[user01@localhost ~]$ ls -l testfile1
-rw-r--r--. 1 user01 user01 5 Jul 27 12:24 testfile1
```

Με την προσθήκη της επιλογής **`-d`** μπορούμε να δούμε αντίστοιχες λεπτομέρειες για έναν κατάλογο:

```
user01@localhost ~]$ ls -ld Desktop/
drwxr-xr-x. 2 user01 user01 33 Jul 18 23:47 Desktop/
```

Στα παραπάνω παραδείγματα, ο **πρώτος** χαρακτήρας του αποτελέσματος της εντολής υποδηλώνει τον τύπο αρχείου ως εξής:

- **`-`** υποδηλώνει απλό αρχείο
- **`d`** υποδηλώνει κατάλογο
- **`l`** υποδηλώνει soft link
- Άλλες τιμές μπορεί να είναι οι **`b`** και **`c`** (υποδηλώνουν συσκευές) ή οι **`p`** και **`s`** για αρχεία ειδικού τύπου.

Οι επόμενοι εννέα χαρακτήρες δείχνουν τα δικαιώματα του αρχείου. Χωρίζονται σε τρεις ομάδες των τριών χαρακτήρων. Η πρώτη τριάδα υποδεικνύει τα δικαιώματα του χρήστη στον οποίον ανήκει το αρχείο, η δεύτερη τριάδα τα δικαιώματα της ομάδας στην οποία ανήκει το αρχείο και η τρίτη τριάδα τα δικαιώματα όλων των υπολοίπων χρηστών.

Εάν μια τριάδα έχει τιμή **`rwx`**, τότε η αντίστοιχη κατηγορία (χρήστης, ομάδα ή υπόλοιποι χρήστες) έχει στο αρχείο αυτό και τα τρία δικαιώματα, δηλαδή ανάγνωσης (**r**ead), εγγραφής (**w**rite) και εκτέλεσης (e**x**ecute). Εάν σε κάποια θέση αντί για κάποιο γράμμα υπάρχει το σύμβολο '**`-`**', τότε η αντίστοιχη κατηγορία δεν έχει το συγκεκριμένο δικαίωμα.

Ερμηνεύοντας το αποτέλεσμα της εντολής ***`ls -l testfile1`*** στο παράδειγμα του αρχείου ***testfile1*** που είδαμε νωρίτερα:

```
-rw-r--r--. 1 user01 user01 5 Jul 27 12:24 testfile1
```

- Ο πρώτος χαρακτήρας (-) υποδηλώνει απλό αρχείο
- Η επόμενη τριάδα χαρακτήρων (`rw-`) αφορά στα δικαιώματα που έχει στο αρχείο ο ***χρήστης*** *user01* και υποδηλώνει ότι αυτός έχει στο αρχείο δικαιώματα ανάγνωσης και εγγραφής, αλλά όχι εκτέλεσης
- Η επόμενη τριάδα (`r--`) αφορά στα δικαιώματα της ***ομάδας*** *user01* και δείχνουν ότι τα μέλη της ομάδας έχουν στο αρχείο μόνο δικαίωμα για ανάγνωση
- Η τελευταία τριάδα χαρακτήρων αφορά σε όλους τους ***υπόλοιπους χρήστες***, εκτός δηλαδή από τον *user01* και εκτός από τα μέλη της ομάδας *user01*, και δείχνουν ότι οι χρήστες αυτοί έχουν στο αρχείο μόνο δικαίωμα για ανάγνωση

Να σημειωθεί ότι, σε περιπτώσεις όπως η παραπάνω, όπου ο **χρήστης** και η **ομάδα** στην οποία αυτός ανήκει έχουν **διαφορετικά** δικαιώματα επάνω στο αρχείο, τότε υπερισχύουν τα δικαιώματα του χρήστη.

5.2 Ερμηνεία του αποτελέσματος πρόσβασης των ιδιοτήτων πρόσβασης σε χρήστες και ομάδες

Ας δούμε κάποια παραδείγματα εφαρμογής δικαιωμάτων πρόσβασης σε αρχεία. Για το παράδειγμα θα υποθέσουμε ότι έχουμε τρεις χρήστες (*user01, user02, operator1*) οι οποίοι ανήκουν σε ομάδες ως εξής:

Χρήστης	**Ομάδα στην οποία ανήκει**
`user01`	`user01`
`user02`	`user02, group03`
`operator1`	`operator1, operators`

Έστω ότι στον κατάλογο ***/testdir*** υπάρχουν τρία αρχεία με τα παρακάτω δικαιώματα πρόσβασης:

```
[user01@localhost testdir]$ ls -al
total 0
drwxr-xr-x.  2 user01    operators  56 Aug  2 18:27 .
dr-xr-xr-x. 19 root      root      250 Aug  2 13:08 ..
-rwxrw-r--.  1 operator1 operator1   0 Aug  2 13:09 file1.sh
-rwxr--r--.  1 user02    operators   0 Aug  2 13:10 file2.txt
-rwxrw----.  1 user01    group03     0 Aug  2 13:11 file3.txt
```

Χρησιμοποιώντας εκτός από την επιλογή **`-l`** και την επιλογή **`-a`** στην εντολή **`ls`**, εμφανίζονται και κρυφά αρχεία, όπως τα **`'.'`** και **`'..'`** που αντιπροσωπεύουν τον τρέχοντα κατάλογο (στο παράδειγμά μας τον ***/testdir***) και τον γονικό του φάκελο αντίστοιχα.

Με δεδομένα τα παραπάνω δικαιώματα των αρχείων, ας δούμε κάποια από τα αποτελέσματα που αυτά έχουν σε ό,τι αφορά την πρόσβαση των χρηστών *user01*, *user02* και *operator1* στα αρχεία αυτά:

Αποτέλεσμα:	**Ισχύει, διότι:**
Ο χρήστης ***operator1*** μπορεί να εκτελέσει το αρχείο ***file1.sh***	Ως κάτοχος του αρχείου, τα δικαιώματά του ορίζονται από την πρώτη τριάδα δικαιωμάτων, δηλαδή '`rw`**`x`**`'`, συνεπώς έχει και το δικαίωμα εκτέλεσης
Ο χρήστης ***user02*** μπορεί να δει και να τροποποιήσει τα περιεχόμενα του αρχείου ***file2.txt***	Τα δικαιώματά του στο αρχείο είναι '**`rw`**`x`' συνεπώς έχει πρόσβαση για ανάγνωση και τροποποίηση
Ο χρήστης ***operator*** μπορεί να δει, αλλά όχι να τροποποιήσει τα περιεχόμενα του αρχείου ***file2.txt***	Ως μέλος της ομάδας ***operators***, έχει δικαίωμα μόνο για ανάγνωση
Ο χρήστης ***user02*** έχει δικαίωμα τροποποίησης του αρχείου ***file3.txt***	Ως μέλος της ομάδας ***group03*** έχει δικαίωμα για ανάγνωση και για τροποποίηση των περιεχομένων του αρχείου
Ο χρήστης ***operator1*** δεν έχει δικαίωμα ανάγνωσης του αρχείου ***file3.txt***	Διότι ούτε είναι κάτοχος του αρχείου, ούτε ανήκει στην ομάδα *group03*, οι δε υπόλοιποι χρήστες δεν έχουν κανένα δικαίωμα πρόσβασης στο αρχείο
Ο χρήστης ***user01*** μπορεί να διαγράψει τα αρχεία ***file1.sh*** και ***file2.txt***	Παρότι δεν έχει επάνω στα αρχεία αυτά κάποιο δικαίωμα πρόσβασης, έχει δικαίωμα εγγραφής στον συγκεκριμένο φάκελο (όπως φαίνεται από τα δικαιώματα του '**`.`**') και επομένως μπορεί να διαγράψει οποιοδήποτε αρχείο βρίσκεται στον φάκελο αυτόν.

5.3 Αλλαγή των ιδιοτήτων πρόσβασης και ιδιοκτησίας των αρχείων από τη γραμμή εντολών

Αλλαγή των ιδιοτήτων πρόσβασης

Η αλλαγή των ιδιοτήτων πρόσβασης από τη γραμμή εντολών γίνεται με την εντολή **`chmod`** που προέρχεται από τις λέξεις **ch**ange **mod**e, μιας και τα δικαιώματα πρόσβασης (permissions) ενός αρχείου, ονομάζονται αλλιώς και *κατάσταση* (*mode*) του αρχείου).

Η εντολή δέχεται ως ορίσματα τα δικαιώματα που θέλουμε να δώσουμε σε χρήστες/ομάδες/άλλους και τη λίστα των αρχείων ή των καταλόγων επάνω στα οποία θα εφαρμοστούν τα δικαιώματα αυτά.

Υπάρχουν δύο τρόποι χρήσης της εντολής, είτε με χρήση συμβόλων, είτε με χρήση αριθμών.

Αλλαγή των ιδιοτήτων πρόσβασης με χρήση συμβόλων

Η σύνταξη της εντολής έχει τη γενική μορφή:

chmod WhoWhatWhich file|directory

όπου:

- Who → αφορά τον χρήστη ή την ομάδα στον οποίο θα αποδοθούν τα δικαιώματα και μπορεί να πάρει τις τιμές **u, g, o, a** (και συνδυασμό αυτών), όπου u σημαίνει *user*, g σημαίνει *group*, o σημαίνει *others* και a σημαίνει *all*
- What → με χρήση των συμβόλων **+, -, =** ορίζουμε αν θα προσθέσουμε (**+**) ή αν θα αφαιρέσουμε (**-**) δικαιώματα από τα ήδη ορισμένα, ή αν θα ορίσουμε ρητά νέα δικαιώματα (**=**)
- Which → ορίζει το είδος των δικαιωμάτων και μπορεί να πάρει τις τιμές **r, w, x** (και συνδυασμό αυτών) για τα δικαιώματα ανάγνωσης, εγγραφής και εκτέλεσης αντίστοιχα.

για παράδειγμα, έστω ότι έχουμε το αρχείο ***file1*** με τα εξής δικαιώματα:

```
-rw-rwxrwx. 1 user01 operators 0 Aug  4 12:36 file1
```

Η εκτέλεση της εντολής

chmod go-rw file1

θα αφαιρέσει (**-**) από την ομάδα στην οποία ανήκει το αρχείο file1, καθώς και από όλους τους υπόλοιπους χρήστες (***go***) το δικαίωμα για ανάγνωση και εγγραφή (***-rw***). Τα νέα δικαιώματα θα είναι τα εξής:

```
-rw---x--x. 1 user01 operators 0 Aug  4 12:36 file1
```

Παρατηρούμε ότι το δικαίωμα εκτέλεσης που υπήρχε τόσο για την ομάδα, όσο και για τους υπόλοιπους χρήστες δεν επηρεάστηκε. Αν χρησιμοποιούσαμε το σύμβολο '**=**' αντί για το '-', δηλαδή:

chmod go=rw file1

τα νέα δικαιώματα του αρχείου θα ήταν τα εξής:

```
-rw-rw-rw-. 1 user01 operators 0 Aug  4 12:36 file1
```

Αφού με τη χρήση του συμβόλου '**=**' θα είχαμε ορίσει ρητά τα νέα δικαιώματα, αντικαθιστώντας τα ήδη υπάρχοντα.

Η δυνατότητα να προσθέτουμε και να αφαιρούμε δικαιώματα σε αρχεία και καταλόγους σε σχέση με τα ήδη υπάρχοντα αποτελεί πλεονέκτημα του συγκεκριμένου τρόπου, αφού μας επιτρέπει απλά να τροποποιούμε τα δικαιώματα και όχι να τα ορίζουμε ξανά στο σύνολό τους για όλες τις κατηγορίες χρηστών, ακόμα και αν δεν επιθυμούμε να τα αλλάξουμε.

Μια πολύ χρήσιμη παράμετρος που μπορεί να πάρει η εντολή **chmod** είναι το **-R**. Η παράμετρος αυτή μπορεί να χρησιμοποιηθεί ώστε να ορίσουμε δικαιώματα πρόσβασης αναδρομικά στα αρχεία και τους υποφακέλους ενός καταλόγου.

Ας υποθέσουμε ότι έχουμε τον κατάλογο ***/testdir*** ο οποίος περιέχει κάποια αρχεία και τον υποκατάλογο ***subdir*** ο οποίος με τη σειρά του περιέχει το αρχείο file1_1.txt:

```
[user01@localhost testdir]$ ls -lR .
.:
total 0
-rw-rw-rw-. 1 user01    operators  0 Aug  4 12:36 file1
-rwxrw-r--. 1 operator1 operator1  0 Aug  2 13:09 file1.sh
-rwxr--r--. 1 user02    operators  0 Aug  2 13:10 file2.txt
-rwxrw----. 1 user01    group03    0 Aug  2 13:11 file3.txt
drwxr--r--. 2 user01    user01    25 Aug  4 17:12 subdir

./subdir:
total 0
-rw-r--r--. 1 user01 user01 0 Aug  4 17:12 file1_1.txt
```

Έστω ότι χρειάζεται να δώσουμε δικαιώματα για παράδειγμα **εγγραφής** σε όλα τα αρχεία του καταλόγου /*testdir*, αλλά και στα αρχεία του υποκαταλόγου /subdir, για την κατηγορία χρηστών '***others'***. Αυτό θα μπορούσε να γίνει με μία εντολή, με χρήση της επιλογής **-R** ως εξής:

```
[user01@localhost testdir]$ sudo chmod -R o+w /testdir/
```

Μετά την εκτέλεση της εντολής, τα νέα δικαιώματα έχουν ως εξής:

```
[user01@localhost testdir]$ ls -lR .
.:
total 0
-rw-rw-rw-. 1 user01    operators  0 Aug  4 12:36 file1
-rwxrw-rw-. 1 operator1 operator1  0 Aug  2 13:09 file1.sh
-rwxr--rw-. 1 user02    operators  0 Aug  2 13:10 file2.txt
-rwxrw--w-. 1 user01    group03    0 Aug  2 13:11 file3.txt
drwxr--rw-. 2 user01    user01    25 Aug  4 17:12 subdir
```

```
./subdir:
total 0
-rw-r--rw-. 1 user01 user01 0 Aug  4 17:12 file1_1.txt
```

Αλλαγή των ιδιοτήτων πρόσβασης με χρήση αριθμών

Όπως αναφέρθηκε και νωρίτερα, υπάρχει και δεύτερος τρόπος ορισμού των δικαιωμάτων πρόσβασης σε αρχεία και καταλόγους, πάλι με χρήση της εντολής **chmod**, αλλά με χρήση αριθμών αντί συμβόλων.

Η γενική μορφή σύνταξης της εντολής είναι

```
chmod ### file|directory
```

- Ο χαρακτήρας # αντιπροσωπεύει ένα ψηφίο.
- Κάθε ψηφίο αντιπροσωπεύει τα δικαιώματα μιας κατηγορίας χρηστών. Το πρώτο τα δικαιώματα του χρήστη, το δεύτερο τα δικαιώματα της ομάδας, και το τρίτο τα δικαιώματα των υπολοίπων χρηστών.
- Το κάθε ψηφίο προκύπτει από το άθροισμα των δικαιωμάτων που θέλουμε να ορίσουμε, ως εξής:
    - Η τιμή **4** αντιστοιχεί στο δικαίωμα ανάγνωσης
    - Η τιμή **2** αντιστοιχεί στο δικαίωμα εγγραφής
    - Η τιμή **1** αντιστοιχεί στο δικαίωμα εκτέλεσης

Στη μέθοδο αυτή επομένως, τα δικαιώματα απεικονίζονται με 3-ψήφιους (ή 4-ψήφιους στην περίπτωση που θέλουμε να ορίσουμε ειδικά δικαιώματα πρόσβασης, βλ. παρ. 5.5) οκταδικούς αριθμούς. Κάθε ψηφίο λοιπόν μπορεί να πάρει τιμές από 0 έως 7.

Ας υποθέσουμε ότι το αρχείο *file1_1.txt* έχει τα εξής δικαιώματα:

```
-rwxr-x---. 1 user01 user01 0 Aug  4 17:12 file1_1.txt
```

- Όσον αφορά στα δικαιώματα του χρήστη, τα σύμβολα **rwx** «μεταφράζονται» σε 4+2+1 = **7**.
- Όσον αφορά τα δικαιώματα της ομάδας, τα σύμβολα **r-x** μεταφράζονται σε 4+0+1 = **5**.
- Όσον αφορά τα δικαιώματα των υπολοίπων χρηστών, τα σύμβολα **---** μεταφράζονται σε 0+0+0 = **0**.

Η αριθμητική λοιπόν αναπαράσταση των δικαιωμάτων για το αρχείο αυτό είναι **750**.

Με αντίστροφη λογική, μπορούμε να μετατρέψουμε την αριθμητική τιμή που ορίζει τα δικαιώματα ενός αρχείου σε σύμβολα.

Έστω για παράδειγμα ότι ένα αρχείο έχει δικαιώματα που αντιστοιχούν στην τιμή **640**. Αυτό σημαίνει ότι:

- ο χρήστης/κάτοχος του αρχείου έχει σε αυτό δικαίωμα για ανάγνωση και εγγραφή (**6** = 4 + 2), ή αλλιώς **`rw-`**.
- Η ομάδα, έχει δικαίωμα μόνο για ανάγνωση (**4**), ή αλλιώς **`r--`**.
- Οι υπόλοιποι χρήστες τέλος, δεν έχουν στο αρχείο κανένα δικαίωμα και για αυτόν τον λόγο η τιμή για την κατηγορία αυτή είναι **0,** ή αλλιώς **`---`**.

Συνεπώς, η συμβολική αναπαράσταση της τιμής **640** είναι **`-rw-r-----`**.

Παράδειγμα:

Για το αρχείο *file1_1.txt* που όπως είδαμε νωρίτερα έχει δικαιώματα **`-rwxr-x---` (ή 750)**, θέλουμε να δώσουμε δικαίωμα ανάγνωσης και εγγραφής στον χρήστη και δικαίωμα μόνο για ανάγνωση στην ομάδα και στους υπόλοιπους χρήστες.

Η εντολή που θα έπρεπε να δώσουμε θα ήταν:

```
[user01@localhost subdir]$ chmod 644 file1_1.txt
```

Μετά την εκτέλεσή της, τα δικαιώματα του αρχείου έχουν πλέον ως εξής:

```
[user01@localhost subdir]$ ls -l
total 0
-rw-r--r--. 1 user01 user01 0 Aug  4 17:12 file1_1.txt
```

Αλλαγή ιδιοκτησίας αρχείων και καταλόγων

Εξ ορισμού, κάθε αρχείο που δημιουργείται ανήκει στον χρήστη ο οποίος το δημιούργησε. Επίσης, ανήκει στην κύρια ομάδα του χρήστη που το δημιούργησε. Η ομάδα αυτή συνήθως δεν έχει άλλα μέλη πέρα από τον συγκεκριμένο χρήστη. Πολλές φορές λοιπόν θα χρειαστεί να αλλάξουμε την ομάδα στην οποία ανήκει ένα αρχείο, ώστε να δώσουμε πρόσβαση σε αυτό και σε χρήστες που ανήκουν σε άλλες ομάδες.

Η αλλαγή της ομάδας στην οποία ανήκει το αρχείο μπορεί να γίνει από τον χρήστη *root*, ή και από τον ιδιοκτήτη του αρχείου, με την προϋπόθεση όμως ότι αυτός είναι μέλος της ομάδας στην οποία θέλει να μεταβιβάσει την ιδιοκτησία.

Η αλλαγή του ιδιοκτήτη ενός αρχείου μπορεί να γίνει **μόνο** από τον ***root*** και γίνεται με χρήση της εντολής **`chown`** (**ch**ange **own**er):

Για παράδειγμα:

```
-rw-rw-rw-. 1 user01    operators  0 Aug  4 12:36 file1
[root@localhost testdir]# chown user02 file1
[root@localhost testdir]# ls -al
total 0
drwxr-xrwx.  3 user01    operators  83 Aug  4 17:11 .
dr-xr-xr-x. 19 root      root      250 Aug  2 13:08 ..
-rw-rw-rw-.  1 user02    operators   0 Aug  4 12:36 file1
```

Η εντολή μπορεί να χρησιμοποιηθεί με την επιλογή -R για να αλλάξει την ιδιοκτησία αναδρομικά σε όλα τα αρχεία και τους υποκαταλόγους ενός καταλόγου.

Έστω ότι έχουμε τον κατάλογο `/subdir` με τα παρακάτω περιεχόμενα:

```
[root@localhost subdir]# ls -al
total 0
drwxr--rw-. 2 user01 user01    44 Aug  7 12:59 .
drwxr-xrwx. 3 user01 operators 83 Aug  4 17:11 ..
-rw-r--r--. 1 user01 user01     0 Aug  4 17:12 file1_1.txt
-rw-r--r--. 1 user01 user01     0 Aug  7 12:59 file1_2.txt
```

Η εκτέλεση της εντολής **`chown -R user02 subdir`** έχει το ακόλουθο αποτέλεσμα:

```
root@localhost subdir]# ls -al
total 0
drwxr--rw-. 2 user02 user01    44 Aug  7 12:59 .
drwxr-xrwx. 3 user01 operators 83 Aug  4 17:11 ..
-rw-r--r--. 1 user02 user01     0 Aug  4 17:12 file1_1.txt
-rw-r--r--. 1 user02 user01     0 Aug  7 12:59 file1_2.txt
```

Παρατηρούμε ότι, εκτός από τον ιδιοκτήτη των αρχείων, έχει αλλάξει και ο ιδιοκτήτης του καταλόγου *`subdir`* (**`'.'`**).

Η εντολή μπορεί να χρησιμοποιηθεί και για να αλλάξουμε την ιδιοκτησία ενός αρχείου ή ενός καταλόγου σε επίπεδο ομάδας, με χρήση της *άνω και κάτω τελείας* πριν το όνομα της ομάδας, στη μορφή δηλαδή

`chown :new_group_name file_name`

για παράδειγμα:

```
[root@localhost subdir]# ls -l
total 0
-rw-r--r--. 1 user02 user01 0 Aug  4 17:12 file1_1.txt
-rw-r--r--. 1 user02 user01 0 Aug  7 12:59 file1_2.txt
```

```
[root@localhost subdir]# chown :group02 file1_2.txt
[root@localhost subdir]# ls -l
total 0
-rw-r--r--. 1 user02 user01  0 Aug  4 17:12 file1_1.txt
-rw-r--r--. 1 user02 group02 0 Aug  7 12:59 file1_2.txt
```

Τέλος, μπορούμε να αλλάξουμε ταυτόχρονα και τον κάτοχο, αλλά και την ομάδα στην οποία ανήκει ένα αρχείο ή κατάλογος, βάζοντας το όνομα του νέου κατόχου πριν την άνω και κάτω τελεία, στη μορφή δηλαδή:

chown new_owner:new_group_name file_name

για παράδειγμα:

```
[root@localhost testdir]# chown operator1:operators subdir/
[root@localhost testdir]# ls -al subdir/
total 0
drwxr--rw-. 2 operator1 operators 44 Aug  7 12:59 .
drwxr-xrwx. 3 user01    operators 83 Aug  4 17:11 ..
-rw-r--r--. 1 user02    user01     0 Aug  4 17:12 file1_1.txt
-rw-r--r--. 1 user02    group02    0 Aug  7 12:59 file1_2.txt
```

Εναλλακτικά της εντολής ***chown*** και στην περίπτωση που χρειάζεται να ορίσουμε μόνο νέα ομάδα στην οποία ανήκει ένα αρχείο, μπορούμε να χρησιμοποιήσουμε και την εντολή **chgrp** (**ch**ange **gr**ou**p**). Η σύνταξή της είναι παρόμοια με εκείνη της εντολής `chown`, με τη διαφορά ότι *δεν απαιτείται η άνω και κάτω τελεία* πριν από το όνομα της ομάδας:

chgrp new_group_name file_name

5.4 Έλεγχος των ιδιοτήτων πρόσβασης νέων αρχείων που δημιουργούνται από χρήστες

Στο Linux, κάθε νέο αρχείο ή κατάλογος που δημιουργείται αποκτά αυτόματα κάποια δικαιώματα πρόσβασης. Το ποια θα είναι τα δικαιώματα αυτά καθορίζεται από την «μάσκα» που έχει οριστεί για τον συγκεκριμένο φλοιό. Για τον φλοιό `bash`, οι τιμές αυτές ορίζονται στο αρχείο **/etc/profile** ή στο αρχείο **/etc/bashrc.** Οι χρήστες μπορούν να παρακάμψουν τις τιμές αυτές για τα δικά τους αρχεία μέσω των αρχείων `.bash_profile` και `.bashrc` που βρίσκονται στο αρχικό τους κατάλογο.

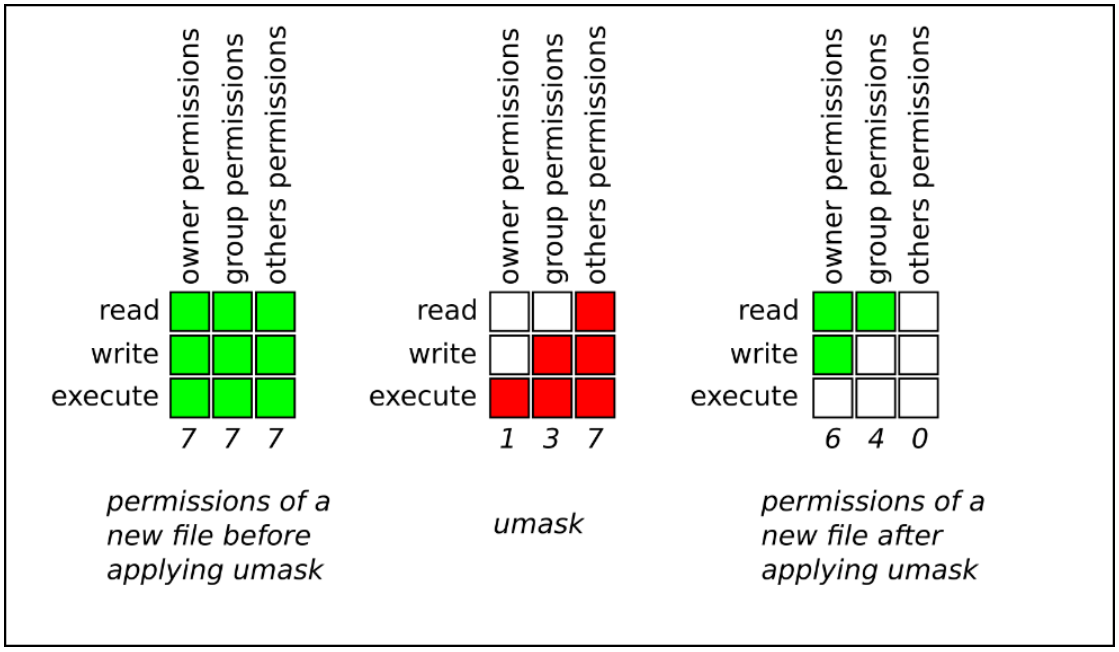
Η μάσκα είναι μια αριθμητική τιμή, στο οκταδικό σύστημα, η οποία όταν αφαιρεθεί από το προκαθορισμένο σύνολο δικαιωμάτων, καθορίζει τα τελικά δικαιώματα του

αρχείου ή του καταλόγου. Τα προκαθορισμένα δικαιώματα πρόσβασης για τις περισσότερες διανομές Linux είναι **666** για τα νέα αρχεία και **777** για τους νέους καταλόγους.

Υπενθυμίζεται ότι η τιμή **4** αντιστοιχεί στο δικαίωμα ανάγνωσης, η τιμή **2** αντιστοιχεί στο δικαίωμα εγγραφής και η τιμή **1** αντιστοιχεί στο δικαίωμα εκτέλεσης.

Η λογική με την οποία λειτουργεί η έννοια της μάσκας είναι ότι σε κάθε αρχείο που δημιουργείται αποδίδονται τα δικαιώματα που προκύπτουν εάν από τα προκαθορισμένα δικαιώματα του συστήματος (π.χ. 666) **αφαιρέσουμε την τιμή της μάσκας** (π.χ. 022). Έτσι, σε ένα τέτοιο περιβάλλον, ένα νέο αρχείο θα είχε δικαιώματα πρόσβασης `666 - 022 = 644`.

Η ακόλουθη εικόνα μας βοηθά να καταλάβουμε καλύτερα την επίδραση που θα έχει στα νέα αρχεία που θα δημιουργηθούν η αλλαγή μάσκας σε **`0137`**[1]:



Εικόνα 44 Επίδραση στα δικαιώματα μετά την αλλαγή μάσκας

Η προκαθορισμένη τιμή της μάσκας αντίστοιχα είναι συνήθως:[2]

- **002** για χρήστες **χωρίς** δικαιώματα **root**. Αυτό σημαίνει ότι για αυτούς τους χρήστες, τα δικαιώματα για τα νέα **αρχεία** που δημιουργούν είναι **`664`** (ή **`rw-rw-r--`**) και τα δικαιώματα για νέους **φακέλους** είναι **`775`** (ή **`rwxrwxr-x`**)

[1] https://access.redhat.com/webassets/avalon/d/Red_Hat_Enterprise_Linux-7-System_Administrators_Guide-en-US/images/1c3738d5acf20b7a7fff36ee1acada1b/Users_Groups-Umask_Example.png
[2] https://phoenixnap.com/kb/what-is-umask

- **022** για χρήστες με δικαιώματα **root**. Αυτό σημαίνει ότι για αυτούς τους χρήστες, τα δικαιώματα για τα νέα **αρχεία** που δημιουργούν είναι **644** (ή **rw-r--r--)** και τα δικαιώματα για νέους **φακέλους** είναι **755** (ή **rwxr-xr-x)**

Η τιμή αυτή αφορά εξίσου αρχεία και καταλόγους.

Στο ακόλουθο παράδειγμα ο χρήστης ***user01*** ανήκει στην ομάδα ***wheel*** και επομένως η προκαθορισμένη για το περιβάλλον του τιμή μάσκας είναι η **0022.** Μπορούμε να δούμε την τρέχουσα τιμή της μάσκας με την εντολή `umask` (το αρχικό '0' μπορεί να αγνοηθεί).

```
[user01@localhost ~]$ umask
0022
```

Για κάθε νέο **αρχείο** που δημιουργείται, τα δικαιώματα πρόσβασης θα είναι **666** – 022 = **644** ή, με χρήση συμβόλων, **-rw-r—r--**

```
[user01@localhost ~]$ touch testfile1
[user01@localhost ~]$ ls -l
total 0
drwxr-xr-x. 2 user01 user01 33 Jul 18 23:47 Desktop
drwxr-xr-x. 2 user01 user01  6 Jul 18 23:22 Pictures
drwxr-xr-x. 2 user01 user01  6 Jul 18 23:22 Public
-rw-r--r--. 1 user01 user01  0 Aug  8 12:35 testfile1
drwxr-xr-x. 2 user01 user01  6 Jul 18 23:22 Videos
```

Αντίστοιχα, για κάθε νέο **φάκελο** που δημιουργείται, τα δικαιώματα θα είναι **777** – 022 = **755**, ή με χρήση συμβόλων, **drwxr-xr-x**

```
[user01@localhost ~]$ mkdir testdir1
[user01@localhost ~]$ ls -l
total 0
drwxr-xr-x. 2 user01 user01 33 Jul 18 23:47 Desktop
drwxr-xr-x. 2 user01 user01  6 Jul 18 23:22 Pictures
drwxr-xr-x. 2 user01 user01  6 Jul 18 23:22 Public
drwxr-xr-x. 2 user01 user01  6 Aug  8 12:39 testdir1
-rw-r--r--. 1 user01 user01  0 Aug  8 12:35 testfile1
drwxr-xr-x. 2 user01 user01  6 Jul 18 23:22 Videos
```

Για να ορίσουμε **νέα τιμή μάσκας** για το περιβάλλον μας, χρησιμοποιούμε την εντολή **umask** ακολουθούμενη από τη νέα τιμή μάσκας.

Αν για παράδειγμα επιθυμούμε τα αρχεία και οι φάκελοι που δημιουργούνται εφεξής να **μην** είναι προσβάσιμα από την ομάδα '***others'*** μπορούμε να ορίσουμε τιμή μάσκας **027**:

```
[user01@localhost ~]$ umask 027
[user01@localhost ~]$ touch no_others_access.txt
[user01@localhost ~]$ ls -l
total 0
drwxr-xr-x. 2 user01 user01  6 Jul 18 23:22 Music
-rw-r-----. 1 user01 user01  0 Aug  8 13:12 no_others_access.txt
drwxr-xr-x. 2 user01 user01  6 Jul 18 23:22 Pictures
drwxr-xr-x. 2 user01 user01  6 Aug  8 12:39 testdir1
-rw-r--r--. 1 user01 user01  0 Aug  8 12:35 testfile1 (umask 022)
drwxr-xr-x. 2 user01 user01  6 Jul 18 23:22 Videos
```

```
[user01@localhost ~]$ mkdir no_others_dir
[user01@localhost ~]$ ls -l
total 0
drwxr-xr-x. 2 user01 user01  6 Jul 18 23:22 Music
-rw-rw----. 1 user01 user01  0 Aug  8 13:12 no_others_access.txt
drwxr-x---. 2 user01 user01  6 Aug  8 13:24 no_others_dir
drwxr-xr-x. 2 user01 user01  6 Jul 18 23:22 Pictures
drwxr-xr-x. 2 user01 user01  6 Aug  8 12:39 testdir1 (umask 022)
-rw-r--r--. 1 user01 user01  0 Aug  8 12:35 testfile1
drwxr-xr-x. 2 user01 user01  6 Jul 18 23:22 Videos
```

Εκτός από τον αριθμητικό τρόπο, μπορούμε να δούμε ή να ορίσουμε την τιμή της μάσκας χρησιμοποιώντας σύμβολα.

Για να δούμε την τρέχουσα τιμή της μάσκας (***027***), χρησιμοποιούμε την εντολή **umask** με την επιλογή **-S**:

```
[user01@localhost ~]$ umask -S
u=rwx,g=rx,o=
```

Για να ορίσουμε ως νέα τιμή μάσκας την ***027*** με τον συμβολικό τρόπο θα εκτελούσαμε την εντολή:

```
[user01@localhost ~]$ umask u=rwx,g=rx,o=
```

Να σημειωθεί τέλος ότι, σε σχέση με την εντολή ***chmod***, η εντολή ***umask*** επηρεάζει αρχεία που θα δημιουργηθούν μελλοντικά, ενώ η εντολή ***chmod*** εφαρμόζεται σε ήδη υπάρχοντα αρχεία.

5.5 Ειδικές ιδιότητες πρόσβασης (special permissions)

Οι ειδικές ιδιότητες πρόσβασης στο Linux αποτελούν ένα επιπλέον, τέταρτο επίπεδο ορισμού δικαιωμάτων, εκτός από τα επίπεδα χρήστη (user), ομάδας (group) και υπόλοιπων χρηστών (other). Με τη χρήση τους είναι εφικτό να δοθούν σε αρχεία ακόμη πιο εξειδικευμένα δικαιώματα πρόσβασης, πέραν των τυπικών.[3]

Τα ειδικά δικαιώματα αρχείων είναι τρία: **SUID**, **SGID** και **Sticky Bit**.

SUID (Set User ID): Όταν ένα εκτελέσιμο **αρχείο** έχει το SUID bit ενεργοποιημένο, τότε όταν ένας χρήστης εκτελέσει το αρχείο, η διεργασία θα εκτελεστεί με τα δικαιώματα του **ιδιοκτήτη** του αρχείου, και όχι με εκείνα του χρήστη που εκτέλεσε την εντολή. Αυτό είναι χρήσιμο, για παράδειγμα, σε προγράμματα που χρειάζονται ειδικά δικαιώματα για να λειτουργήσουν σωστά.

Χαρακτηριστικό τέτοιο παράδειγμα είναι η εντολή ***passwd***:

```
[user01@localhost ~]$ ls -l /usr/bin/passwd
-rwsr-xr-x. 1 root root 32656 May 15  2022 /usr/bin/passwd
```

Παρατηρούμε ότι στην ομάδα δικαιωμάτων που αφορούν τον κάτοχο του αρχείου, το '**x**' που θα περιμέναμε να δούμε για ένα εκτελέσιμο αρχείο, έχει αντικατασταθεί από το πεζό '**s**'.

SGID (Set Group ID): Εφαρμόζεται σε επίπεδο ομάδας στην οποία ανήκει ένα αρχείο ή ένας κατάλογος. Όταν ένας κατάλογος έχει το SGID bit ενεργοποιημένο, τότε τα αρχεία που δημιουργούνται μέσα σε αυτόν τον κατάλογο θα ανήκουν στην ίδια ομάδα με τον ιδιοκτήτη του καταλόγου, αντί για την προεπιλεγμένη ομάδα του χρήστη που δημιούργησε το αρχείο. Αυτό είναι χρήσιμο, για παράδειγμα, σε κοινόχρηστους καταλόγους όπου θέλουμε όλα τα αρχεία να ανήκουν σε μια συγκεκριμένη ομάδα.

Παράδειγμα τέτοιου καταλόγου είναι ο κατάλογος **/run/log/journal**:

```
[user01@localhost ~]$ ls -ld /run/log/journal/
drwxr-sr-x+ 3 root systemd-journal 60 Aug  4 17:06 /run/log/journal/
```

[3] https://www.redhat.com/sysadmin/suid-sgid-sticky-bit

Όταν ένα εκτελέσιμο αρχείο έχει το SGID bit ενεργοποιημένο, τότε η εντολή εκτελείται με τα δικαιώματα της ομάδας στην οποία ανήκει το αρχείο και όχι με εκείνα του χρήστη που εκτελεί την εντολή.

Παράδειγμα τέτοιας εντολής είναι η **/usr/bin/locate**:

```
[user01@localhost ~]$ ls -al /usr/bin/locate
-rwx--s--x. 1 root slocate 41048 May 16  2022 /usr/bin/locate
```

Sticky Bit: Εφαρμόζεται σε καταλόγους και επιτρέπει τη διαγραφή αρχείων στον κατάλογο αυτό μόνο από τον root και από τον χρήστη που δημιούργησε το αρχείο, ακόμη και αν οι υπόλοιποι χρήστες έχουν δικαίωμα εγγραφής στον κατάλογο. Κάτι τέτοιο είναι ιδιαίτερα χρήσιμο σε κοινόχρηστους καταλόγους όπου θέλουμε να αποτρέψουμε τη διαγραφή αρχείων από άλλους χρήστες.

5.6 Χρήση ειδικών και default ιδιοτήτων πρόσβασης για τη ρύθμιση του ιδιοκτήτη ομάδας αρχείων που δημιουργούνται σε συγκεκριμένο κατάλογο

Τα ειδικά δικαιώματα μπορούν να εφαρμοστούν με χρήση της εντολής **chmod** και των αντίστοιχων συμβόλων ή αριθμών.

- Με χρήση συμβόλων: **setuid = u+s, setgid = g+s, sticky bit = o + t**

  Για παράδειγμα:

```
[root@localhost subdir]# mkdir gid_dir
[root@localhost subdir]# ls -l
drwxr-xr-x. 2 root   root    6 Aug  8 18:17 gid_dir
[root@localhost subdir]# chmod g+s gid_dir/
[root@localhost subdir]# ls -l
drwxr-sr-x. 2 root   root    6 Aug  8 18:17 gid_dir
```

- Με χρήση τετραψήφιου αριθμού, όπου το ειδικό δικαίωμα αντιπροσωπεύεται από το πρώτο από τα τέσσερα ψηφία με την ακόλουθη αντιστοίχιση:
- **setuid = 4, setgid = 2, sticky bit = 1.**

  Αν στον κατάλογο του προηγούμενου παραδείγματος θέλαμε να ορίσουμε το **SGID** και ταυτόχρονα να δώσουμε δικαίωμα ανάγνωσης/εγγραφής/εκτέλεσης στον κάτοχο του καταλόγου και στην ομάδα στην οποία ανήκει ο κατάλογος και καμία πρόσβαση σε όλους τους υπόλοιπους χρήστες, θα εκτελούσαμε την εντολή:

```
[root@localhost subdir]# chmod 2770 gid_dir/
[root@localhost subdir]# ls -l
drwxrws---. 2 root   root    6 Aug  8 18:17 gid_dir
```

Άσκηση – ερμηνεία ιδιοτήτων στο σύστημα αρχείων του Linux, διαχείριση των ιδιοτήτων πρόσβασης αρχείων από τη γραμμή εντολών, διαχείριση default ιδιοτήτων πρόσβασης αρχείων

Α. Ερμηνεία ιδιοτήτων στο σύστημα αρχείων του Linux

Έστω ότι σε ένα σύστημα υπάρχουν τέσσερις χρήστες οι οποίοι ανήκουν σε ομάδες ως εξής:

Χρήστης	**Ομάδες όπου ανήκει**
consultant1	consultant1 και database1
operator1	operator1 και database1
contractor1	contractor1 και contractor3
operator2	operator2 και contractor3

Στον τρέχοντα κατάλογο (`.`), στον οποίο ο χρήστης ***operator1*** είναι κάτοχος με πλήρη δικαιώματα, υπάρχουν τέσσερα αρχεία με τις εξής ιδιότητες πρόσβασης:

```
drwxrwxr-x. 2 operator1 database1    .
-rw-rw-r--. 1 consultant1     consultant1 file1
-rw-r--rw-. 1 consultant1     database1   file2
-rw-rw-r--. 1 operator1       database1   file3
-rw-r-----. 1 operator1       database1   file4
```

1. Ποιο αρχείο ανήκει στον ***operator1*** και είναι αναγνώσιμο από όλους τους χρήστες;
    a. file1
    b. file2
    c. **file3**
    d. file4

Το *file3* είναι το μοναδικό που ανήκει στον *operator1* και ταυτόχρονα είναι αναγνώσιμο και από όλους τους χρήστες, ασχέτως ομάδας στην οποία ανήκουν ($3^{η}$ ομάδα ιδιοτήτων, `-rw-rw-r--`).

2. Ποιο αρχείο μπορεί να τροποποιήσει ο χρήστης ***contractor1***;
    a. file1
    b. **file2**
    c. file3
    d. file4

Καταρχάς ο *contractor1* δεν είναι κάτοχος κάποιου αρχείου, συνεπώς δεν εξετάζουμε την 1η τριάδα δικαιωμάτων των αρχείων. Σε ό,τι αφορά τα δικαιώματα ομάδων, τα 2 αρχεία που δίνουν δικαίωμα τροποποίησης σε ομάδες είναι τα *file1* και *file3,* τα οποία ανήκουν στις ομάδες *consultant1* και *database1* αντίστοιχα, στις οποίες όμως δεν είναι μέλος ο *contractor1*. Επομένως ο χρήστης ανήκει στην κατηγορία '*others'* και το *file2* είναι το μόνο αρχείο το οποίο δίνει δικαίωμα εγγραφής στην κατηγορία αυτή (`-rw-r--r`**`w`**`-`). Συνεπώς, είναι και το μόνο που μπορεί να τροποποιήσει ο *contractor1*.

3. Ποιο αρχείο είναι **μη** αναγνώσιμο από τον χρήστη operator2;
    a. file1
    b. file2
    c. file3
    d. **file4**

Ο *operator2* δεν είναι κάτοχος κάποιου αρχείου, ούτε είναι μέλος κάποιας από τις ομάδες για τις οποίες έχουν εφαρμοστεί δικαιώματα πρόσβασης στα αρχεία. Συνεπώς εξετάζουμε την 3η τριάδα ιδιοτήτων των αρχείων, από τα οποία μόνο το *file4* **δεν** δίνει δικαίωμα ανάγνωσης στην κατηγορία '*others'*.

4. Ποιο αρχείο ανήκει στην ομάδα *consultant1*;
    a. **file1**
    b. file2
    c. file3
    d. file4

`-rw-rw-r--. 1 consultant1 `**`consultant1`**` file1`

5. Ποια αρχεία έχει δικαίωμα να διαγράψει ο χρήστης *operator1*;
    a. **Όλα τα αρχεία**
    b. Κανένα αρχείο
    c. Τα αρχεία file1 και file2
    d. Τα αρχεία file3 και file4

Καθώς ο *operator1* είναι κάτοχος του καταλόγου (`drwxrwxr-x. 2 `**`operator1`**` database1 .`), έχει δικαίωμα διαγραφής οποιουδήποτε αρχείου βρίσκεται σε αυτόν, ακόμα και αν δεν έχει δικαίωμα διαγραφής στο ίδιο το αρχείο.

Β. Διαχείριση των ιδιοτήτων πρόσβασης αρχείων από τη γραμμή εντολών

Σκοπός της άσκησης είναι η δημιουργία ενός φακέλου στον οποίον όλοι οι χρήστες που ανήκουν στην ομάδα ***operators*** θα έχουν δικαίωμα να προσθέτουν και να διαγράφουν αρχεία. Υποθέτουμε ότι οι χρήστες ***operator1*** και ***operator2*** είναι ήδη μέλη της ομάδας.

1. Συνδεόμαστε στο σύστημα με λογαριασμό απλού χρήστη (έστω *student1*)
2. Αλλάζουμε σε λογαριασμό *root*

```
[student1@localhost ~]$ su -
Password:
[root@localhost ~]#
```

3. Με χρήση της εντολής `mkdir` δημιουργούμε τον φάκελο */home/operators*

```
root@localhost ~]# mkdir /home/operators
```

4. Με χρήστη της εντολής **`chown`** ορίζουμε ως ιδιοκτήτρια ομάδα του φακέλου την ομάδα operators

```
root@localhost ~]# chown :operators /home/operators
```

5. Επιβεβαιώνουμε ότι τα δικαιώματα του φακέλου είναι τέτοια, ώστε τα μέλη της ομάδας *operators* να μπορούν να δημιουργήσουν και να διαγράψουν αρχεία στον φάκελο. Επίσης, οι χρήστες που δεν ανήκουν στην ομάδα, δε θα πρέπει να έχουν πρόσβαση στα αρχεία
   5.1. Χρησιμοποιούμε την εντολή **`ls`** ώστε να δούμε τα δικαιώματα του φακέλου

```
[root@localhost ~]# ls -ld /home/operators
drwxr-xr-x. 2 root operators 6 Aug  9 13:14 /home/operators
```

Παρατηρούμε ότι η ομάδα *operators* **δεν** έχει δικαίωμα εγγραφής στον φάκελο

   5.2. Με χρήση της εντολής **`chmod`** δίνουμε στην ομάδα *operators* δικαίωμα εγγραφής στον φάκελο

```
[root@localhost ~]# chmod g+w /home/operators
[root@localhost ~]# ls -ld /home/operators
drwxrwxr-x. 2 root operators 6 Aug  9 13:14 /home/operators
```

   5.3. Επίσης με την εντολή **`chmod`** αφαιρούμε από όλους τους υπόλοιπους χρήστες το δικαίωμα πρόσβασης στον φάκελο

```
[root@localhost ~]# chmod 770 /home/operators
[root@localhost ~]# ls -ld /home/operators
drwxrwx---. 2 root operators 6 Aug  9 13:14 /home/operators
```

6. Αποσυνδεόμαστε από τον φλοιό του *root* και συνδεόμαστε ως χρήστης *operator1*

```
[root@localhost ~]# exit
logout
[student1@localhost ~]$ su - operator1
Password:
[operator1@localhost ~]$
```

7. Δημιουργούμε το αρχείο ***operator1.txt*** μέσα στον φάκελο ***/home/operators***

7.1. Με χρήση της εντολής **cd** μπαίνουμε στον φάκελο

```
[operator1@localhost ~]$ cd /home/operators
```

7.2. Με την εντολή **touch** δημιουργούμε ένα κενό αρχείο με όνομα ***operator1.txt***

```
[operator1@localhost operators]$ touch operator1.txt
```

8. Εμφανίζουμε τις ιδιότητες πρόσβασης και τους κατόχους του αρχείου με την εντολή **ls -l**

```
[operator1@localhost operators]$ ls -l operator1.txt
-rw-rw-r--. 1 operator1 operator1 0 Aug  9 16:01 operator1.txt
```

9. Τροποποιούμε τα δικαιώματα ομάδας του αρχείου, ώστε τα μέλη της ομάδας ***operators*** να μπορούν να το τροποποιήσουν

9.1. Με την εντολή **chown** αλλάζουμε την ομάδα στην οποία ανήκει το αρχείο

```
[operator1@localhost operators]$ chown :operators operator1.txt
```

9.2. Επιβεβαιώνουμε με χρήση της εντολής **ls -l**

```
[operator1@localhost operators]$ ls -l operator1.txt
-rw-rw-r--. 1 operator1 operators 0 Aug  9 16:01 operator1.txt
```

10. Αποσυνδεόμαστε και συνδεόμαστε ως χρήστης ***operator2***, ο οποίος επίσης ανήκει στην ομάδα ***operators***

```
[operator1@localhost ~]$ exit
logout
[student1@localhost ~]$ su - operator2
Password:
[operator2@localhost ~]$
```

11. Θα επιβεβαιώσουμε ότι ο χρήστης ***operator2*** μπορεί να τροποποιήσει το αρχείο ***operator1.txt*** που βρίσκεται στον φάκελο ***/home/operators***

11.1. Μπαίνουμε στον φάκελο και με την εντολή **echo** προσθέτουμε περιεχόμενο στο αρχείο

```
[operator2@localhost ~]$ cd /home/operators
[operator2@localhost operators]$ echo "newtext" >> operator1.txt
[operator2@localhost operators]$
```

11.2. Επιβεβαιώνουμε (π.χ. με την εντολή **cat**) ότι το περιεχόμενο έχει προστεθεί στο αρχείο

```
[operator2@localhost operators]$ cat operator1.txt
newtext
[operator2@localhost operators]$
```

11.3. Αποσυνδεόμαστε από τον φλοιό του χρήστη ***operator2***

```
[operator2@localhost operators]$ exit
logout
[student1@localhost ~]$
```

Γ. Διαχείριση default ιδιοτήτων πρόσβασης αρχείων

Σκοπός της άσκησης είναι ο ορισμός των ιδιοτήτων πρόσβασης για νέα αρχεία που δημιουργούνται σε έναν φάκελο, με χρήση της μάσκας (**umask**) και του bit **setgid**. Πιο συγκεκριμένα:

- Θα δημιουργήσουμε έναν κοινόχρηστο φάκελο του οποίου τα αρχεία θα ανήκουν αυτόματα στην ομάδα ***operators***
- Θα ερμηνεύσουμε τις επιπτώσεις από την αλλαγή της μάσκας
- Θα ορίσουμε default ιδιότητες για συγκεκριμένους χρήστες
- Θα επιβεβαιώσουμε ότι οι αλλαγές έχουν το επιθυμητό αποτέλεσμα

1. Συνδεόμαστε στο σύστημα με λογαριασμό απλού χρήστη (έστω ***student1***)
2. Συνδεόμαστε ως χρήστης ***operator1*** με χρήση της εντολής **`su`**

```
[student1@localhost ~]$ su - operator1
Password:
[operator1@localhost ~]$
```

3. Με χρήση της εντολής **`umask`** εμφανίζουμε την τρέχουσα μάσκα για τον χρήστη *operator1*

```
[operator1@localhost ~]$ umask
0002
```

4. Δημιουργούμε τον φάκελο ***/tmp/shared*** και μέσα σε αυτόν, το αρχείο ***defaults***. Ελέγχουμε τις ιδιότητες πρόσβασης του φακέλου και του αρχείου
   4.1. Με την εντολή **`mkdir`** δημιουργούμε τον φάκελο και με την εντολή **`ls -ld`** ελέγχουμε τις ιδιότητες πρόσβασής του

```
[operator1@localhost /]$ mkdir /tmp/shared
[operator1@localhost /]$ ls -ld /tmp/shared/
drwxrwxr-x. 2 operator1 operator1 6 Aug  9 17:25 /tmp/shared/
```

   4.2. Με την εντολή **`touch`** δημιουργούμε ένα αρχείο με όνομα ***defaults***

```
[operator1@localhost /]$ touch /tmp/shared/defaults
```

   4.3. Με την εντολή **`ls -l`** βλέπουμε τα δικαιώματα του νέου αρχείου

```
[operator1@localhost /]$ ls -l /tmp/shared/defaults
-rw-rw-r--. 1 operator1 operator1 0 Aug 10 16:45 /tmp/shared/defaults
```

5. Αλλάζουμε την ομάδα στην οποία ανήκει ο φάκελος ***/tmp/shared*** από την ομάδα ***operator1*** στην ομάδα ***operators*** και επιβεβαιώνουμε την αλλαγή
   5.1. Με χρήση της **`chown`** αλλάζουμε την ομάδα κάτοχο

```
[operator1@localhost /]$ chown :operators /tmp/shared/
```

5.2. Με χρήση της **ls -ld** επιβεβαιώνουμε ότι τα νέα δικαιώματα του φακέλου έχουν εφαρμοστεί

```
[operator1@localhost /]$ ls -ld /tmp/shared/
drwxrwxr-x. 2 operator1 operators 22 Aug 10 16:45 /tmp/shared/
```

5.3. Με χρήση της εντολής **touch** δημιουργούμε στον φάκελο το αρχείο ***group*** και ελέγχουμε τα δικαιώματά του

```
[operator1@localhost /]$ touch /tmp/shared/group
[operator1@localhost /]$ ls -l /tmp/shared/group
-rw-rw-r--. 1 operator1 operator1 0 Aug 10 16:57 /tmp/shared/group
```

Παρατηρούμε ότι η ομάδα κάτοχος του νέου αρχείου **δεν** είναι η ***operators*** αλλά η ***operator1***

6. Εξασφαλίζουμε ότι τα αρχεία που θα δημιουργούνται εφεξής στον φάκελο ***/tmp/shared*** θα έχουν ως κάτοχο την ομάδα ***operators***

6.1. Με την εντολή **chmod** ενεργοποιούμε το ***group ID*** bit του φακέλου και ελέγχουμε με την **ls -ld** ότι έχει ενεργοποιηθεί

```
[operator1@localhost /]$ chmod g+s /tmp/shared/
[operator1@localhost /]$ ls -ld /tmp/shared/
drwxrwsr-x. 2 operator1 operators 35 Aug 10 16:57 /tmp/shared/
```

6.2. Με την εντολή **touch** δημιουργούμε ένα νέο αρχείο με όνομα ***operations_database.txt***

```
[operator1@localhost /]$ touch /tmp/shared/operations_database.txt
```

6.3. Ελέγχουμε τα δικαιώματα πρόσβασης του νέου αρχείου με την εντολή **ls -l** και διαπιστώνουμε ότι το νέο αρχείο ανήκει πράγματι στην ομάδα ***operators.*** Αντίθετα, το αρχείο ***group*** που είχαμε δημιουργήσει πριν την ενεργοποίηση του SGID, εξακολουθεί να ανήκει στην ομάδα ***operator1***

```
[operator1@localhost /]$ ls -l /tmp/shared/operations_database.txt
-rw-rw-r--. 1 operator1 operators 0 Aug 10 17:07
/tmp/shared/operations_database.txt
[operator1@localhost /]$ ls -l /tmp/shared/group
-rw-rw-r--. 1 operator1 operator1 0 Aug 10 16:57 /tmp/shared/group
```

7. Θα δημιουργήσουμε στον φάκελο ***/tmp/shared*** το αρχείο ***operations_network.txt***. Θα σημειώσουμε τις ιδιότητες πρόσβασης του αρχείου και την ομάδα στην οποία ανήκει. Θα αλλάξουμε το **umask** για τον χρήστη ***operator1***. Στη συνέχεια, θα δημιουργήσουμε ένα νέο αρχείο με όνομα ***operations_production.txt*** και θα σημειώσουμε τις ιδιότητες πρόσβασής του.

7.1. Με την εντολή echo δημιουργούμε στον φάκελο το αρχείο ***operations_network.txt*** με περιεχόμενο το κείμενο ***«newtext».***

```
[operator1@localhost shared]$ echo newtext >> /tmp/shared/operations_network.txt
```

7.2. Σημειώνουμε τις ιδιότητες πρόσβασης του νέου αρχείου.

```
[operator1@localhost shared]$ ls -l operations_network.txt
-rw-rw-r--. 1 operator1 operators 8 Aug 21 09:15 operations_network.txt
```

7.3. Αλλάζουμε τη μάσκα για τα αρχεία του χρήστη ***operator1*** σε ***027*** και επιβεβαιώνουμε την αλλαγή.

```
[operator1@localhost shared]$ umask 027
[operator1@localhost shared]$ umask
0027
```

7.4. Δημιουργούμε ένα νέο αρχείο ***operations_production.txt***, ελέγχουμε τις ιδιότητες πρόσβασης και επιβεβαιώνουμε ότι τα νέα αρχεία που δημιουργούνται εφεξής επιτρέπουν μόνο ανάγνωση για την ομάδα ***operators*** και απαγορεύουν την πρόσβαση σε όλους τους υπόλοιπους χρήστες.

```
[operator1@localhost shared]$ touch operations_production.txt
[operator1@localhost shared]$ ls -l operations_production.txt
-rw-r-----. 1 operator1 operators 0 Aug 21 09:23 operations_production.txt
```

8. Από ένα **νέο τερματικό**, συνδεόμαστε ως χρήστης ***operator1*** και ελέγχουμε την τιμή της μάσκας. Παρατηρούμε ότι η ***default*** μάσκα για τον χρήστη παραμένει ***0002*** παρότι προηγουμένως την είχαμε αλλάξει σε ***027***.

```
[operator1@localhost ~]$ umask
0002
```

9. Για να αλλάξουμε μόνιμα την ***default*** μάσκα για τον χρήστη ***operator1,*** για παράδειγμα σε ***027***, θα πρέπει να τροποποιήσουμε ανάλογα το αρχείο ***.bashrc*** προσθέτοντας την αντίστοιχη οδηγία. Το αρχείο ***.bashrc*** βρίσκεται στον αρχικό κατάλογο του χρήστη.

```
[operator1@localhost ~]$ echo "umask 027" >> ~/.bashrc
```

10. Για να επιβεβαιώσουμε ότι η νέα ρύθμιση έχει εφαρμοστεί, **αποσυνδεόμαστε** από το τερματικό και συνδεόμαστε ξανά ως ***operator1***.

```
[operator1@localhost ~]$ exit
[operator1@localhost ~]$ umask
0027
```