

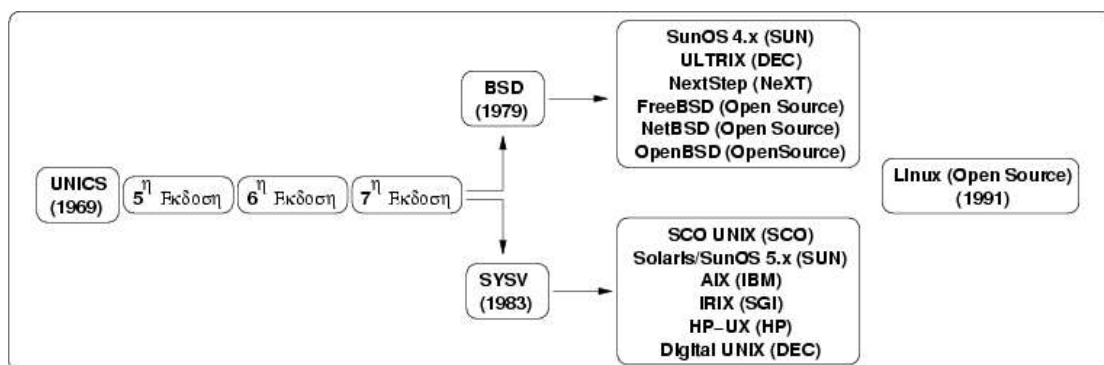
# Εισαγωγή στο Λειτουργικό Σύστημα UNIX & GNU/Linux

## ΤΙ ΕΙΝΑΙ ΕΝΑ ΛΕΙΤΟΥΡΓΙΚΟ ΣΥΣΤΗΜΑ & Η ΕΞΕΛΙΞΗ ΤΟΥ UNIX

Με τον όρο Λειτουργικό Σύστημα (Λ.Σ), εννοούμε μια ομάδα από προγράμματα υπολογιστή, που δίνουν την δυνατότητα στον άνθρωπό να εκμεταλλευτεί τους hardware πόρους που διαθέτει ο υπολογιστής. Με πιο απλά λόγια, να απεικονίσει γραφήματα στην οθόνη, να κάνει υπολογισμούς, να δημιουργήσει μια βάση δεδομένων ή ακόμα και να παίξει κάποιο παιχνίδι. Για να μπορέσουν να γίνουν όλα αυτά, πρέπει να υπάρχει ένα σεν από προγράμματα που να αναλαμβάνουν να κάνουν τουλάχιστον τα βασικές λειτουργίες. Έτσι, είναι λογικό να υπάρχει ένα πρόγραμμα που να είναι υπεύθυνο για το χειρισμό της οθόνης, την απεικόνιση με άλλα λόγια των λειτουργιών του υπολογιστή. Αντίστοιχα, όταν κάποιος παίζει ένα παιχνίδι και ακούγονται οι ήχοι του παιχνιδιού, πρέπει να υπάρχει ένα πρόγραμμα που να ξέρει πως θα κάνει αυτούς τους ήχους να ακούγονται. Έτσι, για όλες τις επιμέρους λειτουργίες που πρέπει να εκτελεστούν προκειμένου να έχουμε ένα αποτέλεσμα, πρέπει να υπάρχει ένα πρόγραμμα που να είναι αρκετά εξειδικευμένο ώστε να μπορούν να γίνουν όλα σωστά.

Μια δυνατότητα που θα μπορούσε υπάρχει, θα ήταν το κάθε πρόγραμμα (ένα παιχνίδι, ένα πρόγραμμα μουσικής κλπ), να ήταν σχεδιασμένο έτσι ώστε να έχει από μόνο του μια ομάδα από τέτοια εξειδικευμένα προγράμματα και να εκτελείται αυτόνομα χωρίς την ανάγκη ενός Λ.Σ. Αυτό όμως θα οδηγούσε μοιραία (αν λάβει κανείς υπ' όψιν του την πρόοδο που υπάρχει στις δυνατότητες που έχουν οι υπολογιστές) στο σχεδιασμό τεράστιων προγραμμάτων μόνο και μόνο για να ελέγξουν τον υπολογιστή, πόσο μάλλον για να κάνουν και την δουλειά για την οποία προορίζονται.

Έτσι λοιπόν προέκυψαν στην πάροδο του χρόνου πολλά και διαφορετικά Λ.Σ από διάφορες εταιρίες με σκοπό την καλύτερη αξιοποίηση των πόρων – των δυνατοτήτων που δίνει ο υπολογιστής. Ένα από τα πιο παλιά Λ.Σ που δημιουργήθηκαν είναι το Λ.Σ UNIX. Οι αρχές τις ιστορίας του φτάνουν στο έτος 1969. Αναπτύχθηκε κυρίως μέσα σε ακαδημαϊκούς και ερευνητικούς χώρους πανεπιστημίων και εταιριών. Η ανάγκη που οδήγησε στην δημιουργία ενός Λ.Σ από μέρους τους, ήταν ότι χρειαζόνταν ένα τρόπο να μπορούν να μπορούν πολλοί διαφορετικοί άνθρωποι να χρησιμοποιούν ένα υπολογιστή (μην ξεχνάτε ότι στην αρχή της ιστορίας των υπολογιστών, ένας και μόνο υπολογιστής κόστιζε αρκετά εκατομμύρια δολάρια και ήταν πραγματικά ογκώδης), να μπορεί κάποιος να κάνει πολλά διαφορετικά πράγματα την ίδια στιγμή, να μην σταματάει η λειτουργία του για κάποιο λόγο και να δίνει την δυνατότητα σε πολλούς διαφορετικούς ανθρώπους να ανταλλάσσουν μεταξύ τους δεδομένα. Όλα αυτά που με την συνήθη ορολογία των υπολογιστών ονομάζονται *multi-user*, *multi-tasking environment*, *stability*, *network* και *connectivity* τα έδωσε το Λ.Σ. UNIX.



Σχήμα 1: Συνολική ιστορία του UNIX

## ΔΟΜΗ ΚΑΙ ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ Λ.Σ UNIX

Ένα τυπικό Λ.Σ απαρτίζεται από τέσσερα κύρια μέρη:

- **kernel:** πρόκειται για το πιο βασικό τμήμα του Λ.Σ και είναι το software εκείνο που αναλαμβάνει τη βασική διαχείριση του υπολογιστή (κάρτες οθόνης, περιφερειακά, σκληρούς δίσκους κλπ). Όλα τα προγράμματα που χρησιμοποιούν κάποια τμήματα του υπολογιστή, συνεργάζονται με τον *kernel* προκειμένου να χρησιμοποιήσουν κάποιο κομμάτι του hardware.
- **Shells και GUIs:** είναι ένα σετ από προγράμματα που έχουν σαν σκοπό να δώσουν στον χρήστη του Η/Υ την δυνατότητα να “επικοινωνήσει” με το Λ.Σ και κατά συνέπεια με το hardware του υπολογιστή. Δεν είναι τίποτε άλλο από προγράμματα που περιμένουν από τον χρήστη εντολές/οδηγίες για το τι θέλει να κάνει ο υπολογιστής. Στο UNIX γενικά, ο όρος *shell* έχει να κάνει με προγράμματα που περιμένουν ο χρήστης να πληκτρολογήσει κάποιες εντολές για να εκτελεστούν, ενώ με τον όρο *GUI* εννοούμε προγράμματα που εκτελούν εντολές με την χρήση γραφικής απεικόνισης. Για παράδειγμα, αν θέλουμε να ξεκινήσουμε ένα πρόγραμμα αναπαραγωγής μουσικών CDs, ένα *GUI* θα εμφανίζει πάνω στην οθόνη ένα κουμπί που όταν πατηθεί θα ξεκινήσει το πρόγραμμα που “παίζει” CD. Σε ένα UNIX *shell* αντίθετα, θα πρέπει να θυμάται ο χρήστης το όνομα του προγράμματος και να το πληκτρολογήσει προκειμένου να εκτελεστεί το πρόγραμμα που τον ενδιαφέρει.
- **System Utilities / Daemons:** είναι προγράμματα που επιτελούν μερικές βασικές διεργασίες σε ένα UNIX σύστημα. Για παράδειγμα, ένα πρόγραμμα που υπολογίζει ολοκληρώματα δεν μπορεί να πει κανείς ότι είναι *system utility*, αλλά ένα άλλο που ελέγχει αν το hardware του υπολογιστή λειτουργεί σωστά, είναι.
- **Application Programs:** είναι προγράμματα που επιτελούν μερικές από τις πιο συνηθισμένες λειτουργίες ενός υπολογιστή. Για παράδειγμα στην κατηγορία αυτή θα ανήκει ένα πρόγραμμα που χρησιμοποιείται για τη συγγραφή κειμένων ή την επεξεργασία εικόνων.

## ΧΡΗΣΤΕΣ, ΠΕΡΙΟΧΕΣ ΧΡΗΣΤΩΝ & ΚΩΔΙΚΟΙ ΠΡΟΣΒΑΣΗΣ

### Χρήστες (users)

Ένα από τα βασικά χαρακτηριστικά του Λ.Σ UNIX, είναι η δυνατότητα που δίνει σε πολλούς διαφορετικού ανθρώπους να χρησιμοποιήσουν τον ίδιο υπολογιστή ανεξάρτητα ο ένας από τον άλλο. Για να είναι διακριτοί οι χρήστες, κάθε ένας από αυτούς έχει ένα μοναδικό όνομα, το *username*. Η πρώτη πληροφορία που πρέπει να δώσει κανείς ώστε να αποκτήσει πρόσβαση στον υπολογιστή είναι αυτό το χαρακτηριστικό όνομα (*username*).

### Κωδικοί πρόσβασης (passwords)

Το *username* του κάθε χρήστη είναι γενικά κάτι που χρειάζεται να το ξέρουν πολλά διαφορετικά άτομα προκειμένου να μπορούν να επικοινωνήσουν μαζί του ή να ανταλλάξουν πληροφορίες. Έτσι είναι απαραίτητο να υπάρχει κάποιο στοιχειώδες επίπεδο ασφάλειας στην χρήση του *username* ώστε να μην είναι δυνατή η χρήση του για μη ενδεδειγμένους σκοπούς. Έτσι λοιπόν όταν κάποιος ζητάει πρόσβαση στον υπολογιστή δίνοντας το *username* του, ο υπολογιστής ζητά αμέσως μετά τον κωδικό πρόσβασης (*password*). Το *password* έχει προσωπικό χαρακτήρα και πρέπει να παραμένει κρυφό σε κάθε περίπτωση. Η σημασία του μπορεί να αντιστοιχηθεί με το PIN (Personal Identification Number) που αποκτάει κανείς όταν προμηθεύεται μια κάρτα ανάληψης μετρητών από μια τράπεζα.

### Ομάδες χρηστών (user groups)

Εκτός από το *username* που έχει ο κάθε χρήστης ένα άλλο χαρακτηριστικό του είναι και η ομάδα (*group*) στο οποίο ανήκει. Ο λόγος που υπάρχει η έννοια του *group* είναι ότι υπάρχουν περιπτώσεις που χρειάζεται για μια συγκεκριμένη δουλειά, να χειριστεί το λειτουργικό σύστημα ένα σύνολο από χρήστες με κάποια κοινά χαρακτηριστικά. Είναι απλούστερη η διαδικασία όταν αυτοί οι χρήστες είναι ομαδοποιημένοι σε κάποιο *group*.

### Περιοχές χρηστών (home directories)

Κάθε υπολογιστής έχει κάποιες μονάδες αποθήκευσης στις οποίες κρατάει δεδομένα που αφορούν το Λ.Σ., αλλά και τα δεδομένα των χρηστών. Κάθε χρήστης ενός UNIX συστήματος έχει μια προσωπική περιοχή στις μονάδες αποθήκευσης του υπολογιστή μέσα στην οποία μπορεί να κρατά τις πληροφορίες που τον αφορούν. Στις περισσότερες περιπτώσεις ο χώρος που είναι διαθέσιμος στον κάθε χρήστη είναι περιορισμένος και το μέγεθος της περιοχής που διαθέτει εξαρτάται από την συνολική χωρητικότητα των αποθηκευτικών μονάδων.

## ΕΙΣΟΔΟΣ / ΕΞΟΔΟΣ ΣΕ ΕΝΑ UNIX ΣΥΣΤΗΜΑ

Σε ένα UNIX σύστημα το πρώτο πράγμα που χρειάζεται να κάνει κάποιος για να αποκτήσει πρόσβαση είναι να δώσει στον υπολογιστή το username του. Κατόπιν, ο υπολογιστής θα ζητήσει το password που αντιστοιχεί στο username που δόθηκε. Με την επιτυχή είσοδο αυτών των δύο δεδομένων, ο υπολογιστής δίνει πρόσβαση στον χρήστη κατόπιν εκτελεί το shell του χρήστη. Αν πρόκειται για γραφικό περιβάλλον (GUI) τότε θα εμφανιστεί στην οθόνη κάτι αντίστοιχο με αυτό που φαίνεται στο ακόλουθο σχήμα:



Σχήμα 2 Παράδειγμα οθόνης login σε γραφικό περιβάλλον

Ανάμεσα στα άλλα το βασικό σημείο σε αυτή την οθόνη είναι το κεντρικό σημείο όπου το σύστημα ζητάει από τον χρήστη το username του προκειμένου να του επιτρέψει την πρόσβαση. Το πως θα μοιάζει αυτή η οθόνη είναι κάτι τελείως υποκειμενικό, αλλά το σίγουρο είναι ότι κάπου θα υπάρχει το σημείο στο οποίο ζητάει το username. Με την πληκτρολόγηση του username από τον χρήστη, το σύστημα θα ζητήσει το password του και στη συνέχεια, αν ο συνδυασμός username και password είναι σωστός θα επιτρέψει την είσοδο του χρήστη στο σύστημα.

Σε πλήρη αντιστοιχία με το παραπάνω σχήμα, αν δεν είναι διαθέσιμη μια γραφική οθόνη για την είσοδο του username και του password του χρήστη, τότε αυτό που θα εμφανιστεί θα έχει μια μορφή όπως την ακόλουθη:

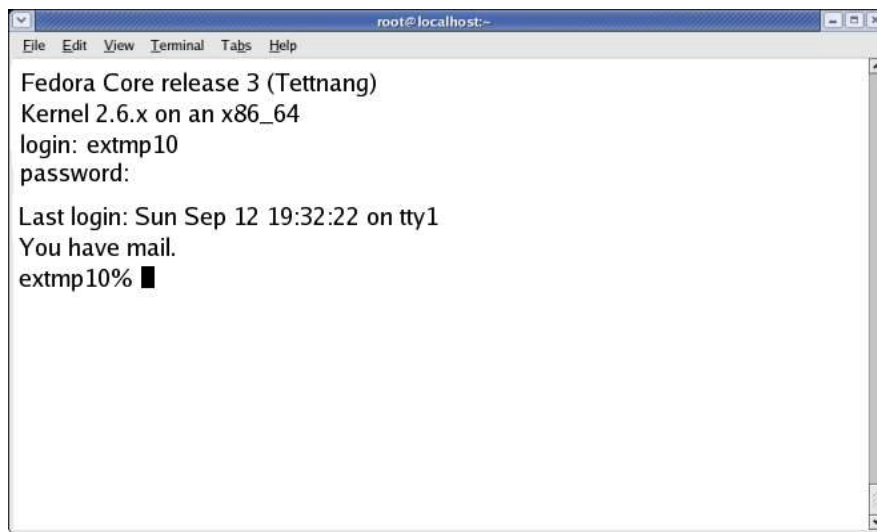
## Εισαγωγή στο Λειτουργικό Σύστημα UNIX



Οι δύο πρώτες γραμμές που εμφανίζονται έχουν γενικές πληροφορίες σχετικά με το σύστημα, και στην τρίτη όπου υπάρχει η λέξη **login**, ο υπολογιστής περιμένει από τον χρήστη να εισάγει το username του. Σαν παράδειγμα, ας θεωρήσουμε ότι το username του χρήστη είναι το extmp10. Αμέσως μετά από αυτό, ζητάει το password:



Κατά την εισαγωγή του, το password **δεν** εμφανίζεται στην οθόνη. Επίσης δεν εμφανίζεται κάτι άλλο που να δηλώνει την εισαγωγή στοιχείων στον υπολογιστή. Αμέσως μετά και αν όλα είναι σωστά, το σύστημα δίνει πρόσβαση στον χρήστη ξεκινώντας ένα shell για αυτόν.



```
root@localhost:~  
File Edit View Terminal Tabs Help  
Fedora Core release 3 (Tettnang)  
Kernel 2.6.x on an x86_64  
login: extmp10  
password:  
Last login: Sun Sep 12 19:32:22 on tty1  
You have mail.  
extmp10% █
```

Κατά την είσοδο του χρήστη στο σύστημα, εμφανίζονται διάφορες πληροφορίες (όπως για παράδειγμα αν έχει e-mail, ή κάποιο μήνυμα από τον διαχειριστή του συστήματος) και στο τέλος εμφανίζεται το **prompt**. Το prompt είναι απλά μια ακολουθία χαρακτήρων ή κάποια λέξη ή κάποιο σύμβολο ή και συνδυασμός όλων αυτών που εμφανίζει το shell προκειμένου να δείξει ότι είναι έτοιμο να δεχτεί εντολές. Το τι θα είναι το prompt είναι κάτι που αποφασίζει ο ίδιος ο χρήστης, αν και γενικά υπάρχει ένα προκαθορισμένο από το σύστημα για κάθε περίπτωση. Στο παράδειγμα που χρησιμοποιήσαμε το prompt είναι το “**extmp10 %**”. Μετά από αυτό εμφανίζεται ο **cursor** (δρομέας) με το χαρακτηριστικό σχήμα (█) που αναβοσβήνει, δείχνοντας την αναμονή του συστήματος για την είσοδο νέων εντολών.

Για την αποσύνδεσή του από το σύστημα, ο χρήστης πρέπει να δώσει την εντολή **logout** ή **exit** που τερματίζει το shell και κατά συνέχεια και την σύνδεσή του.

## ΣΥΣΤΗΜΑ ΑΡΧΕΙΩΝ (FILESYSTEM)

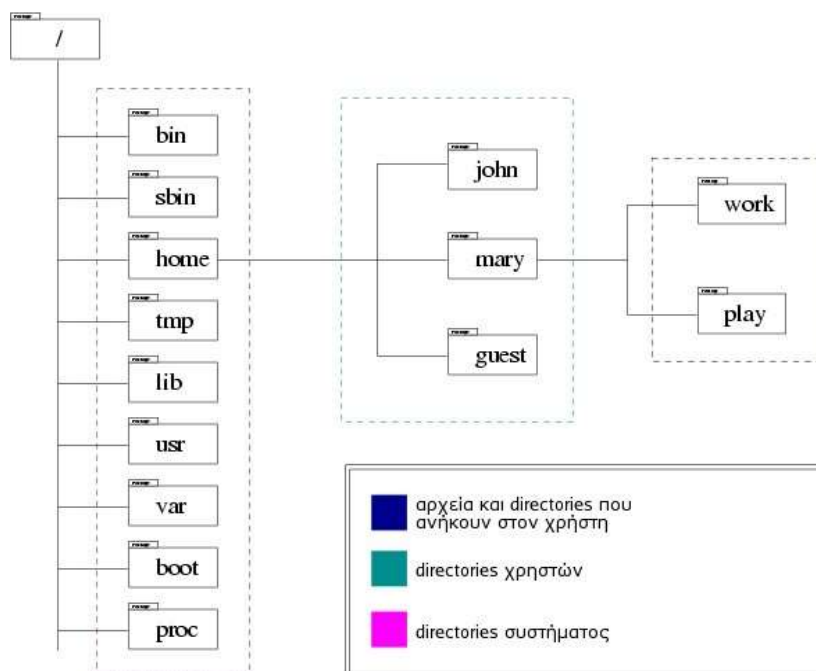
## Γενικά

Με τον όρο αυτό, εννοούμε το πως σε ένα Λ.Σ και ειδικότερα στην περίπτωση μας στο UNIX είναι οργανωμένα τα αρχεία και τα προγράμματα που το απαρτίζουν, όπως επίσης και τα αρχεία στα οποία αποθηκεύουν τα δεδομένα τους οι διάφοροι χρήστες του υπολογιστή. Υπάρχουν δύο βασικές έννοιες στο filesystem:

- **file (αρχείο):** η αντιστοιχία του με τον "πραγματικό" κόσμο είναι τα βιβλία που βρίσκονται μέσα σε μια βιβλιοθήκη. Κάθε ένα από αυτά μπορεί να περιέχει μια πληθώρα από πληροφορίες, όπως ονόματα και τηλέφωνα, αποτελέσματα από μια εργαστηριακή δουλειά αλλά και τα προγράμματα που σχηματίζουν το Λ.Σ. κλπ.
- **directory (κατάλογος):** συνεχίζοντας την προηγούμενη αντιστοίχιση, είναι τα ράφια της βιβλιοθήκης μέσα στα οποία βρίσκονται τα βιβλία – τα αρχεία δηλαδή.

Για όλα αυτά ορίζεται άλλο ένα επίπεδο οργάνωσης, το *filesystem*. Στην αντιστοιχία που έχουμε κάνει ως τώρα, το *filesystem* είναι η βιβλιοθήκη που περιέχει τα ράφια (*directories*) που με την σειρά τους περιέχουν τα βιβλία (*files*). Το ποια ακριβώς είναι η θέση του κάθε πράγματος στην δομή αυτή δεν είναι αυστηρά καθορισμένο. Αποτελεί αποτέλεσμα σύμβασης ανάμεσα στους διάφορους σχεδιαστές λειτουργικών συστημάτων και ο καθένας τα τοποθετεί ανάλογα με τις ανάγκες που προκύπτουν από τον σχεδιασμό τους.

Ωστόσο, σε ένα Λ.Σ UNIX θα συνατήσκει κανείς ένα βασικό μοντέλο δόμησης που έχει αποδειχτεί με την πάροδο του χρόνου αρκετά βολικό για το ακολουθεί κανείς. Ένας λόγος ακόμα, είναι ότι, η κοινή τάση που υπάρχει ανάμεσα στους σχεδιαστές και προγραμματιστές Λ.Σ UNIX είναι να μπορούν τα ίδια προγράμματα να έχουν την δυνατότητα να λειτουργήσουν σε διαφορετική έκδοση ενός UNIX Λ.Σ χωρίς να χρειάζονται σημαντικές ή και καθόλου αλλαγές στο σχεδιασμό τους. Μια τυπική δομή ενός UNIX Λ.Σ. φαίνεται στο Σχ. 3:



Σχήμα 3: Directory structure



Διακρίνουμε δύο βασικές ομάδες από directories:

- Τα directories του συστήματος. Εκεί βρίσκονται αποθηκευμένα αρχεία δεδομένων και προγράμματα που χρησιμοποιεί τόσο το Λ.Σ, όσο και οι χρήστες του υπολογιστή.
- Τα directories των χρηστών. Προκειμένου ο κάθε χρήστης του υπολογιστή να έχει ένα δικό του χώρο για την αποθήκευση των δεδομένων του διασφαλίζοντας ταυτόχρονα την αφάλειά τους, έχει προβλεφτεί από το διαχειριστή του συστήματος να υπάρχει ένα directory που να ανήκει αποκλειστικά και μόνο στον κάθε χρήστη. Από αυτό το directory και μετά, μπορεί ο κάθε χρήστης να φτιάξει άλλα directories αν θέλει και να αποθηκεύσει τα αρχεία του.

Μία πάρα πολύ βασική έννοια σε όλη αυτή τη δομή, είναι αυτή του λεγόμενου "*root directory*". Πρόκειται για ένα ειδικό directory που υπάρχει με σκοπό να μπορεί κανείς να αναφερθεί σε όλα τα directories που υπάρχουν στο σύστημα έχοντας ένα κοινό σημείο αναφοράς. Σε ένα directory μπορεί κάποιος να αναφερθεί με το ονομά του (για παράδειγμα αν θέλει να αναφερθεί στο directory με όνομα "*john*" απλά το ζητάει με το όνομά του, δηλαδή "*john*". Ωστόσο, αυτός ο τρόπος δεν είναι πάντα και ο καλλύτερος δυνατός. Θα μπορούσε να υπάρχει και άλλο ένα directory με ακριβώς το ίδιο όνομα κάπου αλλού στις αποθηκευτικές μονάδες του υπολογιστή. Προκειμένου λοιπόν να είναι ξεκάθαρο το σε πιο directory αναφερόμαστε, πρέπει κανείς να μπορεί να προσδιορίσει το κάθε αρχείο ή directory με ένα μοναδικό τρόπο.

Εδώ είμαστε αναγκασμένοι πια να εισάγουμε την έννοια του "*path*". Σαν "*path*" (μονοπάτι) ορίζουμε την διαδρομή που πρέπει να ακολουθήσει κανείς προκειμένου να βρει ή να αναφερθεί σε ένα directory ή αρχείο. Στο παράδειγμά μας, φαίνεται ότι για να βρει κανείς το directory με όνομα "*john*" πρέπει να να το ψάξει κάτω από το directory με όνομα "*home*" που με την σειρά του είναι κάτω από το directory με όνομα *"/"*. Έτσι, το path στο παράδειγμά μας είναι *"/home/john"*. Το πρώτο – και μόνο το πρώτο – *"/"* που εμφανίζεται είναι το "*root directory*" που αναφέραμε προηγουμένως. Το επόμενο, ή όλα όσα άλλα *"/"* υπάρχουν, χρησιμοποιούνται απλά για να διακρίνονται τα ονόματα των directories που βρίσκονται μέσα σε "*path*" ή ένα "*pathname*" όπως είναι πιο σωστό να λέμε. Εδώ οφείλουμε να κάνουμε ένα διαχωρισμό. Υπάρχουν δύο είδη "*path*".











- **absolut path** (απόλυτο path): είναι το κάθε path που στην αρχή έχει το "*root directory*" (πχ *"/home/john"*).
- **relative path** (σχετικό path): είναι το κάθε path που προσδιορίζεται σε σχέση το path που εργαζόμαστε. Αν για παράδειγμα εργαζόμαστε κάτω από το absolut path *"/home/mary"* και ζητήσουμε από τον υπολογιστή ένα αρχείο με όνομα *"file1"* από το directory *"work"*, τότε ο υπολογιστής θα θεωρήσει, ότι το directory *"work"* που περιέχει το αρχείο *"file1"* βρίσκεται κάτω από το absolute path *"/home/mary"*.

**Σημείωση:** Αυτό που πρέπει να θυμάται κανείς όταν δουλεύει σε ένα UNIX μηχανήμα, είναι ότι τα ονόματα τόσο των αρχείων και directories, όσο και των εντολών είναι **case sensitive**, κατά συνέπεια, ένα αρχείο με όνομα *file1* δεν μπορούμε ποτέ να το καλέσουμε σαν *FILE1*.

## Αδειες χρήσης (*permissions*) αρχείων & directories

Το κάθε αρχείο ή directory που υπάρχει μέσα σε ένα UNIX filesystem ανήκει σε κάποιον χρήστη. Σε κάθε αρχείο ή directory κατά τη στιγμή της δημιουργίας του ορίζονται και οι άδειες χρήσης (*permissions*) που θα έχει. Τα *permissions* στο UNIX ελέγχουν την πρόσβαση σε κάποιο αρχείο ή directory ελέγχοντας με βάση τον χρήστη που ζητάει πρόσβαση ή το *group*. Οι άδειες χρήσης, έχουν να κάνουν με την πρόσβαση σε ένα αρχείο ή directory από τον χρήστη στον οποίο ανήκει, από το *group* στο οποίο ανήκει ο χρήστης ή από τους υπόλοιπους χρήστες του συστήματος.

## Τύποι αρχείων του UNIX

-  Κανονικά αρχεία:
  -  χρησιμοποιούνται για να αποθηκεύουν κείμενο ή εικόνες
  -  βρίσκονται πάντα κάτω από ένα directory
-  Directories
  -  χρησιμοποιούνται για οργανώνουν άλλα αρχεία ή directories
  -  μπορεί να περιέχουν κανονικά αρχεία, αρχεία ειδικού τύπου ή άλλα directories
  -  δεν περιέχουν περιέχουν “πραγματική” πληροφορία όπως κείμενο, αλλά λειτουργούν ως κατάλογοι
-  Ειδικά αρχεία
  -  χρησιμοποιούνται από το Λ.Σ για την διαχείριση του hardware του υπολογιστή. Για ένα UNIX Λ.Σ σχεδόν όλες οι hardware συσκευές χειρίζονται μέσω αρχείων ειδικού τύπου.
  -  συνήθως, βρίσκονται κάτω από το directory `/dev`

## Ονοματολογία αρχείων

### 1. Περιορισμοί

Υπάρχουν κάποιοι περιορισμοί στο τι όνομα μπορεί να δώσει κανείς σε ένα αρχείο στο UNIX. Οι περιορισμοί αυτοί προκύπτουν από το γεγονός ότι διάφορα σύμβολα έχουν ειδική σημασία για το shell. Τα σύμβολα αυτά που δεν μπορούν να χρησιμοποιηθούν μέσα στο όνομα ενός αρχείου (ή directory) είναι τα εξής:

`& ; ( ) | ? \ ' " ` [ ] { } < > $ ! /`

Επίσης, καλό είναι να αποφεύγεται η χρήση του συμβόλου “—” στην αρχή του ονόματος ενός αρχείου. Ο λόγος είναι ότι όλες οι εντολές του UNIX οι οποίες χρησιμοποιούν παραμέτρους, αναγνωρίζουν τότε ένας χαρακτήρας ή μια ακολουθία χαρακτήρων είναι παράμετρος κοιτώντας αν ξεκινούν με τον ειδικό χαρακτήρα “—”.

### 2. Case Sensitive

Το UNIX κάνει διάκριση ανάμεσα στα κεφαλαία και στα μικρά γράμματα (*case-sensitivity*). Αυτό σημαίνει ότι για το UNIX ένα αρχείο με όνομα *NOVEMBER* είναι διαφορετικό από ένα αρχείο με όνομα *november* ή με όνομα *November*.

### 3. File extensions (καταλήξεις του ονόματος των αρχείων)

Οι καταλήξεις που έχουν τα αρχεία στο UNIX μπορεί να χρησιμοποιηθεί για την αναγνώριση του τύπου τους, χωρίς όμως αυτό να είναι αναγκαίο. Μερικές από τις πιο συνηθισμένες καταλήξεις αρχείων είναι οι ακόλουθες:

Κατάληξη	Παράδειγμα	Επεξήγηση
.a	libmath.a	πρόκειται για μια βιβλιοθήκη του συστήματος
.c	pi_calc.c	ένα πρόγραμμα σε C για τον υπολογισμό του $\pi$
.f	integral.f	ένα πρόγραμμα σε FORTRAN για τον υπολογισμό ολοκληρώματος
.o	arctan.o	ένα <i>object</i>
.gz	phonebook.gz	ένα αρχείο που είναι συμπιεσμένο
.tar.gz	home.tar.gz	ένα αρχείο που περιέχει άλλα αρχεία ή directories και κατόπιν έχει συμπιεστεί

Από το τελευταίο παράδειγμα, φαίνεται ότι πολλές φορές είναι χρήσιμο να χρησιμοποιηθεί ένας συνδυασμός από καταλήξεις προκειμένου να γίνει πιο σαφές το περιεχόμενο ενός αρχείου ή η διαδικασία με την οποία δημιουργήθηκε.

### Κ ρ υ φ ά   α ρ χ ε ί α

Τα αρχεία ή τα directories των οποίων το όνομα ξεκινάει με τον ειδικό χαρακτήρα “.” ονομάζονται *κρυφά αρχεία*. Συνήθως, περιέχουν πληροφορίες για τα προγράμματα του συστήματος όπως για παράδειγμα τις προσωπικές επιλογές που ορίζει ο κάθε χρήστης για αυτό το πρόγραμμα. Παράδειγμα:

.cshrc .tcshrc .login .mozilla .emacsrc

### Δ ε σ μ ε υ μ έ ν α   ο ν ό μ α τ α   α ρ χ ε ί ω ν

Ένα αρχείο ή directory, δεν μπορεί να ονομαστεί με τα εξής ονόματα:

- / → είναι το όνομα του αρχικού directory κάτω από το οποίο βρίσκονται όλα τα άλλα
- .
- .. → είναι το όνομα του *parent directory*
- ~ → αντιπροσωπεύει σε κάποια shells το HOME directory

## ΒΑΣΙΚΕΣ ΕΝΤΟΛΕΣ ΔΙΕΧΕΙΡΙΣΗΣ ΑΡΧΕΙΩΝ & DIRECTORIES

### 1. pwd – print working directory

Εμφανίζει το directory στο οποίο βρισκόμαστε. Για παράδειγμα:

```
# pwd
/home/john/test
```

### 2. cd – change directory

Με την εντολή *cd* αλλάζουμε directory.

#### Σύνταξη εντολής:

```
cd [directory]
```

#### Παραδείγματα:

<b>cd</b>	χωρίς κανένα όρισμα, μεταφέρει τον χρήστη το <i>home</i> directory του
<b>cd /</b>	μεταφέρει τον χρήστη στο <i>root</i> directory
<b>cd ..</b>	μεταφέρει τον χρήστη ένα directory πιο πάνω από αυτό που βρίσκεται
<b>cd ../../</b>	μεταφέρει τον χρήστη δύο directories πιο πάνω από αυτό που βρίσκεται
<b>cd /usr/tmp</b>	μεταφέρει τον χρήστη στο directory με <i>απόλυτο path</i> /usr/tmp
<b>cd share/lib</b>	μεταφέρει τον χρήστη στο directory share/lib (το οποίο δεν γράφεται εδώ σαν relative path άρα προϋποθέτει ότι το <i>share/lib</i> βρίσκεται κάτω από το directory στο που βρίσκεται ήδη ο χρήστης)
<b>cd ~john/papers</b>	μεταφέρει τον χρήστη στο directory με όνομα <i>papers</i> που βρίσκεται κάτω από το <i>home directory</i> του χρήστη <i>john</i> (ο συμβολισμός “~john” μεταφράζεται από το σύστημα σαν το <i>home directory</i> του χρήστη με <i>username john</i> )

### 3. mkdir – make directory

Η εντολή *mkdir* δημιουργεί directories

#### Σύνταξη εντολής:

```
mkdir [παράμετροι] directory
```

#### Χρήσιμες παράμετροι:

<b>-p</b>	δημιουργεί το directory και μαζί τα <i>parent</i> directories αν δεν υπάρχουν
<b>-m mode</b>	δημιουργεί το directory και δίνει τα permissions που ορίζονται με το <i>mode</i>

#### Παραδείγματα:

```
# mkdir /home/test
# mkdir -p test1/test2/test3
```

**4. rmdir – remove directory**

Σβήνει ένα directory με την προϋπόθεση ότι είναι άδειο

**Παραδείγματα:**

```
# rmdir test
# rmdir /home/test/test1
```

**5. ls – list directory contents**

Δείχνει τα περιεχόμενα ενός directory και μαί ότι στοιχεία ζητηθούν από τις παραμέτρους της.

**Σύνταξη εντολής:**

*ls [παράμετροι] [όρισμα]*

**Χρήσιμες παράμετροι:**

<b>-a</b>	δείχνει όλα τα αρχεία και τα directories που υπάρχουν μέσα σε ένα directory.
<b>-d</b>	δείχνει μόνο τα ονόματα των directories και όχι τα αρχεία μέσα στο directory
<b>-F</b>	εμφανίζει για κάποιους τύπους αρχείων ένα σύμβολο στο τέλος του ονόματός τους που δείχνει τον τύπο του αρχείου: directories        / sockets         = symbolic links   @ executables     *
<b>-g</b>	εμφανίζει το group στο οποίο ανήκει το αρχείο
<b>-l</b>	δείχνει τα περιεχόμενα του directory αναλυτικά
<b>-t</b>	σε συνδυασμό με την παράμετρο “-l” ταξινομεί την λίστα με τα περιεχόμενα του directory κατά χρονολογική σειρά

**Παραδείγματα:**

```
# ls
debug      linux-2.4  misc      fedora     share
# ls -a
.          debug     linux-2.4  misc      fedora
..         share
# ls -la
drwx-sr-x  5  workshop  acs      512    Jun  7   11:12  .
drwx-xr-x  6  root     sys      512    May 20   09:59  ..
-rwx----- 1  workshop  acs      532    May 20   15:31  .cshrc
-rw-----  1  workshop  acs      525    May 27   21:29  .emacs
-rwx--xr-x  1  workshop  acs       49    May 15   09:44  demofiles
-rwx--x--x  1  workshop  acs      413    May 21   23:52  program10
```

```
drwx-xr-x  3  workshop  acs      512    May  29   11:18  jerry
```

## 6. cp – copy

Η εντολή *cp* αντιγράφει αρχεία ή directories

### Σύνταξη εντολής:

*cp* [παράμετροι] [argument]

### Χρήσιμες παράμετροι:

-i

(*interactive mode*) περιμένει για επιβεβαίωση των ενεργειών της από το χρήστη πριν να εκτελέσει οτιδήποτε

-r

αντιγράφει ένα directory δημιουργώντας αντίγραφα και από directories που ίσως υπάρχουν μέσα σε αυτό, διατηρώντας την αρχική δομή

### Παραδείγματα:

# cp old\_filename new\_filename

## 7. mv – move

Η εντολή *mv* μετακινεί αρχεία ή directories

### Σύνταξη εντολής:

# mv [παράμετροι] [argument]

### Χρήσιμες παράμετροι:

-i

(*interactive mode*) περιμένει για επιβεβαίωση των ενεργειών της από το χρήστη πριν να εκτελέσει οτιδήποτε

-f

μετακινεί το αρχείο ή το directory ακόμα και αν χρειαστεί να γράψει πάνω σε αρχείο που υπάρχει ήδη

### Παραδείγματα:

mv old\_filename new\_filename

## 8. rm – remove

Η εντολή *rm* σβήνει αρχεία ή directories

### Σύνταξη εντολής:

*rm* [παράμετροι] [argument]

### Χρήσιμες παράμετροι:

-i

περιμένει για επιβεβαίωση πριν να εκτελέσει την μετακίνηση

-r

σβήνει ένα directory, αφού πρώτα σβήσει και τα περιεχόμενά του

-f

σβήνει χωρίς να περιμένει για επιβεβαίωση

### Παραδείγματα:

# rm filename

```
# rm -rf directory_name
```

### 9. chmod – change mode

Η εντολή *chmod* αλλάζει τα permissions ενός αρχείου ή ενός directory. Ένα αρχείο μπορεί να έχει τρία δυνατά modes read, write και execute. Το καθένα από αυτά αντιπροσωπεύεται από ένα ακέραιο αριθμό ή ένα γράμμα:

συμβολισμός με αριθμούς	read	4
	write	2
	execute	1
συμβολισμός με χαρακτήρες	read	r
	write	w
	execute	x

#### Σύνταξη εντολής:

```
chmod mode file
```

όπου το *mode* μπορεί να γραφεί είτε με την αριθμητική του μορφή, είτε με τη συμβολική του μορφή.

Ο έλεγχος των permissions γίνεται σε επίπεδο χρήστη, ομάδας χρηστών και του συνόλου των χρηστών που υπάρχουν σε ένα σύστημα. Για καθένα από αυτά, ορίζονται οι άδειες για κάθε ένα από τα τρία δυνατά modes (r,w,x). Ο τρόπος με τον οποίο θέτουμε τα permissions είναι ο εξής: αποφασίζουμε για τον χρήστη (**u**) στον οποίο ανήκει το αρχείο, το group (**g**) στο οποίο ανήκει το αρχείο και για το σύνολο των χρηστών του συστήματος (**o**), τι άδειες πρόσβασης θα έχουν για το αρχείο αυτό. Αν θέλουμε ο χρήστης να έχει δυνατότητα εγγραφής (mode **2**), ανάγνωσης (mode **4**) και εκτέλεσης (mode **1**) του αρχείου, σημαίνει ότι συνολικά για τον χρήστη (**u**) το mode είναι **7** (2+4+1). Εργαζόμαστε με τη ίδια λογική και για το group (**g**) και τους υπόλοιπους χρήστες (**o**). Έτσι, ένα αποδεκτό mode για ένα αρχείο θα ήταν το **744**, που σημαίνει, **read/write/execute** για τον χρήστη, **read** για το group και **read** επίσης για τους υπόλοιπους χρήστες. Στην περίπτωση που θέλαμε να χρησιμοποιήσουμε την συμβολική γραφή για τα permissions θα γράφαμε για mode το **u=rwx,go=r**.

Στην περίπτωση που θέλουμε απλά να αλλάξουμε τα permissions προσθέτοντας ή αφαιρώντας άδειες χρήσης, μπορούμε να χρησιμοποιήσουμε τους τελεστές "-" και "+" σε συνδυασμό με το είδος της άδειας που θέλουμε να δώσουμε, δηλαδή ένα mode **u-r** σημαίνει **αφαίρεσε την δυνατότητα ανάγνωσης του αρχείου από τον χρήστη στον οποίο ανήκει αυτό το αρχείο**.

#### Χρήσιμες παράμετροι:

- f** αλλάζει τα permissions χωρίς να περιμένει για επιβεβαίωση
- R** αλλάζει τα permissions μέσα σε όλη τη δομή του directory

#### Παραδείγματα:

Έστω ότι για ένα αρχείο με όνομα file1 θέλουμε να δώσουμε την δυνατότητα ανάγνωσης, εγγραφής και εκτέλεσης για τον χρήστη στον οποίο ανήκει το αρχείο, δυνατότητα ανάγνωσης και εκτέλεσης για το group και το σύνολο των χρηστών. Αυτό γίνεται ως εξής:

```
# chmod 754 file1
```

ή  
# chmod u=rwx,g=rx,o=r file1

#### 10.chown – change owner

Η εντολή *chown* αλλάζει το καθεστώς ιδιοκτησίας ενός αρχείου η ενός directory

##### Σύνταξη εντολής:

*chown* [παράμετροι] [argument]

##### Χρήσιμες παράμετροι:

**-R**           αλλάζει τον ιδιοκτήτη για όλα τα  
                  αρχεία και directories σε όλο το  
                  δέντρο

##### Παραδείγματα:

# chown -R john directory1

#### 11.chgrp – change group

Η εντολή *chgrp* αλλάζει το group ενός αρχείου η ενός directory

##### Σύνταξη εντολής:

*chgroup* [παράμετροι] [argument]

##### Χρήσιμες παράμετροι:

**-R**           αλλάζει το group για όλα τα  
                  αρχεία και directories σε όλο το  
                  δέντρο

##### Παραδείγματα:

# chgrp group1 file1

#### 12.echo

Η εντολή *echo* τυπώνει στην οθόνη το όρισμα που πέρνει

##### Σύνταξη εντολής:

*echo* [όρισμα]

##### Παραδείγματα:

# echo This is a test  
This is a test

#### 13.cat – concatenate files

Με την εντολή *cat* προβάλλουμε τα περιεχόμενα ενός αρχείου

##### Σύνταξη εντολής:

# *cat* filename(s)

##### Παραδείγματα:

# cat file1 file2 file2

#### 14.more, less, pg

Οι εντολές *more*, *less* και *pg* έχουν παρόμοια λειτουργία με την *cat* αλλά με την επιπλέον δυνατότητα να εμφανίζουν τμηματικά τα περιεχόμενα των αρχείων.



**Σύνταξη εντολής:**

*% more filename*

**15.head**

Εμφανίζει ένα συγκεκριμένο αριθμό γραμμών από την αρχή ενός αρχείου. Χωρίς όρισμα, εμφανίζει τις 10 πρώτες γραμμές από το αρχείο.

**Σύνταξη εντολής:**

*head [ορίσματα] filename*

**Χρήσιμες παράμετροι:**

*-αριθμός* εμφανίζει τόσες γραμμές από την αρχή ενός αρχείου, όσες και ο *αριθμός* που δίνεται

**Παράδειγμα:**

*# head filename*

ή

*# head -100 filename*

**16.tail**

Εμφανίζει ένα συγκεκριμένο αριθμό γραμμών από το τέλος ενός αρχείου. Χωρίς όρισμα, εμφανίζει τις τελευταίες 10 γραμμές από το αρχείο.

**Σύνταξη εντολής:**

*tail [ορίσματα] filename*

**Χρήσιμες παράμετροι:**

*-αριθμός* εμφανίζει τόσες γραμμές από το τέλος ενός αρχείου, όσες και ο *αριθμός* που δίνεται

**Παράδειγμα:**

*# tail filename*

ή

*# tail -100 filename*

**17.date**

Εμφανίζει την ημερομηνία.

**Σύνταξη εντολής:**

*date [ορίσματα] [+format]*

**Χρήσιμες παράμετροι:**

*+format* καθορίζει την μορφή με την οποία θέλουμε να εμφανιστεί η ημερομηνία

*%a* εμφανίζει τα ονόματα των ημερών με συντομογραφία

*%h* εμφανίζει τα ονόματα των μηνών με συντομογραφία

*%j* εμφανίζει την ημέρα σαν αριθμό από το 1 μέχρι το 366

*%D* εμφανίζει την ημερομηνία με την μορφή ΜΗΝΑΣ/ΗΜΕΡΑ/ΕΤΟΣ (MM/DD/YY)

*%H* εμφανίζει την ώρα σαν αριθμό από το 0 μέχρι το 23

**+format** καθορίζει την μορφή με την οποία θέλουμε να εμφανιστεί η ημερομηνία  
%T εμφανίζει την ώρα σαν ΩΡΑ:ΛΕΠΤΟ:ΔΕΥΤΕΡΟΛΕΠΤΟ (HH:MM:SS)  
%y εμφανίζει τα τελευταία δύο ψηφία του έτους

**Παράδειγμα:**

```
# date  
Sun Sep 12 11:42:52 EEST 2003
```

```
# date +%a%t%D  
Sun 10/12/03
```

**18.man**

Η εντολή *man* (*manual*) εμφανίζει στην οθόνη την περιγραφή και οδηγίες σχετικά με κάποια από τις εντολές του συστήματος. Είναι ένας πολύ καλός τρόπος για να μάθει κανείς την χρήση των εντολών του συστήματος.

**Σύνταξη εντολής:**

*man* [παράμετροι] όνομα εντολής ή keyword

**Χρήσιμες παράμετροι:**

-k ψάχνει το σύνολο των οδηγιών χρήσης του συστήματος (manual pages) για αναφορές πάνω στην λέξη "κλειδί" που δίνεται

**Παράδειγμα:**

```
# man man  
(εμφανίζει οδηγίες για την χρήση και τις δυνατότητες της εντολής man)
```

```
# man ls  
(εμφανίζει οδηγίες για την χρήση και τις δυνατότητες της εντολής ls)
```

```
# man -k date  
(εμφανίζει τα manual pages μέσα στα οποία εμφανίζεται η λέξη "κλειδί" date)
```

**19.ps**

Σε ένα UNIX Λ.Σ όπως έχει αναφερθεί ρητά και στην εισαγωγή μπορούν πολλοί διαφορετικοί χρήστες να εκτελέσουν διεργασίες ανεξάρτητα ο ένας από τον άλλο. Με την εντολή *ps* μπορεί κανείς να δει ποιες διεργασίες έχει εκτελέσει και σε τι κατάσταση βρίσκονται.

**Σύνταξη εντολής:**

*ps* [παράμετροι]

**Χρήσιμες παράμετροι:**

**a** εμφανίζει όλες τις διεργασίες που εκτελούνται από το σύστημα ανεξάρτητα από το αν έχουν ξεκινήσει από τερματικό ή όχι

**u** εμφανίζει τις διεργασίες με βάση τον χρήστη στο οποίο ανήκουν

**w** εκτενές output

**x** όλες οι διεργασίες ανεξάρτητα από το αν έχουν ξεκινήσει από τερματικό ή όχι με περισσότερες λεπτομέρειες

**Παράδειγμα:**

```
% ps a
PID TTY    STAT  TIME  COMMAND
4147 pts/10  S      0:00  -tcsh
4204 pts/10  S      0:00  su -
4207 pts/10  S      0:00  -csh
4372 pts/12  S      0:01  ssh host10-g09
```

```
%ps auxx
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
xfx	3100	0.0	0.9	1356	1104	?	S	Oct 10	0:05	xfx -droppriv
daemon	3119	0.0	0.0	1500	604	?	S	Oct 10	0:00	/usr/sbin/atd
dbus	3122	0.0	0.0	1864	896	?	R	Oct 10	0:00	dbus-daemon
john	4987	0.0	0.0	1024	536	ttyp1	R	Oct 10	0:00	/bin/tcsh

## 20.who

Δείχνει τους χρήστες που είναι συνδεδεμένοι στον υπολογιστή

### Σύνταξη εντολής:

```
who [am i]
```

### Παράδειγμα:

```
# who
wmtell          tty1          Oct 12 20:15 (apple.acs.ohio-s)
fbwalk          tty2          Oct 10 23:21 (127.99.25.8)
david           tty3          Oct 12 22:32 ppp-0a.gwdg-te.munh.de
john            tty4          Oct 12 21:12 localhost:0
```

```
%who am i
john pts/10    Oct 12 17:46
```

## 21.which

Εμφανίζει το full path μιας εντολής

### Σύνταξη εντολής:

```
which εντολή
```

### Παράδειγμα:

```
# which man
/usr/bin/man
```

## 22.passwd

Αλλάζει τον κωδικό πρόσβασης (password) ενός χρήστη

### Σύνταξη εντολής:

```
passwd [username]
```

### Παράδειγμα:

```
# passwd
Enter old passwd: *****
Enter new password: *****
Confirm new password: *****
```

## SHELLS

Με τον όρο "shell", εννοούμε το πρόγραμμα εκείνο που δρα ως ενδιάμεσος κρίκος ανάμεσα στον υπολογιστή και τον χρήστη. Μέσω του shell, είναι δυνατόν να δωθούν εντολές στον υπολογιστή, αναλαμβάνει την διασύνδεση του ανθρώπου με τον υπολογιστή. Για όσους είναι εξοικωμένοι με το DOS, το UNIX shell είναι το ανάλογο του command.com.

Υπάρχουν δύο μεγάλες κατηγορίες shells. Τα *γραφικά shells* και τα *command line shells*. Ένα καλό παράδειγμα *γραφικού shell* είναι ο *Explorer*<sup>®</sup> που εμφανίζεται στο περιβάλλον των *Microsoft Windows*<sup>®</sup>. Μέσα από αυτό, έχει κανείς την δυνατότητα να ελέγξει τον υπολογιστή και να χρησιμοποιήσει τα προγράμματα που είναι εγκατεστημένα. Είναι ένα περιβάλλον φιλικό προς τον χρήστη, περιέχει menu και εικονίδια που κατά κάποιο τρόπο αυτοματοποιούν τις διαδικασίες εκτέλεσης ενός προγράμματος. Από την άλλη πλευρά είναι συνηθισμένο διάφορα λειτουργικά συστήματα να διαθέτουν και ένα *command line shell* προκειμένου να δίνεται η δυνατότητα στο χρήστη να μπορεί να έχει καλύτερο έλεγχο στη διαχείριση του υπολογιστή.

Στο UNIX υπάρχουν πολλά διαφορετικά *command line shells* με διάφορες δυνατότητες το καθένα. Μερικά από τα πιο συνηθισμένα είναι τα *sh*, *bash*, *ksh*, *csh* και *tcsh*.

Ο τρόπος με τον οποίο ένα shell εκτελεί τις εντολές του συστήματος, είναι ο εξής: έστω ότι ο χρήστης θέλει να δει τα περιεχόμενα ενός *directory*. Γράφει στο shell την εντολή *ls*. Το shell προκειμένου να εκτελέσει στην εντολή, πρέπει να βρει αρχικά να βρεί που είναι η εντολή ώστε να μπορέσει να την "διαβάσει" και να την "εκτελέσει" (το πως γίνεται αυτό θα το δούμε αργότερα). Αφού γίνει αυτό το βήμα η εντολή εκτελείται και το shell παρουσιάζει την οθόνη το αποτέλεσμα της εντολής.

Το κάθε shell μπορεί να εκτελέσει εκτός από τις εντολές του UNIX που έχουμε ήδη παρουσιάσει (όπως την *ls* για παράδειγμα) και ένα αριθμό εντολών που δεν υπάρχουν (στις περισσότερες περιπτώσεις) μέσα στο Λ.Σ., αλλά είναι, όπως λέμε, *εσωτερικές εντολές του shell* (*build-in shell commands*). Υπάρχουν περιπτώσεις που κάποιες από τις εσωτερικές εντολές ενός shell, ταυτίζονται με τις εσωτερικές εντολές ενός άλλου shell. Για παράδειγμα, σχεδόν όλα τα shells διαθέτουν την εντολή *pwd* (*print working directory*) που δείχνει στον χρήστη σε ποιο *directory* βρίσκεται.

Ένα άλλο χαρακτηριστικό των shells είναι οι *μεταβλητές του shell* (*shell variables*). Διακρίνουμε δύο κατηγορίες μεταβλητών του shell: τις *τοπικές* (*local*) και τις μεταβλητές περιβάλλοντος (*environment variables*). Η διαφορά τους έγκειται στο ότι οι τοπικές μεταβλητές αφορούν το shell στο οποίο ορίζονται, ενώ οι μεταβλητές περιβάλλοντος χρησιμοποιούνται τόσο από το shell στο οποίο ορίζονται όσο και από τις διεργασίες που ξεκινούν από αυτό το shell.

Για να γίνει σαφής η χρησιμότητά τους, θα χρησιμοποιήσουμε το εξής παράδειγμα: έστω ότι χρειάζεται αναφερόμαστε συχνά σε ένα *directory* με το full path του, ας πούμε το */usr/local/bin*. Για να μην χρειάζεται κάθε φορά να γράφουμε όλο το path, μπορούμε να ορίσουμε στο shell μια μεταβλητή με όνομα *lbin* με αποτέλεσμα κάθε φορά που αναφερόμαστε στην μεταβλητή με όνομα *lbin* το shell θα ξέρει ότι στην πραγματικότητα αναφερόμαστε στο *directory* με full path name */usr/local/bin*. Στην πράξη, η χρήση των μεταβλητών έχει πολύ μεγαλύτερη χρησιμότητα από ότι σε αυτό το παράδειγμα και αυτό θα φανεί καλύτερα παρακάτω.

Όταν ξεκινάει η εκτέλεση ενός shell ορίζονται αυτόματα κάποιες μεταβλητές από το ίδιο το shell. Η σκοπιμότητα πίσω από αυτό, είναι ότι ο χρήστης χρειάζεται από την αρχή κάποιο βασικό περιβάλλον για να εργαστεί. Έτσι είναι αναμενόμενο το shell να ξέρει για παράδειγμα τα *directories* που βρίσκονται μερικές από τις πιο βασικές εντολές του συστήματος όπως η *ls* ή να μπορεί ανά πάσα στιγμή να δώσει πληροφορίες στα προγράμματα που εκτελεί ο χρήστης σχετικά με το ποιος τα χρησιμοποιεί, ή ποιο είναι το *home directory* του χρήστη, ή πως ονομάζεται ο υπολογιστής στον οποίο δουλεύει ο συγκεκριμένος χρήστης.

Αφού όμως μιλάμε για "μεταβλητές", είναι αναμενόμενο να υποθέσει κανείς ότι ανεξάρτητα με την τιμή που δίνει το shell σε κάθε μια από αυτές τις μεταβλητές, ο χρήστης μπορεί να αλλάξει την τιμή τους ώστε να τις προσαρμόσει στις ανάγκες του.

## BUILD-IN COMMANDS

Μερικές από τις πιο συνηθισμένες build-in εντολές δύο δημοφιλών shells, του *sh* και του *tcsh*.

## ➤ sh

built-in εντολή	Επεξήγηση
. [παράμετρος]	διάβασε και εκτέλεσε τις εντολές που περιέχονται στο αρχείο που δίνεται σαν <i>παράμετρος</i>
cd [παράμετρος]	( <i>change directory</i> ) μας μετακινεί στο directory που δίνεται σαν <i>παράμετρος</i> . Αν δεν δοθεί παράμετρος, μας μετακινεί στο home directory μας
pwd	( <i>print working directory</i> ) εμφανίζει το directory μέσα στο οποίο βρισκόμαστε
exec εντολή	εκτελεί την εντολή στην θέση του shell (ζητάμε την εκτέλεση μιας εντολής και ταυτόχρονα τον τερματισμό της λειτουργίας του shell)
typeset [var=value]	ορίζει την μεταβλητή με όνομα <i>var</i> και δίνει σε αυτήν την τιμή <i>value</i> . Χωρίς παραμέτρους, εμφανίζει τις μεταβλητές του shell που έχουν ήδη οριστεί
alias	δημιουργεί μια συντομογραφία για μια έκφραση
read	διαβάζει μια γραμμή από την είσοδο

## ➤ tcsh

built-in εντολή	Επεξήγηση
source [παράμετρος]	διάβασε και εκτέλεσε τις εντολές που περιέχονται στο αρχείο που δίνεται σαν <i>παράμετρος</i>
cd [παράμετρος]	( <i>change directory</i> ) μας μετακινεί στο directory που δίνεται σαν <i>παράμετρος</i> . Αν δεν δοθεί παράμετρος, μας μετακινεί στο home directory μας
pwd	( <i>print working directory</i> ) εμφανίζει το directory μέσα στο οποίο βρισκόμαστε
exec εντολή	εκτελεί την εντολή στην θέση του shell (ζητάμε την εκτέλεση μιας εντολής και ταυτόχρονα τον τερματισμό της λειτουργίας του shell)
set [var=value]	ορίζει την τοπική μεταβλητή με όνομα <i>var</i> και δίνει σε αυτήν την τιμή <i>value</i> . Χωρίς παραμέτρους, εμφανίζει τις τοπικές μεταβλητές του shell που έχουν ήδη οριστεί
unset	καταργεί κάποια τοπική μεταβλητή
setenv	ορίζει την μεταβλητή περιβάλλοντος με όνομα <i>var</i> και δίνει σε αυτήν την τιμή <i>value</i> . Χωρίς παραμέτρους, εμφανίζει τις μεταβλητές περιβάλλοντος του shell που έχουν ήδη οριστεί
unsetenv	καταργεί κάποια μεταβλητή περιβάλλοντος
alias	δημιουργεί μια συντομογραφία για μια έκφραση

## ΔΙΑΧΕΙΡΙΣΗ ΔΙΕΡΓΑΣΙΩΝ (JOB CONTROL) FOREGROUND & BACKGROUND

Το Λ.Σ UNIX είναι σχεδιασμένο με το σκεπτικό ότι ο υπολογιστής πρέπει να δίνει την δυνατότητα σε διαφορετικούς χρήστες να μπορούν ταυτόχρονα να χρησιμοποιούν τόσο την επεξεργαστική του ισχύ όσο και τα αποθηκευτικά μέσα που διαθέτει, ταυτόχρονα (*multiuser, multitasking*). Τα shells που υπάρχουν στο UNIX ακολουθούν αυτή τη λογική και δίνουν την δυνατότητα στον χρήστη να μπορεί να εκτελεί πολλαπλές διεργασίες χωρίς να χρειάζεται για κάθε μια από αυτή και ένα διαφορετικό shell. Άλλωστε, το shell θυμίζουμε ότι έχει σαν σκοπό τη διασύνδεση του υπολογιστή με τον χρήστη και αν οι δυνατότητές του περιορίζονταν στο να δίνει προς εκτέλεση μια εντολή και κατόπιν να περιμένει να τελειώσει για να μπορέσει να δεχτεί την επόμενη, τότε η αποδοτικότητα ως προς το χρόνο θα ήταν πολύ χαμηλή.

Εδώ εμφανίζεται η έννοια του *background* και του *foreground*. Οι έννοιες αυτές είναι λίγο αόριστες και θα επιχειρήσουμε να δώσουμε δύο πολύ γενικούς ορισμούς. Σαν *foreground* εννοούμε την κατάσταση στην οποία οι εντολές και τα προγράμματα εκτελούνται και βρίσκονται σε άμεση αλληλεπίδραση με τον χρήστη. Σαν *background* εννοούμε την κατάσταση στην οποία οι διεργασίες εκτελούνται χωρίς να αλληλεπιδρούν με τον χρήστη. Για να γίνει πιο κατανοητό, ας θεωρήσουμε το ακόλουθο παράδειγμα: ένας χρήστης θέλει να εκτελέσει ένα αρχείο εντολών το οποίο αναζητεί πάνω στο δίσκο όλα τα αρχεία με μέγεθος μεγαλύτερο από 1 Kb. Αυτή είναι μια διεργασία που κατά πασα πιθανότητα θα αργήσει να τελειώσει. Στο μεταξύ ο χρήστης πρέπει να περιμένει τον τερματισμό της προκειμένου να εκτελέσει κάποια άλλη εντολή από το ίδιο shell. Εναλλακτικά, μπορεί να αναθέσει, να "στείλει" όπως λέμε, την διεργασία αυτή στο background και όσο αυτή εκτελείται, το shell να είναι σε κατάσταση αναμονής για να δεχτεί και άλλες εντολές. Ένα άλλο παράδειγμα, θα ήταν η περίπτωση στην οποία ένας χρήστης θέλει για όσο χρονικό διάστημα είναι συνδεδεμένος με τον υπολογιστή να εκτελείται ένα πρόγραμμα που θα του αναφέρει ποιο άλλοι χρήστες είναι συνδεδεμένοι ταυτόχρονα με αυτόν στον ίδιο υπολογιστή, ή αν κάποιος χρήστης αποσυνδέθηκε κλπ.

Για να γίνουν δυνατά όλα τα παραπάνω, πρέπει ή για κάθε μια διεργασία να ξεκινάει ο χρήστης ένα διαφορετικό shell για να μπορέσει να ζητήσει την εκτέλεση των εντολών αυτών, ή να μπορεί από το ίδιο shell να στείλει στο background κάποιες από αυτές και να έχει την δυνατότητα να συνεχίσει την δουλειά του όσο αυτές εκτελούνται.

Ο τρόπος με τον οποίο το shell καταλαβαίνει ότι μια διεργασία πρέπει να εκτελεστεί στο background, είναι να δώσουμε στο τέλος της εντολής το σύμβολο **&**.

Για παράδειγμα, αν θέλουμε να εκτελέσουμε το πρόγραμμα με όνομα *pi\_evaluation* το οποίο υπολογίζει την τιμή του αριθμού π με 1.000.000.000.000 δεκαδικά ψηφεία, μπορούμε να το στείλουμε στο background ως εξής:

```
# pi_evaluation&
#
```

Αν έχουμε ήδη ξεκινήσει μια διεργασία και θέλουμε να την στείλουμε στο background γιατί στο μεταξύ χρειάζεται να χρησιμοποιήσουμε το shell, τότε μπορούμε να χρησιμοποιήσουμε το πλήκτρο **Z** σε συνδυασμό με το πλήκτρο **Control (Control-Z και συμβολικά ^Z)**.

Για να δούμε ποιες διεργασίες έχουμε στείλει στο background από το shell στον οποίο εργαζόμαστε, χρησιμοποιούμε την εντολή **jobs**:

```
# jobs
[1] + Running          ls -lR / 2 >& /dev/null
[2] - Running          iostat -k 1
```

Αν θέλουμε να την επαναφέρουμε στο foreground, μπορούμε να χρησιμοποιήσουμε την εντολή **fg** με παράμετρο τον α/α της διεργασίας.

```
# fg %2
```

## ΕΠΕΞΕΡΓΑΣΤΕΣ ΚΕΙΜΕΝΟΥ (EDITORS) ΣΤΟ UNIX

Όπως σε κάθε Λ.Σ., έτσι και στο UNIX, υπάρχει μια ποικιλία από επεξεργαστές κειμένου. Μερικοί από τους πιο συνηθισμένους, είναι οι **vi**, **emacs**, **pico** και **joe**. Ο κάθενας από αυτούς έχει τα πλεονεκτήματα και τα μειονεκτήματά του και εξαρτάται από τον χρήστη ποιον θα διαλέξει. Από αυτούς τους τέσσερις ο **vi** και ο **emacs** είναι αυτοί που έχουν τις περισσότερες δυνατότητες, υστερώντας όμως σε απλότητα σε σχέση με τους άλλους δύο.

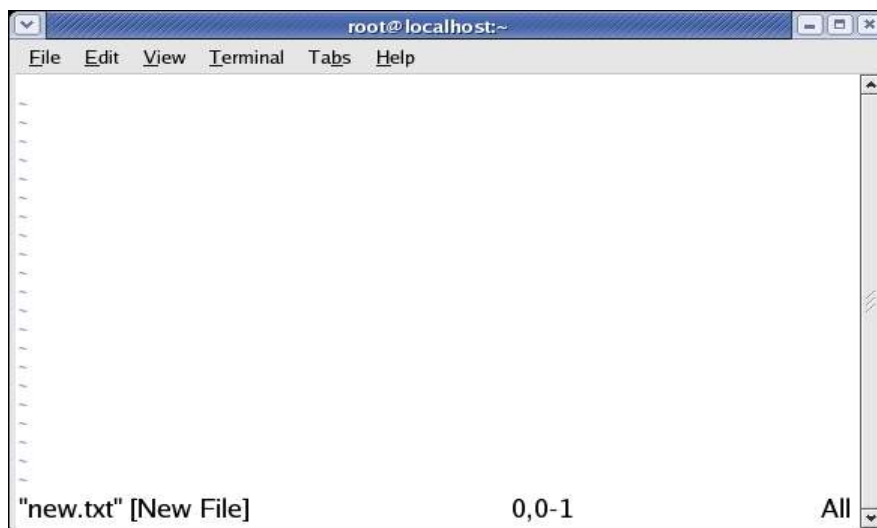
Το βασικό χαρακτηριστικό αυτών των επεξεργαστών κειμένου είναι ότι σχεδόν αποκλειστικά στερούνται την δυνατότητα να δημιουργήσουν κάτι παραπάνω από κείμενο, δεν έχουν δηλαδή την δυνατότητα να χρησιμοποιήσουν γραφικά ή εικόνες. Ωστόσο αν συνδιαστούν με το L<sup>A</sup>T<sub>E</sub>X μπορούν να βοηθήσουν στην δημιουργία σύνθετων εγγράφων με ποικίλες μορφές.

Εκτός από αυτούς τους επεξεργαστές κειμένου, υπάρχουν και άλλοι μέσα σε εφαρμογές γραφείου που υπάρχουν στο UNIX, όπως το Openoffice.org που διαθέτει ένα επεξεργαστή κειμένου με όλες τις δυνατότητες του Microsoft Word®

## Ο επεξεργαστής κειμένου vi

Ο **vi** είναι ένας από τους πρώτους editors που εμφανίστηκαν στο UNIX. Έχει δύο καταστάσεις λειτουργίας, το *command mode* και το *insert mode*. Το *insert mode* χρησιμοποιείται για να γίνει εισαγωγή κειμένου. Το *command mode* για να δοθούν εντολές, όπως, πρόσθεση γραμμών, αντιγραφή λέξεων/γραμμών, διαγραφή κειμένου κλπ μαζικά σε όλο το κείμενο ή σε επιμέρους τμήματά του.

Όταν χρησιμοποιούμε τον vi για την δημιουργία ενός νέου κειμένου, αυτό που βλέπουμε είναι το εξής:



Σχήμα 4 Η εντολή που δώσαμε από το prompt ήταν η: vi new.txt

Στα αριστερά της οθόνης εμφανίζεται το χαρακτηριστικό σύμβολο ~ που οριοθετεί το χώρο στον οποίο θα γραφτεί και θα εμφανίζεται το κείμενο. Στο κάτω μέρος της οθόνης, εμφανίζονται πληροφορίες σχετικά με το αρχείο που θα δημιουργηθεί. Για την ώρα αναφέρει ότι πρόκειται για νέο αρχείο (*New file*) και τις συντεταγμένες της θέσης του cursor (0,0-1).

Στην φάση αυτή, είμαστε σε **command mode**. Ανεξάρτητα με το αν πρόκειται για καινούριο αρχείο ή όχι, το αρχικό mode του vi είναι το *command mode*. Για να μπορέσουμε να γράψουμε κείμενο, πρέπει να πάμε σε **insert mode**, κάτι που γίνεται πατώντας το πλήκτρο **"i"** (δηλαδή *insert*). Εδώ πρέπει να τονιστεί, ότι όπως και στα ονόματα των αρχείων και στις εντολές τα κεφαλαία γράμματα αντιμετωπίζονται διαφορετικά από τα μικρά, έτσι συμβαίνει και με το vi editor. Το πλήκτρο **"I"** (Ι-κεφαλαίο) έχει διαφορετική σημασία από το **"i"** (ι-μικρό).



```

root@localhost:~
File Edit View Terminal Tabs Help
Αυτή είναι η πρώτη γραμμή
Αυτή είναι η δεύτερη γραμμή
εδώ υπάρχουν μερικά κενά

εδώ βρίσκεται ο cursor
εδώ η γραμμή ξεκινάει από πιο μέσα
1234567890~!@#$%^&*(*)+|{}~<>?\\=-
-- INSERT --                    5,20                    All

```

Σχήμα 5 Παράδειγμα εισαγωγής κειμένου

Για να αποθηκευτεί το κείμενο που γράψαμε, πρέπει να επιστρέψουμε σε **command mode**. Αυτό γίνεται με το πλήκτρο **ESC**. Κατόπιν, με το χαρακτήρα " : " μεταφερόμαστε στην γραμμή εντολών του vi (χαμηλά στην οθόνη) όπου γράφουμε **w**, δηλαδή *write* και πατάμε **enter**.

```

root@localhost:~
File Edit View Terminal Tabs Help
Αυτή είναι η πρώτη γραμμή
Αυτή είναι η δεύτερη γραμμή
εδώ υπάρχουν μερικά κενά

εδώ βρίσκεται ο cursor
εδώ η γραμμή ξεκινάει από πιο μέσα
1234567890~!@#$%^&*(*)+|{}~<>?\\=-
:w

```

```

root@localhost:~
File Edit View Terminal Tabs Help
Αυτή είναι η πρώτη γραμμή
Αυτή είναι η δεύτερη γραμμή
εδώ υπάρχουν μερικά κενά

εδώ βρίσκεται ο cursor
εδώ η γραμμή ξεκινάει από πιο μέσα
1234567890~!@#$%^&*(*)+|{}~<>?\\=-
"new.txt" [New] 7, 195C written

```



## Σύντομος οδηγός χρήσης του vi

Εντολές για την μετακίνηση του cursor	Εντολές διαγραφής κειμένου
ο δείκτης <b>n</b> αντιπροσωπεύει αριθμό και η χρήση του είναι προαιρετική	ο δείκτης <b>n</b> αντιπροσωπεύει αριθμό και η χρήση του είναι προαιρετική
(n) <b>h</b> αριστερά n διαστήματα	<b>dd</b> σβήνει την τρέχουσα γραμμή
(n) <b>j</b> κάτω n σειρές	(n) <b>dd</b> σβήνει τις n γραμμές
(n) <b>k</b> πάνω n σειρές	(n) <b>dw</b> σβήνει τις n λέξεις
(n) <b>l</b> δεξιά n διαστήματα	<b>D</b> σβήνει μέχρι το τέλος της γραμμής
<b>^F</b> εμφανίζει την επόμενη σελίδα	<b>x</b> σβήνει μέχρι τον τρέχων χαρακτήρα
<b>^B</b> εμφανίζει την προηγούμενη σελίδα	(n) <b>x</b> σβήνει τους n χαρακτήρα
<b>^D</b> προχωράει μισή σελίδα κάτω	<b>X</b> σβήνει τους προηγούμενους χαρακτήρες
<b>^U</b> προχωράει μισή σελίδα πάνω	Εντολές αλλαγής κειμένου
το σύμβολο <b>^</b> αντιπροσωπεύει το <b>Control</b>	<b>cw</b> αλλάζει τους χαρακτήρες μέχρι το τέλος της λέξης
<b>H</b> μεταφέρει τον cursor στην αρχή της πρώτης γραμμής που εμφανίζεται υπάρχει στην σελίδα	(n) <b>cw</b> αλλάζει τους χαρακτήρες μέχρι το τέλος των n λέξεων
<b>M</b> μεταφέρει τον cursor στην αρχή της μεσαίας γραμμής που εμφανίζεται υπάρχει στην σελίδα	<b>C</b> αλλάζει το κείμενο που απομένει μέχρι το τέλος της γραμμής
<b>L</b> μεταφέρει τον cursor στην αρχή της τελευταίας γραμμής που εμφανίζεται υπάρχει στην σελίδα	<b>~</b> αλλάζει τα κεφαλαία γράμματα σε μικρά και αντίστροφα
<b>G</b> μεταφέρει τον cursor στην αρχή της τελευταίας γραμμής του αρχείου	<b>J</b> ενώνει την γραμμή που βρίσκεται ο cursor με την αμέσως επόμενη
(n) <b>G</b> μεταφέρει τον cursor στην αρχή της n γραμμής του αρχείου	(n) <b>yy</b> αντιγράφει τις n γραμμές
<b>0</b> μεταφέρει τον cursor στην αρχή της γραμμής που βρίσκεται	<b>p</b> γράφει τις γραμμές που έχουν αντιγραφεί ή σβηστεί μετά τον cursor
<b>\$</b> μεταφέρει τον cursor στο τέλος της γραμμής που βρίσκεται	<b>P</b> γράφει τις γραμμές που έχουν αντιγραφεί ή σβηστεί πριν τον cursor
(n) <b>w</b> μπροστά (n) λέξεις	Εντολές διαχείρισης αρχείων
(n) <b>b</b> πίσω (n) λέξεις	<b>:w (file)</b> σώνει το κείμενο στο αρχείο
Είσοδος κειμένου	<b>:wq</b> σώνει το κείμενο και κλείνει ο editor
<b>i</b> είσοδος κειμένου πριν τον cursor	<b>:w! (file)</b> σώνει το κείμενο σε αρχείο που υπάρχει
<b>a</b> είσοδος κειμένου αμέσως μετά τον cursor	<b>:q</b> κλείνει ο editor με/χωρίς τις αλλαγές που έγιναν
<b>I</b> είσοδος κειμένου στην αρχή της γραμμής	<b>:q!</b> κλείνει ο editor χωρίς να κρατήσει τις αλλαγές που έγιναν
<b>A</b> είσοδος κειμένου στο τέλος της γραμμής	<b>:n</b> ανοίγει το επόμενο αρχείο στην λίστα
<b>r</b> αντικατάσταση του χαρακτήρα πάνω στον οποίο βρίσκεται ο cursor με τον επόμενο χαρακτήρα που εισάγεται	<b>:f (file)</b> αλλάζει το όνομα του αρχείου σε <b>file</b>
<b>o</b> δημιουργία καινούριας γραμμής αμέσως μετά τη γραμμή που βρίσκεται ο cursor	<b>:r (file)</b> ανοίγει το αρχείο <b>file</b> και γράφει τα περιεχόμενά του αμέσως μετά τον cursor
<b>O</b> δημιουργία καινούριας γραμμής αμέσως πριν τη γραμμή που βρίσκεται ο cursor	<b>ZZ</b> αποθηκεύει τις αλλαγές που έχουν γίνει και κλείνει ο editor

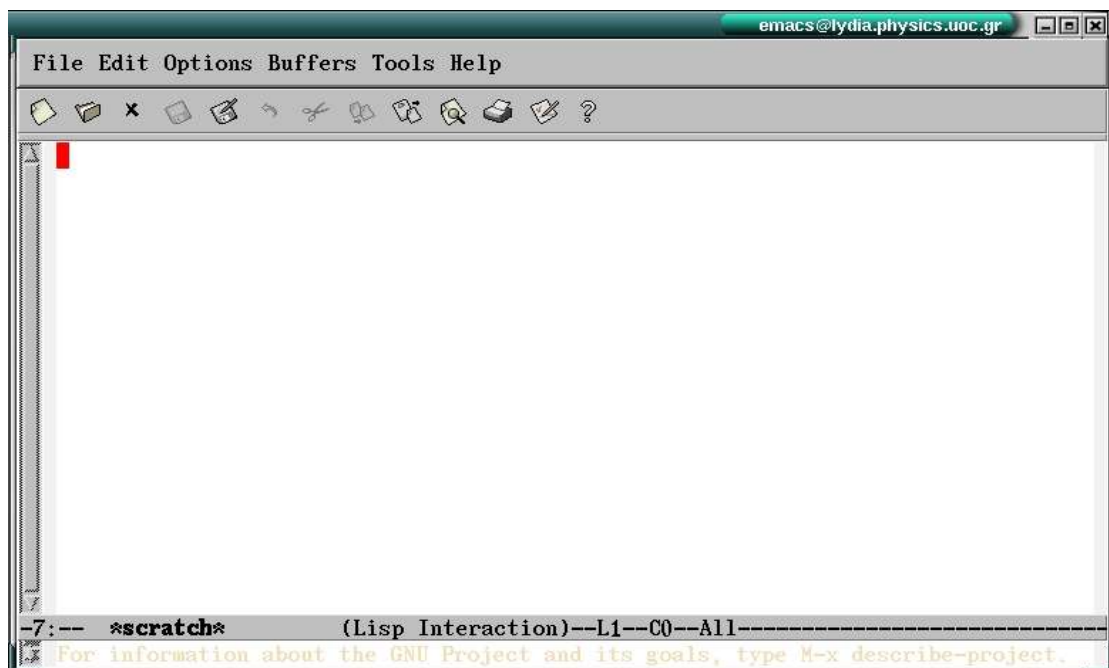
### Ο επεξεργαστής κειμένου emacs

Ο emacs εμφανίστηκε αργότερα από τον vi σε μια προσπάθεια να δημιουργηθεί ένας πιο φιλικός, για τον χρήστη, κειμενογράφος. Ο emacs στην μη γραφική του μορφή φαίνεται όπως στο ακόλουθο σχήμα:



Σχήμα 6 Ο emacs σε text mode

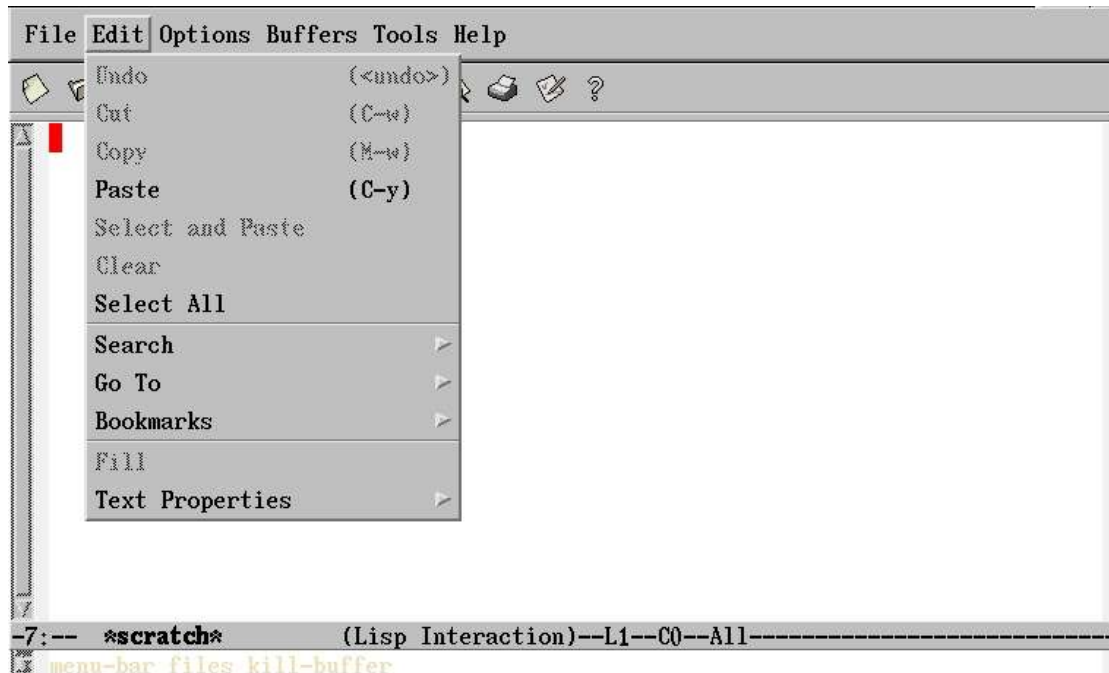
Στην γραφική του μορφή είναι όπως φαίνεται παρακάτω:



Σχήμα 7 Ο emacs σε γραφικό περιβάλλον

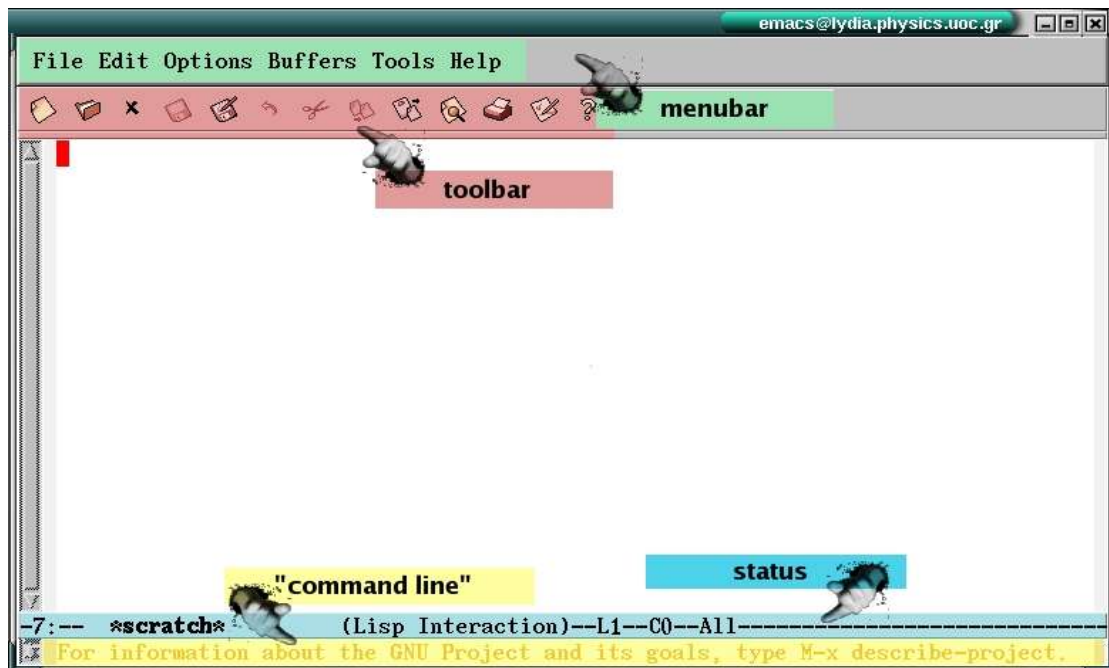
Αυτό που διακρίνει κανείς άμεσα είναι μια πλήρως λειτουργική toolbar στην κορυφή της οθόνης. Με τη χρήση του mouse, ο χρήστης έχει πρόσβαση στα menu και στα κουμπιά με τις

προγραμματισμένες λειτουργίες.



Σχήμα 8 Το menu "Edit" του emacs

Μια σύντομη παρουσίαση του γραφικού interface του emacs φαίνεται στο παρακάτω σχήμα:



Σχήμα 9 Επεξήγηση του interface του emacs

## Σύντομος οδηγός χρήσης του emacs

Το σύμβολο **C-** αντιπροσωπεύει το **Control**Το σύμβολο **M-** αντιπροσωπεύει το **Meta (Alt)**

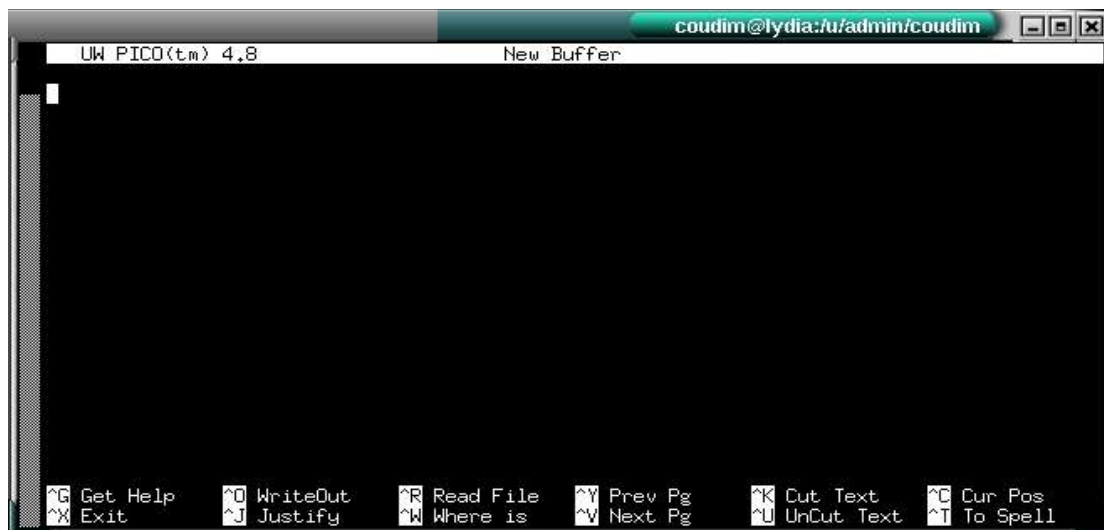
## Άλλες σημαντικές λειτουργίες

Βασικές εντολές			
C-h	online βοήθεια	M-(n)	επανάληψη της επόμενης εντολής n φορές
C-x u	ακύρωση τελευταίας ενέργειας (undo)	C-d	σβήσιμο ενός χαρακτήρα
C-x C-g	έξοδος από την τρέχουσα λειτουργία ή εντολή	M-d	σβήσιμο λέξης
C-x C-s	σώσιμο αρχείου	C-k	διαγραφή γραμμής
C-x C-c	έξοδος από τον emacs	M-k	διαγραφή πρότασης
Εντολές κίνησης του cursor		C-s	αναζήτηση προς τα εμπρός
C-f	κίνηση εμπρός ένα χαρακτήρα	C-r	αναζήτηση προς τα πίσω
C-b	κίνηση πίσω για ένα χαρακτήρα	M-u	αλλαγή της λέξης σε κεφαλαία
C-p	μετακίνηση στην προηγούμενη γραμμή	M-l	αλλαγή της λέξης σε μικρά
C-n	μετακίνηση στην επόμενη γραμμή	C-@	ορίζει την αρχή μιας περιοχής
C-a	μετακίνηση στην αρχή της γραμμής	C-w	αποθηκεύει και κατόπιν σβήνει το κείμενο από την αρχή της περιοχής και μετά
C-e	μετακίνηση στο τέλος της γραμμής	C-y	αντιγράφει το κείμενο που είχε αποθηκευτεί
C-l	μετακίνηση στο κέντρο της γραμμής	M-w	αντιγράφει την περιοχή που έχει επιλεγεί
C-v	μετακίνηση στην επόμενη σελίδα	C-x C-f	αναζήτηση αρχείου και ανάγνωσή του
M-v	μετακίνηση στην προηγούμενη σελίδα	C-x i	προσθέτει τα περιεχόμενα αρχείου στην θέση που βρίσκεται ο cursor
M-f	μετακίνηση μια λέξη μετά	C-x C-s	σώνει το αρχείο
M-b	μετακίνηση μια λέξη πίσω	C-x C-w	αποθηκεύει τις αλλαγές σε αρχείο με διαφορετικό όνομα
M-a	μετακίνηση στην αρχή της πρότασης	C-x C-c	κλείνει ο emacs ρωτώντας τον χρήστη αν θέλει να σώσει το κείμενο ή όχι
M-e	μετακίνηση στο τέλος της πρότασης		
M-{	μετακίνηση στην αρχή της παραγράφου		
M-}	μετακίνηση στο τέλος της παραγράφου		

### Ο επεξεργαστής κειμένου pico

Ο pico είναι ένας από τους πιο απλούς text editors που υπάρχουν στο UNIX. Αρκετά πιο απλός στην λειτουργία του και από τον vi αλλά και από τον emacs, υστερεί ωστόσο σε δυνατότητες. Για την συγγραφή απλών κειμένων με χαμηλό βαθμό δυσκολίας θεωρείται κατάλληλος.

Ξεκινώντας τον pico αυτό που περιμένει κανείς να δει είναι κάτι σαν και αυτό:



Σχήμα 10 Ο text editor pico

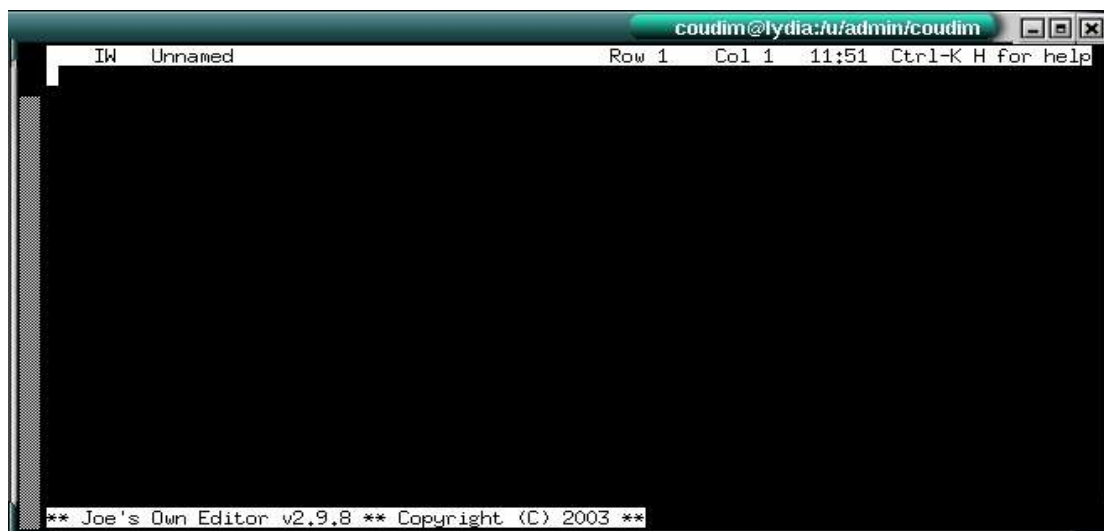
Στο κάτω μέρος της οθόνης μπορεί κανείς να διακρίνει ένα σύντομο οδηγό χρήσης με τις βασικές λειτουργίες και την συντομεύσεις του πληκτρολογίου που μπορεί να χρησιμοποιήσει.

#### Σύντομος οδηγός χρήσης του pico

Γενικές εντολές		Εντολές μετακίνησης	
<b>^G</b>	on-line βοήθεια	<b>^F</b> ή <b>►</b>	κίνηση του cursor προς τα δεξιά
<b>^O</b>	αποθήκευση του κειμένου σε αρχείο	<b>^B</b> ή <b>◄</b>	κίνηση του cursor προς τα αριστερά
<b>^X</b>	αποθήκευση και έξοδος από τον editor	<b>^P</b> ή <b>▲</b>	κίνηση του cursor προς τα πάνω
<b>^T</b>	έλεγχος ορθογραφίας	<b>^N</b> ή <b>▼</b>	κίνηση του cursor προς τα κάτω
<b>^J</b>	στοιχειοθέτηση κειμένου	<b>^A</b>	μετακίνηση στην αρχή της γραμμής
<b>Εντολές αναζήτησης</b>		<b>^E</b>	μετακίνηση στο τέλος της γραμμής
<b>^W</b>	αναζήτηση	<b>^Y</b>	εμφάνιση της προηγούμενης σελίδας
<b>Εισαγωγή, σβήσιμο, αλλαγή κειμένου</b>		<b>^V</b>	εμφάνιση της επόμενης σελίδας
<b>^K</b>	αποκόλληση μιας γραμμής του κειμένου	<b>^C</b>	εμφάνιση της θέσης του cursor
<b>^U</b>	επικόλληση μιας γραμμής του κειμένου		
<b>^R</b>	επικόλληση των περιεχομένων ενός άλλου αρχείου στην αμέσως μετά τη θέση που βρίσκεται ο cursor		
<b>^^</b>	επιλογή μιας περιοχής κειμένου		

### Ο επεξεργαστής κειμένου joe

Ο joe είναι ένας πολύ απλός text editor όπως και ο pico αλλά με μερικές επιπλέον λειτουργίες με τις οποίες όμως δεν θα ασχοληθούμε εδώ. Ξεκινώντας τον joe αυτό που περιμένει κανείς να δει είναι κάτι σαν και αυτό:



Σχήμα 11 Ο text editor joe

Στο κάτω μέρος της οθόνης μπορεί κανείς να διακρίνει ένα σύντομο οδηγό χρήσης με τις βασικές λειτουργίες και την συντομεύσεις του πληκτρολογίου που μπορεί να χρησιμοποιήσει.

#### Σύντομος οδηγός χρήσης του joe

Γενικές εντολές		Εντολές μετακίνησης	
<b>^Kd</b>	αποθήκευση αρχείου	<b>^F ή ►</b>	κίνηση του cursor προς τα δεξιά
<b>^Kx</b>	αποθήκευση αρχείου και έξοδος	<b>^B ή ◀</b>	κίνηση του cursor προς τα αριστερά
<b>^C</b>	έξοδος χωρίς αποθήκευση αλλαγών	<b>^P ή ▲</b>	κίνηση του cursor προς τα πάνω
<b>^Kh</b>	on-line βοήθεια	<b>^N ή ▼</b>	κίνηση του cursor προς τα κάτω
<b>^L</b>	επανασχεδιασμός (redraw) της οθόνης	<b>^A</b>	μετακίνηση στην αρχή της γραμμής
Εντολές αναζήτησης		<b>^E</b>	μετακίνηση στο τέλος της γραμμής
<b>^K^F</b>	αναζήτηση κειμένου	<b>^U</b>	εμφάνιση της προηγούμενης σελίδας
Εισαγωγή, σβήσιμο, αλλαγή κειμένου		<b>^V</b>	εμφάνιση της επόμενης σελίδας
<b>^Kr</b>	εισαγωγή κειμένου από άλλο αρχείο	<b>^Ku</b>	μετακίνηση στην αρχή του κειμένου
<b>^Ke</b>	άνοιγμα νέου αρχείου	<b>^Kv</b>	μετακίνηση στο τέλος του κειμένου
<b>^K-</b>	ακύρωση διαγραφής λέξης	<b>^Kl #</b>	μετακίνηση στην γραμμή με αριθμό #

## Σύνοψη εντολών του UNIX ανά κατηγορία

Διαχείριση directories		Γενικές εντολές	
<b>cd</b>	αλλαγή directory	<b>date</b>	εμφανίζει την ημερομηνία
<b>mkdir</b>	δημιουργία directory	<b>cal</b>	εμφανίζει το ημερολόγιο
<b>rmdir</b>	διαγραφή directories	<b>df</b>	εμφανίζει στατιστικά χρήσης των filesystems
<b>pwd</b>	εμφάνιση του τρέχοντος directory	<b>du</b>	εμφανίζει το χώρο που καταλαμβάνουν αρχεία ή directories
<b>ls</b>	εμφάνιση των περιεχομένων ενός directory	<b>quota</b>	εμφανίζει το όριο χρήσης του δίσκου για τους χρήστες
Διαχείριση αρχείων και directories		<b>hostname</b>	εμφανίζει το όνομα του υπολογιστή
<b>cp</b>	αντιγραφή αρχείων ή directories	<b>man</b>	εμφανίζει τα manual pages
<b>mv</b>	μετακίνηση αρχείων ή directories	<b>which</b>	δείχνει το full path μιας εντολής
<b>rm</b>	διαγραφή αρχείων ή directories	<b>passwd</b>	αλλάζει το passwd
<b>chmod</b>	αλλαγή permissions αρχείων ή directories	<b>ps</b>	δείχνει τις διεργασίες του/των χρηστών
Προβολή αρχείων		<b>kill</b>	τερματίζει διεργασίες
<b>cat</b>	εμφάνιση των περιεχομένων ενός αρχείου	<b>who</b>	δείχνει τους χρήστες που είναι συνδεδεμένοι με τον υπολογιστή
<b>echo</b>	εμφανίζει το κείμενο που δίνεται σαν παράμετρος στην οθόνη	<b>last</b>	δείχνει τους χρήστες που είχαν/έχουν κάνει login στον υπολογιστή
<b>head</b>	εμφανίζει τις πρώτες <b>N</b> γραμμές	<b>grep</b>	αναζήτηση λέξεων ή χαρακτήρων από αρχείο
<b>tail</b>	εμφανίζει τις τελευταίες <b>N</b> γραμμές	<b>find</b>	αναζήτηση αρχείων ή directories
<b>more</b>	εμφάνιση των περιεχομένων ανά σελίδα	<b>bc</b>	ένας text-based υπολογιστής "τσέπης"
Εντολές διαχείρισης εκτυπώσεων		Εντολές σύνδεσης με άλλους υπολογιστές	
<b>lpr</b>	εκτυπώνει ένα ή περισσότερα αρχεία	<b>ssh</b>	Οι εντολές αυτές, επιτρέπουν την σύνδεση, το άνοιγμα ενός καινούριου shell ή την εκτέλεση εντολών, σε διαφορετικό υπολογιστή από αυτόν που έχει ήδη κάνει login ο χρήστης. Για περισσότερες πληροφορίες, πρέπει κανείς να ανατρέξει στα αντίστοιχα manual pages.
<b>lpq</b>	δείχνει την κατάσταση ενός εκτυπωτή	<b>telnet</b>	
<b>lpstat</b>	δείχνει την κατάσταση των εκτυπωτών	<b>rlogin</b>	
<b>lprm</b>	ακυρώνει τις προς εκτύπωση εργασίες	<b>rsh</b>	

## ΠΡΟΓΡΑΜΜΑΤΑ ΔΙΑΧΕΙΡΙΣΗΣ ΗΛΕΚΤΡΟΝΙΚΗΣ ΑΛΛΗΛΟΓΡΑΦΙΑΣ (E-MAIL CLIENTS) ΣΤΟ UNIX

Υπάρχει μια ποικιλία από προγράμματα διαχείρισης αλληλογραφίας στο UNIX. Μπορεί κανείς να συναντήσει προγράμματα με πλήρες γραφικό περιβάλλον αλλά και πιο απλούς text-based e-mail clients.

Θα προχωρήσουμε σε μια περιγραφή μερικών από τους πιο απλούς και δημοφιλείς text-based e-mail clients στο UNIX. Το βασικό πλεονέκτημα που έχουν σε σχέση με άλλους γραφικούς e-mail clients είναι ότι είναι γρήγοροι και ιδιαίτερα απλοί στην λειτουργία τους, χωρίς να υστερούν σε δυνατότητες τουλάχιστον στα βασικές απαιτήσεις που μπορεί να έχει κανείς για την αποστολή ενός απλού e-mail.

### Ο πιο απλός e-mail client του UNIX, το *mail*

Ο πιο παραδοσιακός και απλός στην χρήση του e-mail client είναι το **mail**. Για να στείλει κανείς e-mail με την εντολή **mail** δεν έχει παρά να καλέσει την εντολή **mail**, δίνοντας σαν argument την e-mail διεύθυνση του παραλήπτη. Θα δούμε την λειτουργία της εντολής **mail** μέσα από ένα συγκεκριμένο παράδειγμα. Όπου εμφανίζονται μέσα στο παράδειγμα έντονοι χαρακτήρες είναι η απάντηση του συστήματος και όπου εμφανίζεται ο χαρακτήρας ↵ δηλώνει το πλήκτρο **Enter**.

```
# mail mhjohn@univ.edu
```

Αυτό που θα γίνει στη συνέχεια, είναι ότι το πρόγραμμα **mail**, θα ζητήσει από το *Subject* του μηνύματος που πρόκειται να αποσταλεί και στη συνέχεια θα αφήσει τον χρήστη να γράψει το περιεχόμενο του e-mail.

```
# mail mhjohn@univ.edu
Subject: this is a test e-mail ...
```

```
This is a test, please ignore ...
```

Αφού γράψει και τον τελευταίο χαρακτήρα του μηνυμάτος του, τότε για να ειδοποιήσει το **mail** ότι το μήνυμα είναι έτοιμο για αποστολή, πρέπει να αλλάξει γραμμή (πατώντας το πλήκτρο **Enter**) και να πατήσει τον χαρακτήρα **τελεία** (•) ή τον συνδιασμό πλήκτρων που δηλώνει το **End Of File (EOF)** στο UNIX, κάτι που συνήθως είναι το **^D**:

```
# mail mhjohn@univ.edu ↵
Subject: this is a test e-mail ...↵
This is a test, please ignore ... ↵
.↵
```

Το επόμενο πράγμα που ζητάει το **mail**, είναι αν θέλει ο αποστολέας να στείλει το e-mail και σε κάποιον άλλο παραλήπτη. Αυτό δηλώνεται από την εντολή **mail** ζητώντας **Cc:** που σημαίνει **carbon copy** (αντίγραφο). Για το συγκεκριμένο παράδειγμα θεωρούμε ότι το e-mail πρέπει να το παραλάβει και ο χρήστης *mary* στο domain *univ-2.edu*.

```
# mail mhjohn@univ.edu ↵
Subject: this is a test e-mail ...↵
This is a test, please ignore ... ↵
.↵
Cc: mary@univ-2.edu ↵
```

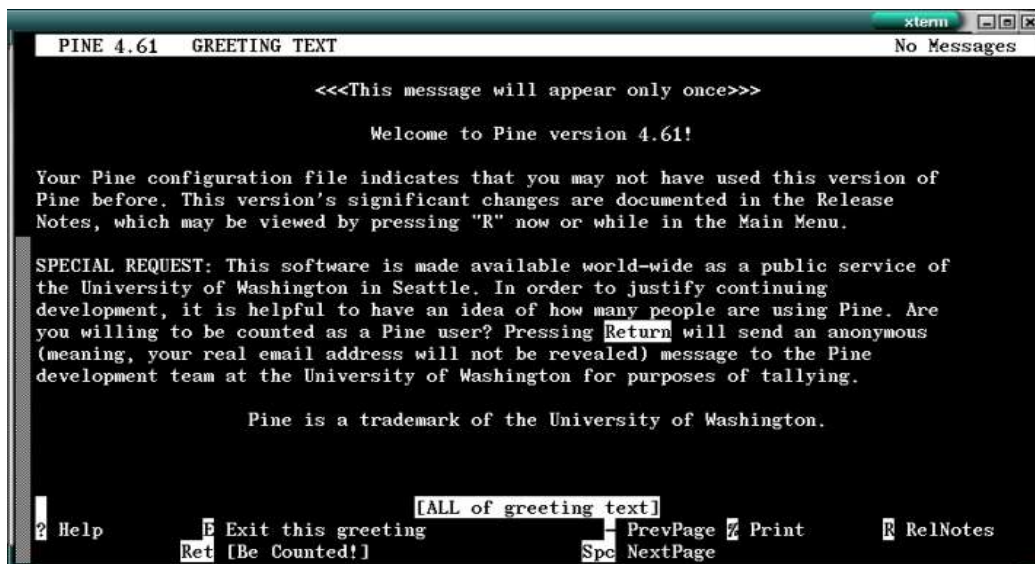
Το αν θα δώσει κάποιος **Subject:** ή **Cc:** είναι προαιρετικό.



### Ο e-mail client, *pine*

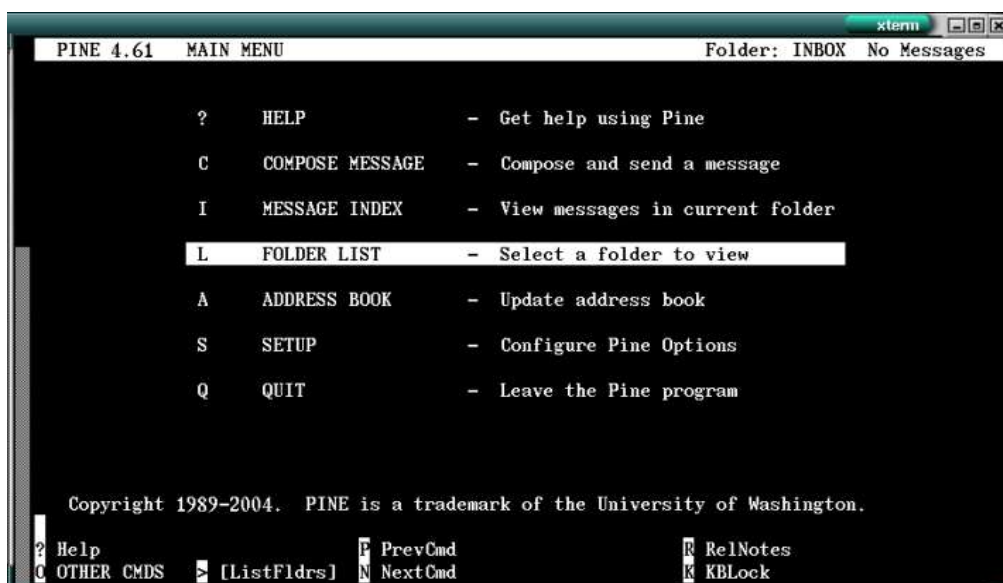
Ένας από τους πιο δημοφιλείς e-mail clients του UNIX στην κατηγορία που συζητάμε, είναι το **pine**. Πρόκειται για ένα πιο “δυνατό” client με περισσότερες δυνατότητες και ευκολίες από το **mail**.

Για να χρησιμοποιήσει κανείς το **pine** πρέπει να το καλέσει με την εντολή **pine**. Την πρώτη φορά που θα χρησιμοποιήσει κάποιος το **pine** θα δει ένα χαιρετισμό όπως αυτό που φαίνεται στην παρακάτω εικόνα:



Σχήμα 12 Η οθόνη χαιρετισμού του *pine*

Στο κάτω μέρος της οθόνης μπορεί κανείς να διακρίνει δύο σειρές με τις βασικές συντομεύσεις για τον χειρισμό του **pine**. Σε κάθε μια από τις οθόνες που εμφανίζονται στο **pine**, υπάρχουν πάντα στο κάτω μέρος οι αντίστοιχες συντομεύσεις για τις διάφορες διεργασίες που μπορούν να τελεστούν. Πατώντας “E” (όπως φαίνεται και από την εικόνα), επιλέγουμε έξοδο από το μήνυμα χαιρετισμού και βλέπουμε το κεντρικό menu του **pine**:



Σχήμα 13 Το βασικό menu του *pine*

Σε κάθε επιλογή που υπάρχει μέσα στο menu υπάρχει στα δεξιά και η αντίστοιχη εξήγηση. Στο κάτω μέρος της οθόνης εμφανίζονται και οι σχετικές συντομεύσεις του πληκτρολογίου που σχετίζονται με αυτή την οθόνη.

Η επιλογή *Folder List* εμφανίζει τους φακέλους στους οποίους είναι αποθηκευμένα τα e-mails που έχουν αποσταλεί και τα e-mails που έχουν σωθεί από τον χρήστη:



Ο φάκελος *INBOX* αφορά τα νέα μηνύματα που υπάρχουν στο e-mail του χρήστη, ο φάκελος *sent-mail* e-mails που έχουν αποσταλεί και ο φάκελος *saved-messages* μηνύματα που έχουν σωθεί. Σε περίπτωση που κανείς θέλει να σώσει σε διαφορετικούς φακέλους τα διάφορα e-mails που λαμβάνει, μπορεί να δημιουργήσει ξεχωριστούς φακέλους όπως φαίνεται και στο παράδειγμα (οι φάκελοι *john* και *mary*). Για να γυρίσει κανείς στην αρχική οθόνη του **pine** πρέπει να χρησιμοποιήσει το πλήκτρο "<".

Για να εμφανιστούν τα νέα μηνύματα, μπορεί κανείς να επιλέξει το *INBOX* μέσα από το *Folder List* ή το *MESSAGE INDEX* από το αρχικό menu. Αυτό που θα εμφανιστεί, είναι κάτι σαν το ακόλουθο:



Για το κάθε μήνυμα που υπάρχει στο *INBOX* εμφανίζεται το *Subject* (αν έχει), ο αποστολέας,

ο αύξων αριθμός του μηνύματος και στα αριστερά η ένδειξη "N" αν πρόκειται για νέο μήνυμα που δεν έχει διαβαστεί ακόμα. Για να εμφανιστεί το περιεχόμενο ενός μηνύματος, αρκεί να κινηθεί κανείς με τα βελάκια του πληκτρολογίου και να πατήσει *Enter*. Η οθόνη που εμφανίζεται είναι η εξής:



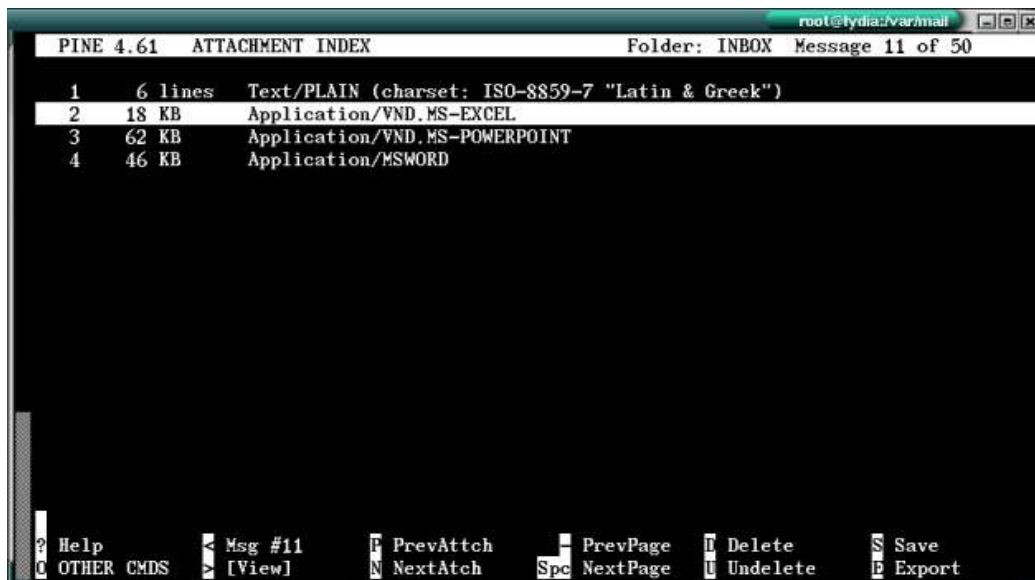
Σχήμα 15 Εμφάνιση ενός e-mail στο pine

Εμφανίζονται διάφορα τμήματα από την επικεφαλίδα του e-mail (ημερομηνία άφιξης, αποστολέας, το θέμα κλπ), το περιεχόμενο του e-mail και στο κάτω μέρος της οθόνης όπως πάντα οι συντομώσεις του πληκτρολογίου για τον χειρισμό του **pine**. Σε περίπτωση που το e-mail περιείχε κάποιο attachment, η συντόμωση του **pine** για να εμφανιστεί είναι το πλήκτρο ">". Με τα πλήκτρα "P" (*previous*) και "N" (*next*) μπορούμε να εμφανίσουμε το προηγούμενο και το επόμενο e-mail αντίστοιχα, με το *space* την επόμενη σελίδα του e-mail αν δεν εμφανίζεται όλο το μήνυμα στην οθόνη, και τέλος με το "D" (*delete*) να το σβήσουμε, με το "R" (*reply*) να απαντήσουμε και με το "F" (*forward*) να το μεταβιβάσουμε σε άλλο χρήστη.

Ένα e-mail με attachment θα εμφανιζόταν ως εξής:

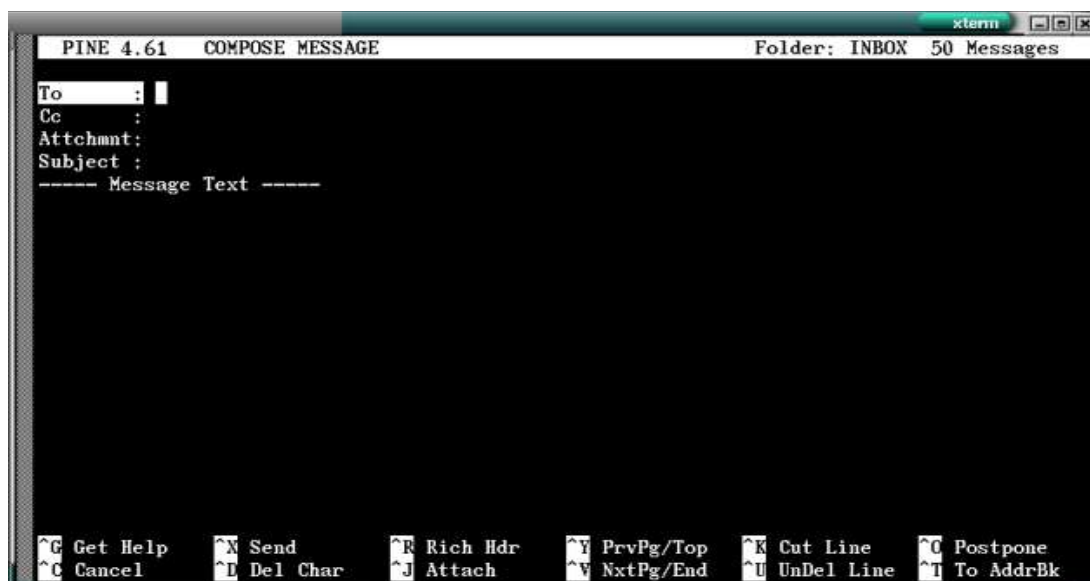


Οι πρόσθετες πληροφορίες που υπάρχουν εδώ, είναι ότι αμέσως κάτω από την επικεφαλίδα του e-mail υπάρχουν πληροφορίες σχετικά με τα attachments που περιέχει το e-mail. Με το πλήκτρο ">" εμφανίζεται η οθόνη για τον χειρισμό των attachments:



Ισχύουν και εδώ τα βασικά πλήκτρα "P", "N" και *space*, αλλά για τα attachments αυτή τη φορά. Υπάρχουν και μερικά επιπλέον για άλλες λειτουργίες (αποθήκευση κλπ). Επιλέγοντας με τα βελάκια και πατώντας *Enter* εμφανίζεται το συγκεκριμένο attachment. Εδώ πρέπει να σημειωθεί, ότι προκειμένου να εμφανιστεί κάποιο attachment, πρέπει να υπάρχει εγκατεστημένο στο σύστημα και το αντίστοιχο πρόγραμμα που μπορεί να το χειριστεί, για παράδειγμα, μια εικόνα προκειμένου να προβληθεί, χρειάζεται ένα πρόγραμμα προβολής-επεξεργασίας εικόνας κλπ. Με το πλήκτρο "<" επιστρέφουμε πάλι στο μήνυμα του οποίου προβάλαμε τα attachments και με "<" άλλη μια φορά στην λίστα με τα μηνύματα του *INBOX*.

Για την δημιουργία ενός καινούριου μηνύματος, πρέπει να πάει κανείς στην επιλογή *COMPOSE MESSAGE* από την αρχική οθόνη του **pine**:

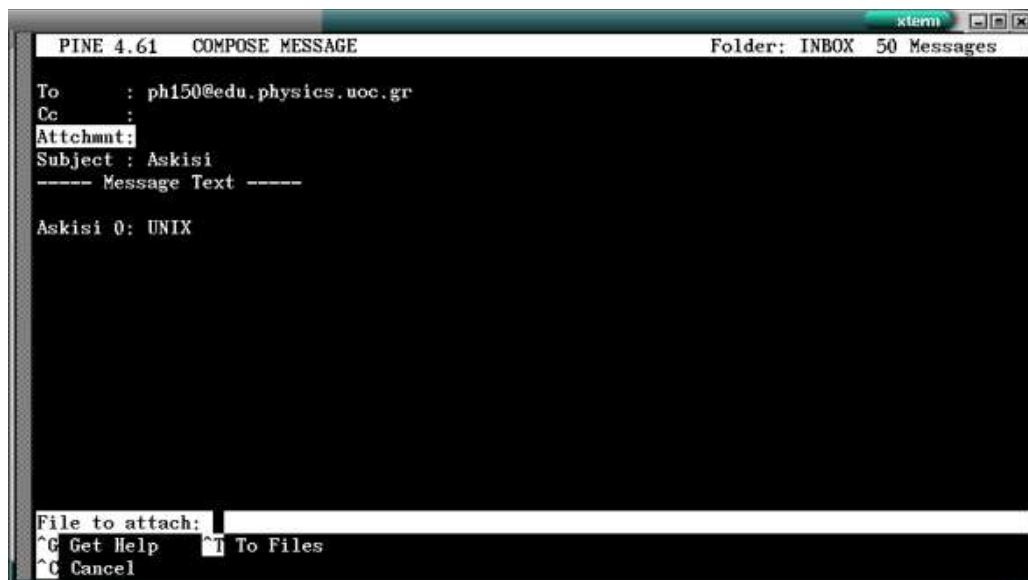


Στην οθόνη αυτή διακρίνονται τα πεδία *To:*, *Cc:*, *Attachement:* και *Subject:* που έχουν την ίδια

σημασία όπως και στους υπόλοιπους e-mail clients. Το περιεχόμενο του μηνύματος γράφεται αμέσως κάτω από το σημείο ----- *Message Text* ----- . Για να φτάσει κανείς εκεί, πρέπει να συμπληρώσει τα πεδία που τον ενδιαφέρουν (*Το:* και προαιρετικά κάποιο από τα υπόλοιπα) ή να κινηθεί προς το κάτω μέρος της οθόνης με τα βελάκια του πληκτρολογίου. Στο κάτω μέρος της οθόνης, εμφανίζονται όπως πάντα οι συντομεύσεις του πληκτρολογίου. Ένα συμπληρωμένο e-mail θα έμοιαζε κάπως έτσι:



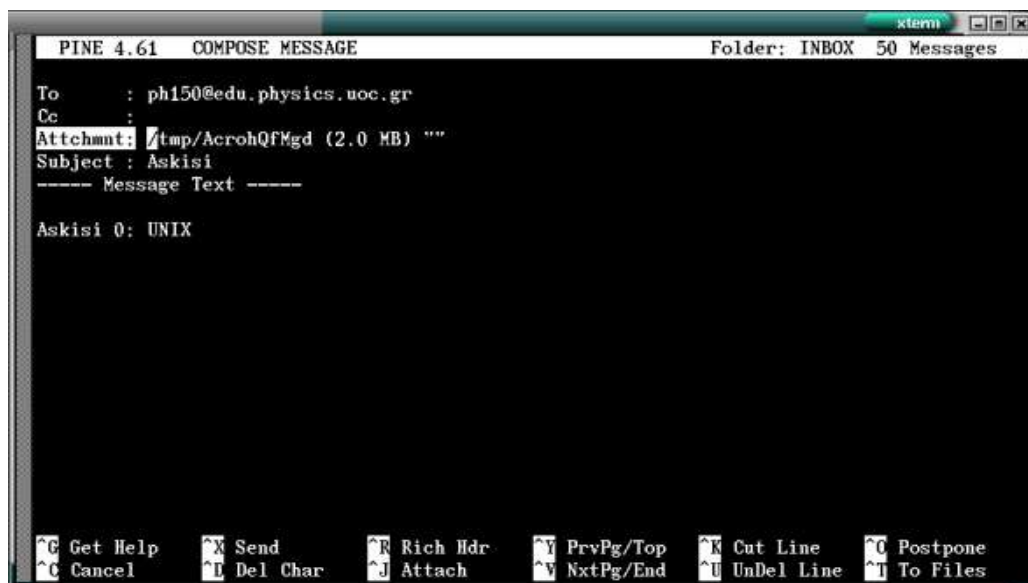
Για να προσθέσουμε κάποιο attachment, πρέπει να κινηθούμε στο πεδίο *Attachmnt* στην οθόνη αυτή και να πατήσουμε το πλήκτρο **^J** (Control-J). Η οθόνη αλλάζει λίγο και εμφανίζει στο κάτω μέρος τις λειτουργίες για την είσοδο του attachment:



Με το πλήκτρο **^T** εμφανίζεται η λίστα με τα αρχεία και τα directories που βρίσκονται κάτω από το **home directory** για να γίνει η επιλογή του αρχείου που θα μπει σαν attachment στο μήνυμα:



Και σε αυτή την οθόνη η κίνηση μέσα στον κατάλογο γίνεται με τα βελάκια και η επιλογή κάποιου αρχείου με το πλήκτρο *Enter* ή το "S":



Το πλήρες path του αρχείου που γίνεται attachment στο e-mail εμφανίζεται δίπλα από το πεδίο *Attachmnt*. Με την ίδια διαδικασία μπορούν να προστεθούν και επιπλέον αρχεία.

Η αποστολή του e-mail γίνεται με το πλήκτρο **^X**. Το **pine** ζητάει επιβεβαίωση για την αποστολή του e-mail και κατόπιν το στέλνει.



## ΤΑ "ΚΡΥΦΑ" ΑΡΧΕΙΑ ΣΤΟ UNIX ΚΑΙ Η ΧΡΗΣΙΜΟΤΗΤΑ ΤΟΥΣ

Με τον όρο "κρυφά" αρχεία στο UNIX, εννοούμε τα αρχεία που το όνομά τους ξεκινάει με τον χαρακτήρα **τελεία (•)**. Η πλειοψηφία των εφαρμογών του UNIX χρησιμοποιούν τέτοιου είδους αρχεία για να αποθηκεύουν τις προσωπικές προτιμήσεις των χρηστών, Καθώς επίσης και τον τρόπο λειτουργίας που επιθυμεί για αυτά ο κάθε χρήστης. Σε ένα shell για παράδειγμα, μπορεί να ορίσει κανείς aliases ώστε να απλοποιήσει την χρήση κάποιων εντολών. Επίσης μπορεί να χρειάζεται κάποιο διαφορετικό *prompt* από το προκαθορισμένο του συστήματος οπότε πρέπει να μπορεί να ορίσει ένα διαφορετικό που να ισχύει κάθε φορά που κάνει login στο σύστημα. Σε μερικές μάλιστα περιπτώσεις, καθαρά για λόγους διαχείρισης από πλευράς της εφαρμογής, απαιτούνται περισσότερα από ένα αρχεία για να αποθηκευτούν οι ρυθμίσεις και τα δεδομένα του προγράμματος για κάποιον συγκεκριμένο χρήστη. Στην περίπτωση αυτή μια συγκεκριμένη τακτική είναι να υπάρχει ένα "κρυφό" directory με αρχεία ή και άλλα directories μέσα για την αποθήκευση όλων αυτών των δεδομένων για κάθε χρήστη.

Σε γενικές γραμμές αυτά τα "κρυφά" αρχεία ή directories βρίσκονται αμέσως κάτω από το **home directory** του κάθε χρήστη. Μερικά από τα πιο συνηθισμένα είναι τα "κρυφά" αρχεία που αφορούν τα shells του συστήματος. Κατά κανόνα, μέσα σε αυτά ορίζονται διάφορα aliases, το prompt, το path και μερικές άλλες environment variables που μπορεί να είναι χρήσιμες για τον χρήστη.

## Τα "κρυφά" αρχεία των shells

Ειδικά στην περίπτωση των shells, επειδή τα αρχεία αυτά διαβάζονται μια μόνο φορά κατά την εκκίνηση κάποιου shell, πολλές φορές αναφέρονται και σαν start-up shell files (αρχεία εκκίνησης του shell). Για το κάθε shell αυτά τα αρχεία έχουν διαφορετικό όνομα που συνήθως σχετίζεται με το όνομα του shell και καταλήγει με τους χαρακτήρες **rc**.

Μερικά από τα πιο συνηθισμένα shells και τα start-up αρχεία τους

```
tcsh .tcshrc
csh .cshrc
bash .bashrc
sh .profile
```