

ΠΑΝΕΠΙΣΤΗΜΙΟ



ΠΕΛΟΠΟΝΝΗΣΟΥ

Σημειώσεις για το
λειτουργικό σύστημα UNIX

Κ. Βασιλάκης, Α. Σωτηροπούλου, Α. Καλόξυλος

| | |
|---|-----------|
| 1 Εισαγωγή στο UNIX..... | 3 |
| 1.1 Το Κέλυφος (shell) | 3 |
| 1.2 Ο πυρήνας του UNIX..... | 4 |
| 1.3 Διεργασίες (processes) | 4 |
| 1.4 Το σύστημα αρχείων του UNIX | 5 |
| 2. Βασικές εντολές UNIX..... | 8 |
| 2.1 Δουλεύοντας με αρχεία και καταλόγους..... | 8 |
| 2.2 Ανακατευθύνσεις & διαυλοι | 11 |
| 2.3 Μεταχαρακτήρες..... | 12 |
| 2.4 Χρήσιμες εντολές..... | 13 |
| 2.4.1 Αλλαγή προνομίων προσπέλασης σε αρχεία..... | 13 |
| 2.4.2 Σύγκριση αρχείων..... | 14 |
| 2.4.3 Υπηρεσίες πληροφόρησης & βασικών πράξεων με αρχεία..... | 14 |
| 2.4.4 Μηχανισμός History | 16 |
| 2.4.5 Προστασία ειδικών χαρακτήρων..... | 17 |
| 2.4.6 Εκτέλεση διεργασιών στο shell..... | 17 |
| 3. Ο κειμενογράφος πλήρους οθόνης vi..... | 18 |
| 3.1 Περιβάλλον λειτουργίας του vi..... | 18 |
| 3.2 Εντολές εισαγωγής κειμένου | 20 |
| 3.3 Εντολές Διαγραφής κειμένου..... | 20 |
| 3.4 Αποθήκευση κειμένου και έξοδος από τον κειμενογράφο | 20 |

1 Εισαγωγή στο UNIX

1.1 Το Κέλυφος (*shell*)

Ο κύριος σκοπός του shell είναι να διαβάζει και να μεταφράζει τις εντολές μας καθώς μας συνδέει με το UNIX. Αυτό σημαίνει ότι δέχεται τις εντολές που δίνουμε στο τερματικό μας, ελέγχει τη σύνταξή τους, καλεί τις κατάλληλες εσωτερικές ή εξωτερικές εντολές του UNIX και επαναφέρει τον έλεγχο στον χρήστη όταν οι εντολές ολοκληρωθούν ορθά.

Είναι αλληλεπιδραστικό (interactive) επειδή απαντά άμεσα στις οδηγίες σας και αναφέρει λάθη και αποτελέσματα.

Το shell προσφέρει και άλλες δυνατότητες:

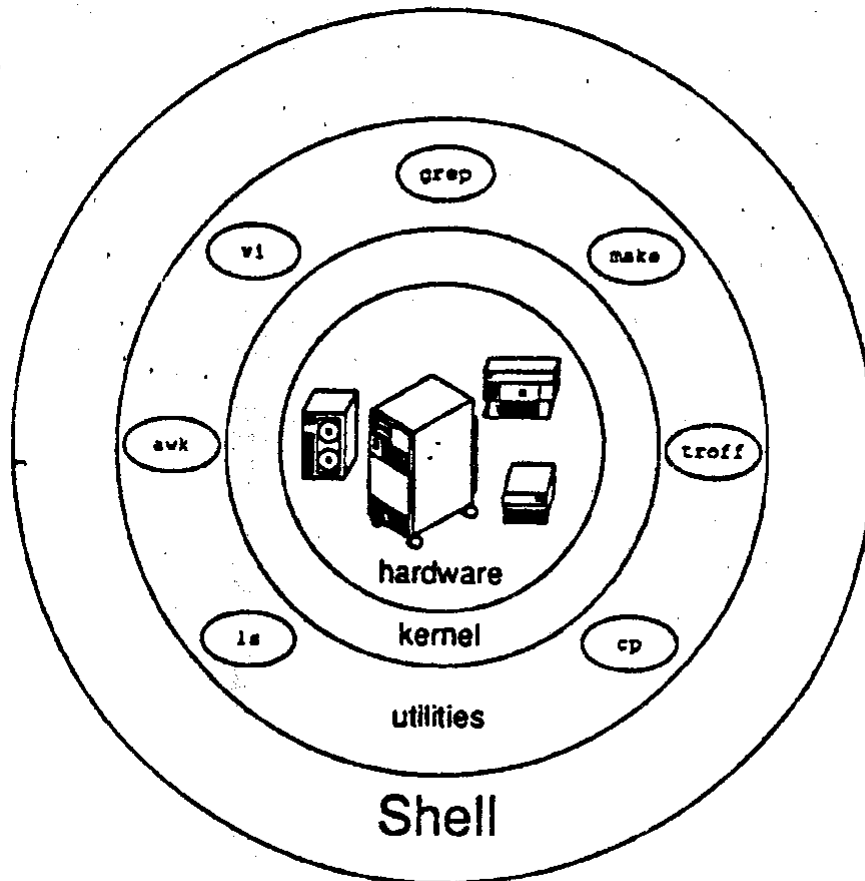
-Επιτρέπει στο χρήστη να φτιάξει το δικό του σύστημα (ή περιβάλλον) που να ταιριάζει στον τρόπο εργασίας του.

-Δίνει τη δυνατότητα για σύντμηση μεγάλων ονομάτων, εντολών και ακολουθιών εντολών .

-Σώζει τις παλιότερες εντολές ώστε να μπορούμε να τις επαναλαμβάνουμε, αναθεωρήσουμε ή να χρησιμοποιήσουμε ένα τμήμα από αυτές τις εντολές.

-Έχει την δυνατότητα διαχείρισης εργασιών (job control), ώστε να μπορούμε να δημιουργήσουμε νέες εργασίες, να τις εκτελέσουμε με ή χωρίς διαλογική επικοινωνία από το τερματικό μας και να τις τερματίσουμε.

-Μέσω του shell γράφουμε προγράμματα τα οποία ονομάζονται scripts και εκτελούνται στο επίπεδο εντολών. Αυτά μπορεί απλώς να εκτελούν εντολές σειριακά ή να χρησιμοποιούν δομές όπως επαναλήψεις (loops) και επιλογή αποφάσεων (decisions). Μπορούν να έχουν ορίσματα (arguments) και να επιτρέπουν αλληλεπιδράσεις (interactions).



1.2 Ο πυρήνας του UNIX

Ο πυρήνας (kernel) του UNIX είναι η καρδιά του λειτουργικού συστήματος. Ελέγχει την προσπέλαση στον υπολογιστή και τα αρχεία, επιμερίζει τη χρήση των τμημάτων του συστήματος μεταξύ των διαφόρων διεργασιών που γίνονται μέσα στον υπολογιστή, συντηρεί το σύστημα αρχείων και διαχειρίζεται τη μνήμη του υπολογιστή. Παρ' όλο που οι απλοί χρήστες σπάνια έχουν άμεση επαφή μαζί του, ο πυρήνας είναι το κυριότερο μέρος του λειτουργικού συστήματος.

Ο διαχειριστής του συστήματος (system administrator) εκτελεί μια διεργασία που λέγεται διαμόρφωση (configuration) του πυρήνα, για να προσαρμόσει τα λειτουργικά χαρακτηριστικά του συστήματος. Με τη διευθέτηση του πυρήνα, ρυθμίζονται οι ορισμένες εσωτερικές παράμετροι του συστήματος και δημιουργούνται οι οδηγίες (drivers), οι οποίοι ελέγχουν τις περιφερειακές μονάδες που είναι συνδεδεμένες με τον υπολογιστή. Όμως, ακόμα και ένας διαχειριστής συστήματος δεν είναι σε θέση να τροποποιήσει τη δομή του πυρήνα γιατί, για να μπορέσει να το κάνει αυτό θα πρέπει να γράψει το ίδιο το UNIX από την αρχή.

1.3 Διεργασίες (processes)

Κατά τη διάρκεια της λειτουργίας του UNIX συμβαίνουν ταυτόχρονα πολλές δραστηριότητες οι οποίες ονομάζονται διεργασίες (processes). Μπορεί κανείς να φανταστεί ότι αυτές οι διεργασίες συμβαίνουν παράλληλα, παρόλο που οι

περισσότεροι υπολογιστές με λειτουργικό σύστημα UNIX δεν υποστηρίζουν πραγματική παράλληλη επεξεργασία παρά μόνο για ορισμένες εξειδικευμένες εργασίες όπως είναι η εκτύπωση.

Κάθε στιγμή υπάρχει τουλάχιστον μια διεργασία στον πυρήνα και μία για κάθε χρήστη που έχει μπει (έχει κάνει login) στο σύστημα. Στην πραγματικότητα για τον πυρήνα «τρέχουν» πάντα πολλές διεργασίες ταυτόχρονα ενώ κάθε χρήστης μπορεί να ξεκινήσει και αυτός αρκετές διεργασίες. Με την εντολή ps (process status) μπορείτε να δείτε έναν κατάλογο με όλες τις διεργασίες που τρέχουν. Για να εκτελέσετε αυτή την εντολή πληκτρολογήστε ps -ef.

Η διαχείριση των διεργασιών γίνεται από τον πυρήνα, ο οποίος περνάει διαδοχικά από τη μία στην άλλη ανάλογα με τις προτεραιότητες και τις ανάγκες της κάθε μίας. Μια διεργασία μπορεί να δημιουργήσει άλλες διεργασίες. Αυτή ονομάζεται τότε γονική (parent) διεργασία και εκείνες που δημιουργούνται ονομάζονται θυγατρικές (child) διεργασίες. Υπάρχουν επίσης κάποιες διεργασίες που ονομάζονται «δαίμονες» (daemons), οι οποίες παραμένουν λίγο-πολύ μόνιμα στο σύστημα γιατί εκτελούν συνεχιζόμενες εργασίες (π.χ., η διαχείριση του ταχυδρομείου, προγραμματισμών δουλειών που εκτελούνται σε τακτά χρονικά διαστήματα, δρομολογούν τα προς εκτύπωση αρχεία από την ουρά εκτύπωσης στους εκτυπωτές)

Ο χρήστης μπορεί να τρέξει πολλά προγράμματα μαζί, δημιουργώντας πολλαπλές διεργασίες. Για παράδειγμα έστω ότι έχετε ένα πρόγραμμα που κάνει πολύ ώρα να ολοκληρωθεί. Αυτό το πρόγραμμα μπορείτε να το τρέξετε στο παρασκήνιο (background) με μικρή προτεραιότητα και όσο εκτελείται αυτό, εσείς να κάνετε άλλα πράγματα στο τερματικό σας. Σημειώστε ότι μερικές φορές δεν είναι απίθανο μια διεργασία που τρέχει να κολλήσει και να πρέπει να την απαλείψετε. Αν η διεργασία αυτή δεν ελέγχεται από το τερματικό σας, μπορείτε να βρείτε τον αριθμό της με την εντολή ps και να την απαλείψετε με την εντολή kill.

1.4 Το σύστημα αρχείων του UNIX

Το σύστημα αρχείων του UNIX αποτελείται από ένα σύνολο αρχείων. Κάθε αρχείο έχει ένα ή περισσότερα ονόματα αρχείων (file names). Υπάρχουν τριών ειδών αρχεία

- Τα κοινά αρχεία τα οποία περιέχουν δεδομένα
- Τα ειδικά αρχεία, με τα οποία γίνεται η προσπέλαση στις περιφερειακές μονάδες, π.χ., τα τερματικά και τους εκτυπωτές, αλλά έχουν και άλλους σκοπούς.
- Οι κατάλογοι αρχείων, που περιέχουν τις πληροφορίες που αφορούν μια ομάδα αρχείων και χρησιμεύουν για να μπορούμε να εντοπίσουμε ένα αρχείο με το όνομα του.

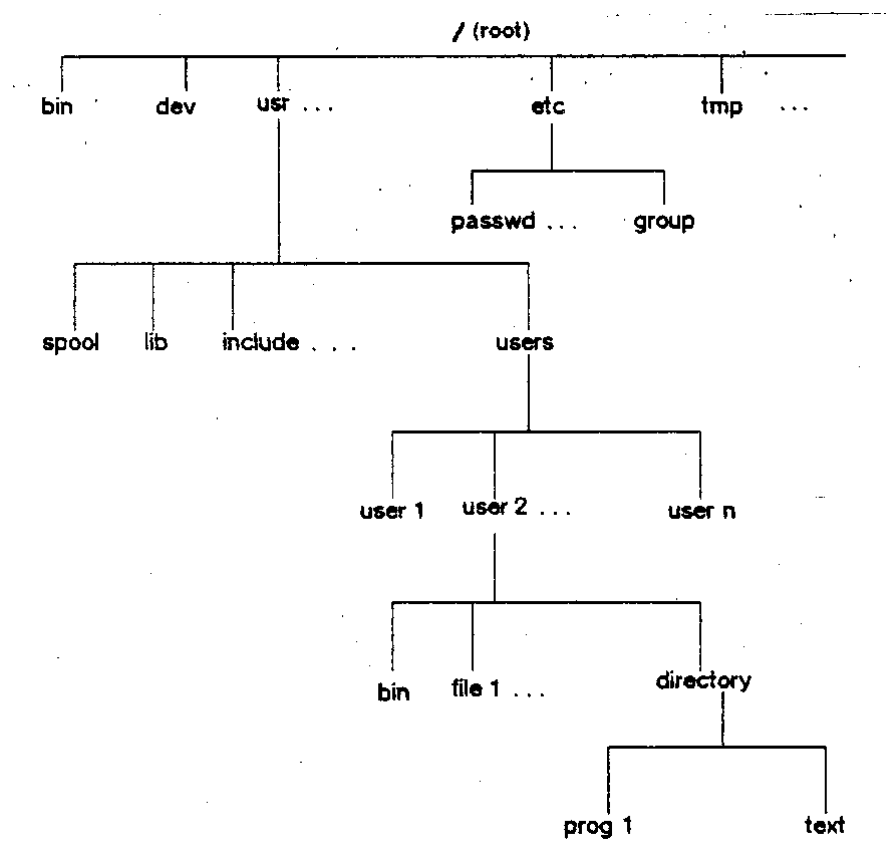
Το σύστημα αρχείων του UNIX είναι ιεραρχικό, διακλαδώνεται από ένα μοναδικό σημείο, τη ρίζα (root). Μπορεί να περιέχει πολλά directories τα οποία με τη σειρά τους να περιέχουν subdirectories ή και αρχεία. Κάθε χρήστης του UNIX έχει το δικό του σημείο εκκίνησης (node) μέσα στο σύστημα αρχείων το οποίο καλείται home directory (οικείος κατάλογος αρχείων). Οτιδήποτε κάτω από αυτό το directory ανήκει στο συγκεκριμένο χρήστη.

Αυτό το directory αποτελεί και το σημείο εκκίνησης κατά την είσοδο του χρήστη στο UNIX. Δηλαδή, όταν κάνουμε login στο σύστημα τότε πάντα μπαίνουμε στο home directory (συμβολίζεται με ~).

Χρησιμοποιώντας εντολές του UNIX ένας χρήστης μπορεί να δημιουργήσει αρχεία και subdirectories, διαμορφώνοντας έτσι ένα προσωπικό περιβάλλον. Στο υπόλοιπο σύστημα αρχείων (πάνω και έξω από το περιβάλλον ενός χρήστη) ο χρήστης έχει περιορισμένη πρόσβαση. Ένας χρήστης μπορεί να κινηθεί οπουδήποτε μέσα στο σύστημα αρχείων αλλάζοντας το τρέχον directory εργασίας.

Για να καθοριστεί ή να αναφερθεί ένα αρχείο UNIX, απλώς γράφουμε το όνομα του αρχείου όσο αυτό βρίσκεται στον τρέχοντα φάκελο εργασίας στον οποίο εργαζόμαστε (current working directory). Αν όμως το αρχείο βρίσκεται οπουδήποτε αλλού μέσα στο σύστημα αρχείων τότε πρέπει να προσδιοριστεί η ακριβής του θέση,

Μονοπάτι (path) ονομάζουμε την ακολουθία των directories, δηλαδή τη διαδρομή εκείνη που οδηγεί στο αρχείο. Το πλήρες όνομα ενός αρχείου λέγεται pathname και περιλαμβάνει το μονοπάτι (path) και το όνομα του αρχείου ώστε το σύστημα να γνωρίζει που βρίσκεται το αρχείο.



Το παραπάνω σχήμα δείχνει τη δενδρική μορφή ενός μέρους του συστήματος αρχείων του UNIX. Ένα path μπορεί να ξεκινά είτε από το root είτε από το δικό μας τρέχον directory (current working directory).

Το pathname μπορεί να είναι τριών ειδών:

1) το **absolute pathname** (απόλυτο όνομα διαδρομής) που πάντα ξεκινά από το root .

π.χ. */usr/users/user2/ directory/ prog1*

2) το **relative pathname** (σχετικό όνομα διαδρομής) που δείχνει πως θα φθάσουμε στον προορισμό μας από το current directory.

πχ- αν current directory είναι το user2 τότε το *directory/ text* είναι relative pathname.

3) **απλό όνομα αρχείου** (filename).

π.χ. prog1 είναι ένα απλό όνομα αρχείου.

Αν για παράδειγμα έχουμε το pathname: */usr/users/user2/file1*

Το τελευταίο τμήμα του pathname (μετά το τελευταίο /) καλείται "ουρά" (tail) ή βασικό όνομα, και το υπόλοιπο μέρος του ονόματος διαδρομής (pathname) καλείται κεφαλή (head).

Προσοχή:

- Τα σχετικά ονόματα διαδρομής ποτέ δεν αρχίζουν με /
- Με το σύμβολο . συμβολίζουμε το current working directory.
- Με το σύμβολο .. συμβολίζουμε το parent directory π.χ- *../user1/file2*
- Με το σύμβολο ../../ αναφέρομαι στο προηγούμενο του parent directory, δηλαδή σε δύο επίπεδα πιο πάνω.
- Τα ονόματα των files και directories μπορούν να φθάνουν μέχρι 14 χαρακτήρες μήκος.
- Κεφαλαία και μικρά γράμματα διαφέρουν για το c-shell.
- Στην ονομασία των files και directories χρησιμοποιούνται γράμματα, αριθμοί και τα σύμβολα " -", " _", και αποφεύγονται τα ειδικά σύμβολα.
- Αν κάποια εντολή ξεπερνά το μήκος μιας γραμμής οθόνης (command line), μπορούμε να συνεχίσουμε στην επόμενη θέτοντας \ (backslash) και πατώντας <return> .

2. Βασικές εντολές UNIX

2.1 Δουλεύοντας με αρχεία και καταλόγους

Τα πάντα στο Unix απεικονίζονται σε αρχεία τα οποία είναι οργανωμένα σε καταλόγους που σχηματίζουν μία ιεραρχία. Για να δούμε σε ποιον κατάλογο βρισκόμαστε χρησιμοποιούμε την εντολή **pwd**.

Μπορούμε να κινηθούμε στην ιεραρχία των καταλόγων χρησιμοποιώντας την εντολή **cd**. Ο προκαθορισμένος κατάλογος όταν μπαίνουμε στο σύστημα είναι ο **/home/users/login_name**, όπου **login_name** είναι ο κωδικός μας. Μπορούμε ανά πάσα στιγμή να μεταφερθούμε στον κατάλογο αυτό δίνοντας **cd** (χωρίς ορίσματα) ή **cd ~** (ο χαρακτήρας «~» δηλώνει πάντα τον κατάλογο του χρήστη). Μπορούμε να αναφερθούμε στον τρέχοντα κατάλογο με τον χαρακτήρα “.” και στο γονικό κατάλογο με τους χαρακτήρες “..” Δίνοντας **cd dirname** κάνουμε τρέχοντα κατάλογο τον κατάλογο **dirname**. Η θέση του **dirname** θεωρείται ότι είναι σχετική με τον τρέχοντα κατάλογο αν ο πρώτος χαρακτήρας δεν είναι ο / ή σχετική με τον πρωταρχικό κατάλογο (root directory) αν ξεκινά με /)

Παράδειγμα: Έστω ότι βρισκόμαστε στον κατάλογο **/home/users/user1**. Αν δώσουμε την εντολή **cd bin** θα μεταφερθούμε στον κατάλογο **/home/users/user1/bin** (εφόσον υπάρχει ο κατάλογος αυτός). Αν δώσουμε την εντολή **cd /bin** τότε θα μεταφερθούμε στον κατάλογο **/bin**.

Για να δημιουργήσουμε έναν κατάλογο χρησιμοποιούμε την εντολή **mkdir dirname**.

Ένας κατάλογος περιέχει πολλά αρχεία και για να δούμε τα ονόματά τους χρησιμοποιούμε την εντολή **ls**. Η εντολή **ls** χωρίς ορίσματα είναι ισοδύναμη με την εντολή “**ls .**”. Αν δώσουμε ορίσματα, τότε αν κάποιο από αυτά είναι αρχείο, η **ls** θα μας παρουσιάσει το όνομά του, ενώ αν είναι κατάλογος θα μας δώσει τα ονόματα των αρχείων και καταλόγων που υπάρχουν σε αυτόν. Η εντολή **ls** εκτός από ορίσματα μπορεί να πάρει και τις πιο κάτω επιλογές:

- a** : Αναγκάζει την **ls** να δείξει και τα αρχεία των οποίων το όνομα ξεκινά με τελεία. Κανονικά τα αρχεία αυτά δεν εμφανίζονται (κρυφά αρχεία)
- A** : Όμοια με την **-a** αλλά δεν δείχνει τους καταλόγους “.” Και “..”
- d** : Αν κάποιο όρισμα είναι κατάλογος τότε παρουσιάζει μόνο το όνομα του καταλόγου και όχι τα αρχεία που περιέχει
- F** : Δίνει πληροφορίες για τον τύπο των αρχείων: οι καταλόγοι έχουν στο τέλος του ονόματός τους ένα /, τα εκτελέσιμα αρχεία ένα * και οι συμβολικοί σύνδεσμοι ένα @
- l** : Δίνει περισσότερες πληροφορίες για κάθε αρχείο: δικαιώματα προσπέλασης, αριθμό συνδέσμων, ιδιοκτήτη, μέγεθος και χρόνο τελευταίας τροποποίησης
- r** : Η σειρά με την οποία παρουσιάζονται τα αρχεία αντιστρέφεται
- R** : Εκτελεί αναδρομικά την **ls** σε κάθε κατάλογο που βρίσκεται

`-t` : Τυπώνει τα αρχεία κατά σειρά που καθορίζεται από το χρόνο τροποποίησης.

Μπορούμε να δώσουμε περισσότερα από ένα option, αλλά πρέπει πάντα να είναι το πρώτο όρισμα.

Παράδειγμα: Δώστε την εντολή `ls -a`. Θα δείτε τα αρχεία `.login` και `.cshrc`.

Τα αρχεία `.login` και `.cshrc` είναι ειδικά αρχεία. Το `.login` εκτελείται κάθε φορά που συνδέεστε (κάνετε login) στο σύστημα. Το `.cshrc` κάθε φορά που αρχίζει η εκτέλεση του φλοιού `csh` (C-shell) ή του φλοιού `tcsh`. Ο φλοιός είναι το περιβάλλον που σας επιτρέπει να επικοινωνείτε με το λειτουργικό σύστημα. Κάθε χρήστης μπορεί να επιλέξει σε ποιον φλοιό θέλει να δουλεύει. Ο εξ'ορισμού φλοιός στο δίκτυο των Sun είναι το `tcsh`. Άλλοι φλοιοί είναι ο Bourne-shell (`sh`) (με την εκκίνηση του οποίου εκτελείται το αρχείο `.profile`, αν υπάρχει), ο `ksh` κλπ.

Άλλα ειδικά αρχεία που μπορεί να υπάρχουν είναι το αρχείο `.logout` που εκτελείται κάθε φορά που αποσυνδέεστε από το σύστημα, το αρχείο `.forward` που καθορίζει σε ποια διεύθυνση προωθούνται τα μηνύματά σας.

Για να δούμε τα περιεχόμενα ενός αρχείου κειμένου χρησιμοποιούμε την εντολή

cat [-options] [filename(s)]

(οι χαρακτήρες [] χρησιμοποιούνται για να δηλώσουν την προαιρετική ύπαρξη κάποιου ορίσματος). Options που μπορούμε να καθορίσουμε είναι τα ακόλουθα:

`-b` : Αγνοεί τις κενές γραμμές και τυπώνει αριθμούς γραμμών
`-n` : Όμοια με το `-b` αλλά δεν αγνοεί τις κενές γραμμές
`-s` : Τυπώνει ένα κενό όπου υπάρχουν πολλά διαδοχικά
`-v` : Δείχνει τους χαρακτήρες που δεν μπορούν να τυπωθούν σαν αριθμούς

Αν η `cat` δεν έχει ονόματα αρχείων στα ορίσματά της τότε διαβάζει από το standard input, που συνήθως είναι το τερματικό. Άρα παρουσιάζει ξανά ότι γράφεται στο τερματικό. Για να δηλώσετε σε κάποια εντολή ότι έχει τελειώσει η είσοδος από το τερματικό, πιέζεται CTRL+D. Γενικά το CTRL+D στο Unix δηλώνει το τέλος του αρχείου (end-of-file).

Ορισμένα αρχεία έχουν μέγεθος μεγαλύτερο απ'όσο χωράει σε μία οθόνη. Για να μπορέσουμε να δούμε τα περιεχόμενα των αρχείων αυτών χρησιμοποιούμε την εντολή `more`, η σύνταξη της οποίας είναι:

more [-nlines] [filename(s)]

Αν υπάρχει η option `-nlines` (π.χ. `more -20 myfile`) το `more` θα τυπώσει `lines` γραμμές και μετά θα περιμένει να πιέσουμε το πλήκτρο `space` για να τυπώσει τις επόμενες `lines` γραμμές ή `return` για να τυπώσει την επόμενη γραμμή. Αν δεν υπάρχει αυτή η option τότε θα τυπώσει τόσες γραμμές όσες είναι η οθόνη. Όπως και με την `cat` αν δεν της δώσουμε για ορίσματα ονόματα αρχείων θα παρουσιάζει ότι υπάρχει στο standard input.

Μπορούμε να αλλάξουμε το όνομα ενός αρχείου με την εντολή **mv**. Η σύνταξή της είναι:

```
mv file1 file2
```

ή

```
mv file1 file2 ... filen directory
```

Στην πρώτη περίπτωση το *file1* αλλάζει όνομα και γίνεται *file2*. Στη δεύτερη περίπτωση τα αρχεία *file1*, *file2* ... *filen* μεταφέρονται (χωρίς να αλλάξουν όνομα) στον κατάλογο με το όνομα *directory*.

Παράδειγμα: η εντολή

```
mv ~/lex_programs/lex.yy.c ~/c_programs/paslex.c
```

μεταφέρει το αρχείο *lex.yy.c* από τον κατάλογο *~/lex_programs* στον κατάλογο *~/c_programs* και το μετονομάζει σε *paslex.c*

Μπορούμε να αντιγράψουμε αρχεία χρησιμοποιώντας την εντολή **cp**. Η σύνταξή της είναι μία από τις ακόλουθες

```
cp f1 f2
```

```
cp f1 f2 ...fn dir
```

```
cp -r dir1 dir2
```

Στην πρώτη περίπτωση το αρχείο *f1* αντιγράφεται στο αρχείο *f2*, στη δεύτερη περίπτωση στον κατάλογο *dir* δημιουργούνται αντίγραφα των αρχείων *f1*, *f2*, ..., *fn* και στην τρίτη περίπτωση δημιουργείται ο κατάλογος *dir2* (αν δεν υπάρχει ήδη) και αντιγράφονται σε αυτόν αναδρομικά όλα τα αρχεία και οι υποκατάλογοι του καταλόγου *dir1*. Στην τελευταία αυτή περίπτωση αν το *dir2* είναι υποκατάλογος του *dir1* τότε η εντολή αυτή δεν θα τελειώσει ποτέ.

Μπορούμε να αναφερόμαστε σε ένα αρχείο (ή έναν κατάλογο) με πολλά ονόματα ή από διαφορετικούς καταλόγους, χωρίς να χρειάζεται να φτιάξουμε πολλά φυσικά αντίγραφα του αρχείου. Αυτό επιτυγχάνεται δημιουργώντας συνδέσμους σε αυτό με την εντολή **ln**, της οποίας η σύνταξη είναι

```
ln [-s] f1 f2
```

όπου *f1* είναι το αρχείο που υπάρχει, ενώ *f2* είναι το νέο όνομα του αρχείου, που θα δημιουργηθεί. Όποιες αλλαγές γίνονται στο ένα από τα δύο αρχεία θα αντικατοπτρίζονται στο άλλο αρχείο και (στην περίπτωση που δεν έχουμε χρησιμοποιήσει την επιλογή *-s*) όταν σβηστεί το ένα το άλλο συνεχίζει να υπάρχει. Στην περίπτωση που θέλουμε να συνδέσουμε καταλόγους ή αρχεία που βρίσκονται σε διαφορετικά συστήματα αρχείων, τότε πρέπει να χρησιμοποιούμε την επιλογή *-s*, αλλά στην περίπτωση αυτή αν σβήσουμε το αρχικό αρχείο ο σύνδεσμος είναι πλέον άχρηστος.

Μπορούμε να διαγράψουμε ένα ή περισσότερα αρχεία χρησιμοποιώντας την εντολή **rm**, η σύνταξη της οποίας είναι:

rm [-irf] f1 ... fn

η οποία σβήνει τα αρχεία *f1 ... fn*. Οι επιλογές έχουν το εξής νόημα:

- i** : (interactive) μας ρωτάει αν θέλουμε να προχωρήσουμε σε διαγραφή του αρχείου
- f** : (force) προχωρά στη διαγραφή του αρχείου χωρίς να μας ρωτήσει
- r** : (recursive) διαγράφει ολόκληρη ιεραρχία από καταλόγους

Ένας κατάλογος αν είναι άδειος και δεν είναι ο τρέχων κατάλογος μπορεί να διαγραφεί χρησιμοποιώντας την εντολή **rmdir dirname**.

2.2 Ανακατευθύνσεις & δίαυλοι

Οι πιο πολλές εντολές του UNIX διαβάζουν από προκαθορισμένη είσοδο και γράφουν σε προκαθορισμένη έξοδο. Όταν ένας χρήστης δουλεύει σε κέλυφος που είναι αλληλεπιδραστικό, όπως συμβαίνει συνήθως, τα αρχεία προκαθορισμένης εισόδου και εξόδου αντιστοιχούν με το τερματικό του χρήστη. Ένα πρόγραμμα παίρνει τα στοιχεία εισόδου από το τερματικό μας και στέλνει τα στοιχεία εξόδου επίσης στο τερματικό μας. Επίσης, υπάρχει ένα προκαθορισμένο αρχείο λαθών που χρησιμοποιείται για τα μηνύματα λαθών και για τις άλλες πληροφορίες που σχετίζονται με εκτέλεση μιας εντολής. Οι πληροφορίες που στέλνονται στο προκαθορισμένο αρχείο λαθών στέλνονται και στο τερματικό μας.

Οι προαναφερθείσες προκαθορισμένες εισοδοι και εξοδοι είναι στην ουσία διακριτικά αρχείων (file descriptors) που αναγνωρίζονται από όλες τις διεργασίες:

- **stdin** (standard input – προκαθορισμένη είσοδος)
- **stdout** (standard output – προκαθορισμένη έξοδος) και
- **stderr** (standard error – προκαθορισμένη έξοδος σφαλμάτων)

Το **stdin** ανακατευθύνεται χρησιμοποιώντας το συμβολισμό **< filename**, το **stdout** με το συμβολισμό **> filename** ή **>> filename** και αν θέλουμε να ενώσουμε το **stdout** και το **stderr** και να στείλουμε το αποτέλεσμα σε ένα αρχείο τότε χρησιμοποιούμε το συμβολισμό: **>& filename** ή **>>& filename**. Όταν χρησιμοποιούμε τους συμβολισμούς **> filename** και **>& filename** δημιουργείται το αρχείο *filename*, ενώ αν τυχόν υπάρχει, τα περιεχόμενά του χάνονται. Αν χρησιμοποιήσουμε τους συμβολισμούς **>> filename** ή **>>& filename**, τότε η έξοδος θα προστεθεί στο τέλος του αρχείου, αν αυτό υπάρχει, ενώ αν δεν υπάρχει τότε θα δημιουργηθεί.

Για παράδειγμα, η εντολή **cat < test > result** αντιγράφονται τα περιεχόμενα του αρχείου *test* στο αρχείο *result*. Όπως προαναφέραμε, βάζοντας **>>** μπροστά από το όνομα του αρχείου, τα στοιχεία γράφονται μέσα στο αρχείο αυτό μετά το τέλος των περιεχομένων που υπάρχουν ήδη μέσα στο αρχείο. Στο προηγούμενο παράδειγμα, τα περιεχόμενα του *test* θα προσαρτηθούν μετά το τέλος του περιεχομένου του *result* με την εντολή **cat < test >> result**

Μπορούμε επίσης να συνδέσουμε την προκαθορισμένη έξοδο μίας διεργασίας με την προκαθορισμένη είσοδο μίας άλλης χρησιμοποιώντας διαύλους ή σωληνώσεις (pipes). Μία σωλήνωση δηλώνεται ως

```
command1 | command2
```

Μπορούμε βέβαια να φτιάξουμε σωληνώσεις με περισσότερες εντολές, π.χ. `cmd1 | cmd2 | cmd3 | ... | cmdn`. Αν σε κάποιο σημείο θέλουμε να πάρουμε την ως τότε έξοδο σε κάποιο αρχείο μπορούμε να χρησιμοποιήσουμε την εντολή `tee filename` π.χ.

```
cmd1 | cmd2 | ... | cmdm | tee my_file | cmdn
```

Στην περίπτωση αυτή η έξοδος της εντολής `cmdm` θα γραφτεί στο αρχείο `my_file` και στη συνέχεια θα περάσει ως είσοδος στην εντολή `cmdn`.

Ως παράδειγμα των παραπάνω θεωρείστε την εντολή ***grep "Smith" phones | sort***. Η εντολή αυτή καλεί το πρόγραμμα `grep` το οποίο ξεδιαλέγει από το αρχείο `phones` όλες τις εγγραφές που περιέχουν τη σειρά χαρακτήρων "Smith" και δίνει αυτές τις εγγραφές στην προκαθορισμένη έξοδο του. Αυτές οι εγγραφές στη συνέχεια κατευθύνονται μέσω του διαύλου στο `sort` το οποίο τις αντιμετωπίζει σαν ήταν η προκαθορισμένη είσοδος του. Το τελικό αποτέλεσμα είναι ένας διατεταγμένος πίνακας στην οθόνη με όλες τις γραμμές του αρχείου `phones` οι οποίες περιέχουν το "Smith".

Σε μία γραμμή εντολών, που περιέχει τόσο ανακατευθύνσεις όσο και διαύλους, οι ανακατευθύνσεις έχουν μεγαλύτερη προτεραιότητα. Δηλαδή πρώτα γίνεται η ανακατεύθυνση των εντολών και μετά σχετίζονται με τους διαύλους. Π.χ., η γραμμή εντολών ***grep "Smith" phones | sort > hangups*** ξεδιαλέγει από το αρχείο `phones` τις εγγραφές που περιέχουν το "Smith" όπως και προηγουμένως, τις κατατάσσει με το `sort` και κατατάσσει τις διατεταγμένες εγγραφές στο αρχείο `hangups`.

Η δημιουργία μιας σωλήνωσης προϋποθέτει τη δημιουργία ενός ζεύγους διεργασιών. Η πρώτη διεργασία θα δημιουργήσει την πληροφορία, η οποία στη συνέχεια θα περάσει από το σωλήνα για να δοθεί ως είσοδος στη δεύτερη διεργασία.

2.3 Μεταχαρακτήρες

Μπορούμε να χρησιμοποιήσουμε ειδικούς χαρακτήρες για να αναφερθούμε σε αρχεία με παρόμοιο όνομα χωρίς να χρειάζεται να πληκτρολογήσουμε όλα τα ονόματα των αρχείων αυτών. Οι χαρακτήρες αυτοί είναι οι εξής:

| | | |
|----------------------------|---|--|
| * | : | Μία οποιαδήποτε ακολουθία χαρακτήρων |
| ? | : | Ένας οποιοσδήποτε χαρακτήρας |
| [c1c2...cn] | : | Ένας χαρακτήρας από τους c1 c2 ... cn |
| {word1, word2, ..., wordn} | : | Μία οποιαδήποτε από τις λέξεις word1, word2, ... wordn |

Παραδείγματα:

```
~/*
```

όλα τα αρχεία στον προσωπικό κατάλογο του χρήστη

```
~/b*
```

όλα τα αρχεία στον προσωπικό κατάλογο του χρήστη που ξεκινάνε με το γράμμα b

```
*.c
```

όλα τα αρχεία στον τρέχοντα κατάλογο που τελειώνουν με .c

chap

όλα τα αρχεία στον τρέχοντα κατάλογο που περιέχουν τη λέξη chap

chap?.doc

όλα τα αρχεία που αρχίζουν με chap, τελειώνουν σε .doc και ανάμεσα στο chap και το .doc υπάρχει μόνο ένας χαρακτήρας.

chap[0123456789].doc

τα αρχεία chap1.doc, chap2.doc, chap3.doc, chap4.doc, chap5.doc, chap6.doc, chap7.doc, chap8.doc, chap9.doc, chap0.doc

chap[0-9].doc

το ίδιο με το προηγούμενο

/home/users/{postg, grad}*

τους προσωπικούς καταλόγους όλων των χρηστών με κωδικούς που ξεκινούν με τις λέξεις postg και grad (Στην περίπτωση του δικτύου των Sun, όλων των μεταπτυχιακών).

Για να δούμε τα αρχεία που ξεκινούν από ., πρέπει να δώσουμε κάποιο pattern που να αρχίζει με τελεία. Δηλαδή οι μεταχαρακτήρες δεν αντιστοιχούν στην αρχική τελεία ποτέ.

Πρέπει να τονίσουμε εδώ ότι ο φλοιός αντικαθιστά τους μεταχαρακτήρες με τα ονόματα των αρχείων που ταιριάζουν με τα patterns πριν εκτελέσει την εντολή. Έτσι για παράδειγμα αν στον κατάλογο υπάρχουν τα αρχεία file1, file2, file3, τότε πληκτρολογώντας command *, είναι σαν να λέμε command file1 file2 file3. Επίσης εντολές του τύπου

cp *.c *.bak

δεν έχουν ως αποτέλεσμα την αντιγραφή όλων των αρχείων με κατάληξη .c σε ομώνυμα με κατάληξη *.bak. Αν έχουμε στον κατάλογο τα αρχεία a.c b.c και c.c τότε η εντολή θα είναι ισοδύναμη με cp a.c b.c c.c που είναι συντακτικά λάθος, ενώ αν υπάρχουν μόνο τα a.c και b.c τότε θα είναι ισοδύναμη με cp a.c b.c, οπότε θα καταστρέψει το αρχείο b.c αντιγράφοντας πάνω του το αρχείο a.c.

Ο φλοιός εκλαμβάνει ως λάθος την περίπτωση να μην υπάρχει κανένα αρχείο του οποίου το όνομα να ταιριάζει με τα patterns που δώσαμε και εμφανίζει το μήνυμα "No match".

2.4 Χρήσιμες εντολές

2.4.1 Αλλαγή προνομίων προσπέλασης σε αρχεία

Η εντολή chmod χρησιμοποιείται για την αλλαγή προνομίων προσπέλασης. Η σύνταξή της είναι

chmod *τριψήφιος-οκταδικός-αριθμός filename*

ή

```
chmod [user_specifier] mode_change_specifier filename
```

Στην πρώτη περίπτωση υπάρχουν οι εξής κανόνες: το πρώτο ψηφίο αναφέρεται στον ιδιοκτήτη, το δεύτερο στο group και το τρίτο στους υπόλοιπους. Σε κάθε ψηφίο προσθέτουμε 4 αν θέλουμε να δώσουμε δικαίωμα ανάγνωσης, 2 αν θέλουμε να δώσουμε δικαίωμα εγγραφής και 1 αν θέλουμε να δώσουμε δικαίωμα εκτέλεσης.

Στη δεύτερη περίπτωση ο (προαιρετικός) *user_specifier* καθορίζει σε ποιον εφαρμόζεται η αλλαγή των προνομίων (u = owner, g = group, o = other) και το *mode_change_specifier* δείχνει ποια αλλαγή θα επέλθει (+r, -r, +w, -w, +x, -x).

Παραδείγματα:

```
chmod 740 myfile
```

δίνουμε όλα τα δικαιώματα στον ιδιοκτήτη, δικαίωμα ανάγνωσης στο group και κανένα δικαίωμα στους υπόλοιπους.

```
chmod +x myfile
```

δίνουμε δικαίωμα εκτέλεσης σε όλους (ή μόνο στον ιδιοκτήτη) στο αρχείο myfile. Το δικαίωμα εκτέλεσης σε έναν κατάλογο έχει το νόημα ότι μπορούμε να κινηθούμε μέσα σε αυτό.

```
chmod o-w myfile
```

αφαιρούμε το δικαίωμα εγγραφής από τους χρήστες που δεν ανήκουν στο δικό μου group από το αρχείο myfile.

Με τις εντολές *chown* και *chgrp* μπορούμε να μεταβιβάσουμε σε άλλους την ιδιοκτησία ενός χρήστη ή μιας ομάδας χρηστών πάνω σε μια ομάδα αρχείων και καταλόγων. Η σύνταξη των εντολών αυτών έχει τη μορφή:

```
chown owner file
```

```
chgrp group file
```

όπου *file* είναι μια ομάδα αρχείων και καταλόγων που θέλετε να αλλάξετε την ιδιοκτησία τους.

2.4.2 Σύγκριση αρχείων

Η εντολή `cmp f1 f2` συγκρίνει τα αρχεία *f1* και *f2*. Η εντολή `diff f1 f2` κάνει την ίδια δουλειά με την `cmp` αλλά δίνει διαφορετικού τύπου έξοδο. Με την `cmp` μπορείτε να εξακριβώσετε πολύ γρήγορα αν υπάρχουν διαφορές μεταξύ των αρχείων αλλά δεν παίρνετε πολλές πληροφορίες για τις διαφορές τους. Αντίθετα με το `diff` δημιουργείται μια όσο το δυνατό πιο σύντομη λίστα με οδηγίες που δείχνουν τι πρέπει να γίνει στο ένα αρχείο για να μετασχηματιστεί στο άλλο.

2.4.3 Υπηρεσίες πληροφόρησης & βασικών πράξεων με αρχεία

`date`: Δίνει την ημερομηνία και την ώρα

`who`: Δίνει ποιοι χρήστες είναι συνδεδεμένοι στον υπολογιστή

`finger`: Δίνει πληροφορίες για κάποιο συγκεκριμένο χρήστη ή όλους τους χρήστες που έχουν κάνει login στο σύστημα.

`file filename(s)`: Δίνει πληροφορίες για το περιεχόμενο του αρχείου

`head -no_lines [filename]`: Τυπώνει τις `no_lines` πρώτες γραμμές του αρχείου `filename`

`tail -no_lines [filename]`: Τυπώνει τις `no_lines` τελευταίες γραμμές του αρχείου `filename`

`tail +no_lines [filename]`: Παραλείπει τις γραμμές `1 - no_lines` και τυπώνει τις υπόλοιπες.

`wc [-lwc] [filename(s)]`: μετρά γραμμές, λέξεις ή χαρακτήρες ενός αρχείου (`l` = lines, `w` = words, `c` = characters)

`echo [-n] string`: Τυπώνει το `string`. Αν δεν υπάρχει το `-n` τότε δεν αλλάζει γραμμή.

`sort [-nrtxu] [+pos] [filename(s)]`: Ταξινομεί τις γραμμές των αρχείων. Τα ορίσματα των επιλογών έχουν την ακόλουθη σημασία

`-n` : θεωρεί ότι τα πεδία σύγκρισης είναι αριθμοί και όχι strings

`-r` : αντιστρέφει την ταξινόμηση

`-tx` : ο χαρακτήρας `x` θεωρείται ως field separator

`-u` : αν υπάρχουν διπλές γραμμές τυπώνει τη μία από αυτές

`+pos` : όπου το `pos` έχει τη μορφή `m.n`. Το `m` είναι ο αριθμός πεδίων που θα παραληφθούν (μετρώντας από την αρχή της γραμμής) και `n` είναι ο αριθμός των χαρακτήρων από την αρχή του επόμενου πεδίου που θα παραλειφθούν. Για τη σύγκριση χρησιμοποιείται η υπόλοιπη γραμμή.

`tr [-cd] [str1 [str2]]`: Αλλάζει τους χαρακτήρες του `str1` στους αντίστοιχους του `str2`. Αν δίνει το option `-c` τότε αλλάζουν οι χαρακτήρες που δεν βρίσκονται στο `str1`. Αν υπάρχει το option `-d` τότε οι χαρακτήρες του `str2` διαγράφονται. Η εντολή δεν δέχεται ως ορίσματα ονόματα αρχείων αλλά λαμβάνει είσοδο από την προκαθορισμένη είσοδο και δίνει έξοδο στην προκαθορισμένη έξοδο.

`find pathname_list options_and_commands`: Ψάχνει στη λίστα των διαδρομών που δίνουμε όλη την υπο-ιεραρχία των καταλόγων για να βρει αρχεία που πληρούν κάποιες προϋποθέσεις και εκτελεί κάποιες εντολές. Η αποτίμηση της λίστας των options γίνεται από αριστερά προς τα δεξιά και σταματά όταν κάποια αποτύχει. Υπάρχουν οι ακόλουθες επιλογές:

`-name fname` : Επιτυγχάνει αν το όνομα του αρχείου ταιριάζει με το pattern `fname`

`-type c` : Οπου `c = d` ή `c = f` ή `c = l`. Επιτυγχάνει αν το αρχείο είναι κατάλογος (`d`), απλό αρχείο (`f`) ή σύνδεσμος (`l`)

`-size [+ ή -] size` : Επιτυγχάνει αν το αρχείο έχει μέγεθος `size`. Αν υπάρχει το `+` τότε το αρχείο πρέπει να είναι μεγαλύτερο από `size`, ενώ αν υπάρχει το `-` τότε πρέπει να είναι μικρότερο από `size`. Το μέγεθος μετράται σε μπλοκς των 512 bytes

`-print` : Τυπώνει το όνομα του αρχείου – επιτυγχάνει πάντα

`-exec command` : Εκτελεί την εντολή `command` και επιτυγχάνει αν αυτή επιστρέψει τον κωδικό 0. Στο τέλος της εντολής πρέπει να

βάλουμε τους χαρακτήρες {} \; ενώ το όνομα του αρχείου που βρέθηκε δηλώνεται με τους χαρακτήρες {}

-OK *command* : Ομοια με την `exec` αλλά ζητά επιβεβαίωση πριν προχωρήσει στην εκτέλεση της εντολής

`grep [-v] regular_expression [filename(s)]`: Ψάχνει για μία κανονική έκφραση (*regular expression*) μέσα στα αρχεία ή στην προκαθορισμένη είσοδο. Η κανονική έκφραση αποτελείται από τα εξής:

| | | |
|------------------|---|--|
| C | : | Ο χαρακτήρας c ταιριάζει με τον εαυτό του |
| ^ | : | Αρχή της γραμμής |
| \$ | : | Τέλος της γραμμής |
| . | : | Οποιοσδήποτε χαρακτήρας |
| [c1c2c3 ... cn] | : | Οποιοσδήποτε από τους χαρακτήρες c1 c2 c3 ... cn |
| [^c1c2c3 ... cn] | : | Οποιοσδήποτε χαρακτήρας ΕΚΤΟΣ από τους c1, c2, c3, ..., cn |
| R* | : | Μηδέν ή περισσότερες κανονικές εκφράσεις r |
| R+ | : | Μία ή περισσότερες κανονικές εκφράσεις r |
| R? | : | Μηδέν ή μία κανονικές εκφράσεις r |
| R1r2 | : | Η κανονική έκφραση r1, ακολουθούμενη από την κανονική έκφραση r2 |

Η `grep` διαβάζει διαδοχικά τις γραμμές των αρχείων και τυπώνει αυτές που περιέχουν την κανονική έκφραση. Αν υπάρχει το `option -v` τότε τυπώνει τις γραμμές που δεν περιέχουν την κανονική έκφραση. Υπάρχει και η εντολή `egrep`, με σύνταξη παρόμοια με την `grep`, αλλά επιτρέπει ομαδοποίηση των κανονικών εκφράσεων με παρενθέσεις και έχει και τον τελεστή ή που είναι ο χαρακτήρας | (δηλαδή το `r1 | r2` σημαίνει η κανονική έκφραση `r1` ή η κανονική έκφραση `r2`).

`alias str word(s)`: Αν δώσουμε την εντολή `str` αυτή θα αντικατασταθεί με τα `word(s)`.

2.4.4 Μηχανισμός History

Το shell κρατάει τις πιο πρόσφατες εντολές που έχετε δώσει (το πόσες φαίνεται στη μεταβλητή `history`) τις οποίες μπορείτε να ξανα-εκτελέσετε, χωρίς να χρειάζεται να τις πληκτρολογήσετε. Αυτό γίνεται με τους εξής συμβολισμούς:

!! : η αμέσως προηγούμενη εντολή

!n : η εντολή n

!-n : η εντολή που δώσατε πριν από n returns

!str : η πιο πρόσφατη εντολή που αρχίζει με str

!?str : η πιο πρόσφατη εντολή που περιέχει το str

Μπορούμε να συμπληρώσουμε τις εντολές αυτές με κάποιο string, με να το πληκτρολογήσουμε στο τέλος των συμβολισμών. Π.χ. αν δώσουμε `ls` και μετά `!! bin`, η τελευταία εντολή θα είναι ισοδύναμη με `ls bin`.

2.4.5 Προστασία ειδικών χαρακτήρων

Αρκετοί χαρακτήρες (π.χ. `*`, `?`, `[`, `]`, `{`, `}`, `$`, `!`, `~`, κενό) έχουν ειδική σημασία για το φλοιό. Για να αναιρέσουμε την ειδική σημασία αυτών μπορούμε:

- να βάλουμε μπροστά τους ένα backslash (`\`)
- να τους εσωκλείσουμε σε μονά εισαγωγικά (`' '`)
- να τους εσωκλείσουμε σε διπλά εισαγωγικά (`" "`)

Οι χαρακτήρες (`\`, `'`, `"`) είναι επίσης ειδικοί και πρέπει να προστατεύονται. Τα μονά εισαγωγικά επιτρέπουν σε όλους τους χαρακτήρες που εσωκλείουν να περνάνε χωρίς μετάφραση. Το ίδιο ισχύει και για τα διπλά εισαγωγικά με την εξαίρεση ότι αντικαθίστανται οι συμβολισμοί `$...` που εσωκλείονται, και επίσης αντικαθίστανται και τα backslashes. Τα εισαγωγικά χρησιμεύουν επίσης για να περνάμε πολλές λέξεις σαν ένα όρισμα. Π.χ. αν θέλουμε να βρούμε τις γραμμές που περιέχουν το string `"Hello there"` στο αρχείο `myfile` θα πρέπει να δώσουμε την εντολή

```
grep "Hello there" myfile
```

γιατί η εντολή

```
grep Hello there myfile
```

θα έχει ως αποτέλεσμα να αναζητηθεί το string `Hello` στα αρχεία `there` και `myfile`.

Ειδικοί χαρακτήρες είναι ακόμα οι `(,)`, `'`, `|`, `>`, `<`, `&`, ```, `tab`.

2.4.6 Εκτέλεση διεργασιών στο shell

Μπορούμε να δώσουμε περισσότερες από μία εντολές σε μια μόνο γραμμή, χωρίζοντάς τις με ελληνικά ερωτηματικά (`;`). Οι εντολές αυτές θα εκτελεστούν σειριακά. Αν αντί του `;` χρησιμοποιήσουμε το `&&`, τότε η δεύτερη εντολή θα εκτελεστεί μόνο αν επιτύχει η πρώτη (επιστρέφει status 0), ενώ αν χρησιμοποιήσουμε `||` τότε η δεύτερη εντολή θα εκτελεστεί αν η πρώτη αποτύχει (επιστρέφει status διάφορο από μηδέν).

Μπορούμε να καθορίσουμε ότι θέλουμε μία διεργασία να τρέχει στο background βάζοντας στο τέλος της εντολής το χαρακτήρα `&` (π.χ. `command &`). Οι διεργασίες που εκτελούνται στο background δεν μπορούν να ζητήσουν είσοδο από το τερματικό, μπορούν όμως να τυπώσουν στο τερματικό. Μπορούμε να σταματήσουμε τη διεργασία που τρέχει στο foreground διατηρώντας το δικαίωμα να τη συνεχίσουμε μετά από εκεί που σταμάτησε πατώντας `Ctrl+Z`.

Υπάρχουν οι εξής εντολές ελέγχου διεργασιών:

`jobs` : μας δείχνει τις διεργασίες του background

`ps` : μας δείχνει όλες τις διεργασίες. Μπορεί να πάρει διάφορα όρισματα που διαφέρουν σε συστήματα SunOS και Solaris

`kill [-signal] job_specifier`: στέλνει ένα σήμα (*signal*) στη διεργασία που καθορίζουμε. Το εξ' ορισμού σήμα είναι το `TERM` (terminate), ενώ άλλα χρήσιμα

signals είναι τα HUP (hangup, δηλαδή logout αν αναφερόμαστε στο φλοιό, ή επανεκκίνηση της διεργασίας για άλλες), KILL, CONT (continue), STOP και QUIT.

fg *%job_no*: Συνεχίζει στο foreground την εργασία *job_no*

bg *%job_no*: Συνεχίζει την εκτέλεση της εργασίας *job_no* στο background

nohup *command* &: Συνεχίζει την εκτέλεση της εντολής *command* ακόμη και αν κάνουμε logout

wait: περιμένει να τελειώσουν όλες οι διεργασίες στο background

Ο *job_specifier* που αναφέρεται στην εντολή kill μπορεί να είναι είτε *process_id* (τον αριθμό που δίνει η ps) είτε *%job_no* (τον αριθμό που επιστρέφει η jobs). Εναλλακτικά αντί για *%job_no* μπορούμε να χρησιμοποιούμε τους συμβολισμούς *%str* ή *;%str* που αντιστοιχούν στην εργασία της οποίας η γραμμή εντολών αρχίζει ή περιλαμβάνει, αντίστοιχα, το *str*, ή τους συμβολισμούς *%* και *%+* που αναφέρονται στην τρέχουσα εργασία ή *%-* που αναφέρεται στην προηγούμενη εργασία.

Για να μας ειδοποιεί το c-shell για τις αλλαγές στην κατάσταση μίας διεργασίας που εκτελείται στο background χρησιμοποιούμε την εντολή *notify %job_no*. Αν δώσουμε *set notify* τότε το C-shell θα μας ειδοποιεί για τις αλλαγές στην κατάσταση όλων των διεργασιών του background. Το αποτέλεσμα της *set notify* αίρεται με την εντολή *unset notify*.

3. Ο κειμενογράφος πλήρους οθόνης vi

Εντολή *vi*: Η εντολή για την προετοιμασία κειμένου μέσα στο αρχείο “όνομα αρχείου” είναι:

vi όνομα αρχείου

Αν δεν υπάρχει τέτοιο αρχείο, θα δημιουργηθεί όταν δοθεί αυτή η εντολή.

3.1 Περιβάλλον λειτουργίας του vi

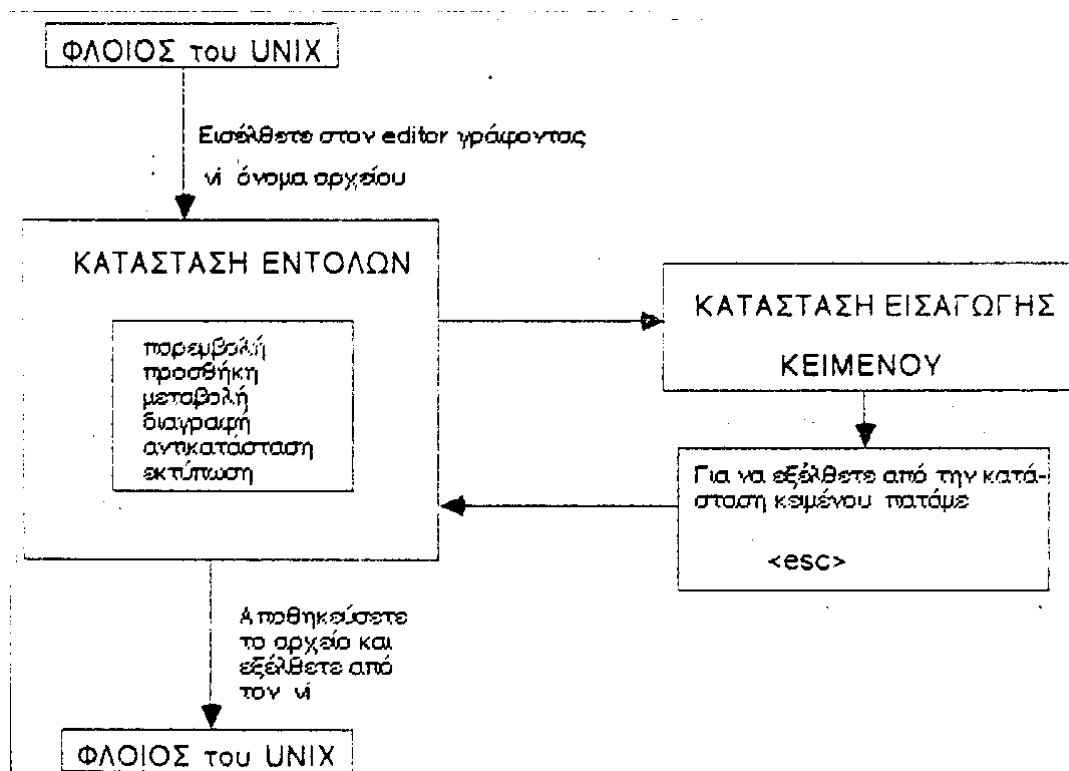
A) Κατάσταση εντολών

Για να εισέλθετε: Σας τοποθετεί στην κατάσταση εντολών όταν καλείτε τον *vi* και σας επιτρέπει να χρησιμοποιήσετε τις εντολές που περιγράφονται στη συνέχεια.

B) Κατάσταση κειμένου

Για να εισέλθετε: Οποιαδήποτε από τις παρακάτω εντολές θα σας τοποθετήσει στην κατάσταση εισαγωγής κειμένου: a, i, o, O. και σας επιτρέπει να χρησιμοποιήσετε το πληκτρολόγιο για να εισάγετε κείμενο.

Για να εισέλθετε εκ νέου σε κατάσταση εντολών από την κατάσταση κειμένου, πατήστε το πλήκτρο ESC.



Οι εντολές του νί εφαρμόζονται στη θέση του cursor .Οι εντολές αυτές σας βοηθούν να τοποθετήσετε τον cursor στη θέση που θέλετε να βρίσκεται μέσα στο κείμενο. Ο cursor δεν θα μετακινηθεί πέρα από τα όρια του υπάρχοντος κειμένου. Τα βέλη ↑,↓,←,→, σας δίνουν τη δυνατότητα να μετακινηθείτε μέσα στο κείμενο. Ισοδύναμα ο cursor μπορεί να μετακινηθεί με τη χρήση των παρακάτω πλήκτρων (όταν βρισκόμαστε σε κατάσταση εντολών).

j: Μετακινεί τον cursor προς τα κάτω μία γραμμή.

k: Μετακινεί τον cursor προς τα πάνω μία γραμμή.

h: Μετακινεί τον cursor αριστερό ένα χαρακτήρα.

l: Μετακινεί τον cursor δεξιά ένα χαρακτήρα.

Αυτών των εντολών μπορεί να προηγείται ένας θετικός ακέραιος για να προσδιορίζει τον αριθμό των γραμμών ή χαρακτήρων μετακίνησης του cursor .

nG Μετακινεί τον cursor στην n-οστή γραμμή του αρχείου.

^ Μετακινεί τον cursor στην αρχή της τρέχουσας γραμμής.

\$ Μετακινεί τον cursor στο τέλος της γραμμής.

3.2 Εντολές εισαγωγής κειμένου

a: Προσθέτει κείμενο μετά τη θέση του cursor

i: Παρεμβάλλει κείμενο πριν τη θέση του cursor.

o: Εισάγει μία νέα γραμμή κάτω από τη θέση του cursor.

O: Εισάγει μία νέα γραμμή πάνω από τη θέση του cursor.

3.3 Εντολές Διαγραφής κειμένου

x: Διαγράφει τον χαρακτήρα που βρίσκεται κάτω από τον cursor.

dd: Διαγράφει τη γραμμή που περιέχει τον cursor.

Επίσης ισχύει nx, ndd για διαγραφή πολλών χαρακτήρων ή γραμμών.

r: Αντικαθιστά τον χαρακτήρα που βρίσκεται κάτω από τον cursor με τον επόμενο χαρακτήρα που πληκτρολογείται.

3.4 Αποθήκευση κειμένου και έξοδος από τον κειμενογράφο

Η προετοιμασία του κειμένου γίνεται σε μια προσωρινή περιοχή εργασίας και μπορεί να αποθηκευτεί σε ένα μόνιμο αρχείο.

<esc>: w Γράφει το τρέχον κείμενο μέσα στο μόνιμο αρχείο.

<esc> : q Εξέρχεται, αν δεν υπάρχουν μεταβολές από την τελευταία εντολή w.

<esc>: q! Εξέρχεται χωρίς να γράφονται οι μεταβολές στο κείμενο.

<esc> : wq Γράφει το αρχείο και εξέρχεται.

<esc>:zz Γράφει και εξέρχεται

<esc>: fname Αλλάζει το όνομα του αρχείου σε «name».

<esc> : set nu Δείχνει τους αριθμούς γραμμών του κειμένου που γράφουμε