

Μαθήματα UNIX για Αρχάριους

Ένας οδηγός αρχαρίων για τα λειτουργικά συστήματα **Unix** και **Linux**. Οκτώ απλά μαθήματα που καλύπτουν τα βασικά της γραμμής εντολών του UNIX και του Linux.

Εισαγωγή στο Λειτουργικό Σύστημα UNIX

- Τι είναι το UNIX;
- Αρχεία και Διεργασίες
- Δομή Καταλόγων
- Ξεκινώντας ένα Τερματικό του UNIX

Πρώτο Μάθημα

- Εμφάνιση Αρχείων και Καταλόγων
- Δημιουργία Καταλόγων
- Αλλαγή σε Διαφορετικό Κατάλογο
- Οι Κατάλογοι . και ..
- Διαδρομές
- Περισσότερες Πληροφορίες για Καταλόγους και Διαδρομές

Δεύτερο Μάθημα

- Αντιγραφή Αρχείων
- Μεταφορά Αρχείων
- Διαγραφή Αρχείων και Καταλόγων
- Εμφάνιση Περιεχομένων Αρχείου στην Οθόνη
- Αναζήτηση στα Περιεχόμενα Ενός Αρχείου

Τρίτο Μάθημα

- Ανακατεύθυνση
- Ανακατεύθυνση της Εξόδου
- Ανακατεύθυνση της Εισόδου
- Σωληνώσεις (pipes)

Τέταρτο Μάθημα

- Χαρακτήρες Μπαλαντέρ
- Συμβάσεις στα Ονόματα Αρχείων
- Αναζήτηση στη Βοήθεια

Πέμπτο Μάθημα

- Ασφάλεια Συστήματος Αρχείων (Δικαιώματα Πρόσβασης)
- Αλλαγή Δικαιωμάτων
- Διεργασίες και Jobs
- Εμφάνιση Σταματημένων Διεργασιών και Διεργασιών Παρασκηνίου
- Τερματισμός μιας Διεργασίας

Έκτο Μάθημα

- Άλλες Χρήσιμες Εντολές UNIX

Έβδομο Μάθημα

- Μεταγλώττιση Λογισμικού UNIX
- Ανάκτηση του Πηγαίου Κώδικα
- Αποσυμπίεση του Πηγαίου Κώδικα
- Δημιουργία και Προσαρμογή Makefile
- Δημιουργία του Πακέτου
- Εκτέλεση του Λογισμικού
- Διαγραφή μη-απαραίτητου Κώδικα

Όγδοο Μάθημα

- Μεταβλητές του UNIX
- Μεταβλητές Περιβάλλοντος
- Μεταβλητές Κελύφους
- Χρήση των Μεταβλητών



Το κείμενο αυτό διατίθεται υπό την άδεια [Creative Commons License](#).

Αυτό το παράγωγο έργο βασίζεται στην αρχική εργασία του **M.Stonebank@surrey.ac.uk**, 19 Οκτωβρίου 2001

Μπορείτε να βρείτε το πρωτότυπο Αγγλικό Κείμενο στην τοποθεσία <http://www.ee.surrey.ac.uk/Teaching/Unix/index.html>

Προσαρμογή και απόδοση στα Ελληνικά: Μανώλης Κιαγιάς (manolis at FreeBSD dot org)

Εισαγωγή στο UNIX

Τι είναι το UNIX;

Το UNIX είναι ένα λειτουργικό σύστημα που ξεκίνησε να αναπτύσσεται τη δεκαετία του 1960 και βρίσκεται από τότε σε διαρκή ανάπτυξη. Η έννοια του Λειτουργικού Συστήματος αναφέρεται σε ένα αριθμό προγραμμάτων που κάνει έναν υπολογιστή να λειτουργεί. Το



UNIX είναι ένα σταθερό σύστημα το οποίο έχει την ικανότητα να εκτελεί πολλά προγράμματα ταυτόχρονα και να εξυπηρετεί πολλούς χρήστες την ίδια στιγμή. Είναι κατάλληλο για εξυπηρετητές (servers) μηχανήματα γραφείου (desktops) σταθμούς εργασίας (workstations) και φορητούς υπολογιστές.

Τα συστήματα UNIX διαθέτουν και γραφικό περιβάλλον εργασίας (GUI) παρόμοιο με αυτό που χρησιμοποιείται στα Microsoft Windows. Το γραφικό περιβάλλον παρέχει ένα εύκολο τρόπο εργασίας. Ωστόσο ή γνώση του UNIX είναι απαραίτητη για εργασίες που δεν καλύπτονται από κάποιο γραφικό εργαλείο, ή όταν το γραφικό περιβάλλον δεν είναι διαθέσιμο, για παράδειγμα όταν είμαστε συνδεδεμένοι μέσω απομακρυσμένης πρόσβασης.

Τα Είδη του UNIX

Υπάρχουν αρκετές διαφορετικές εκδόσεις του UNIX, αν και οι περισσότερες έχουν αρκετές ομοιότητες. Οι πιο συνηθισμένες είναι το Sun Solaris, το GNU/Linux, διάφορες εκδόσεις BSD και το MacOS X.

Στο σχολείο μας χρησιμοποιούμε FreeBSD στο Εργαστήριο 1 και Debian Linux στο Εργαστήριο 2. Διαθέτουμε ακόμα ένα FreeBSD και ένα Debian εξυπηρετητή.



Το Λειτουργικό Σύστημα UNIX

Το λειτουργικό σύστημα UNIX αποτελείται από τρία βασικά τμήματα: Τον πυρήνα, το κέλυφος και τα υπόλοιπα προγράμματα.

Ο Πυρήνας

Ο πυρήνας του UNIX είναι το κέντρο του λειτουργικού συστήματος. Διαθέτει χρόνο και μνήμη στα προγράμματα και χειρίζεται το σύστημα αρχείων και τις επικοινωνίες, ανταποκρινόμενο στις κλήσεις συστήματος.

Ως παράδειγμα του τρόπου με τον οποίο συνεργάζονται το κέλυφος και ο πυρήνας, ας υποθέσουμε ότι ένας χρήστης γράφει την εντολή `rm myfile` (η οποία έχει ως αποτέλεσμα τη διαγραφή του αρχείου `myfile`). Το κέλυφος ψάχνει στο σύστημα αρχείων για το αρχείο που περιέχει το πρόγραμμα `rm` και έπειτα ζητάει από τον πυρήνα, μέσω κλήσεων συστήματος να εκτελέσει το `rm` στο `myfile`. Όταν ολοκληρωθεί η διεργασία `rm myfile` το κέλυφος επιστρέφει την προτροπή `%` στο χρήστη, δείχνοντας του έτσι ότι είναι έτοιμο να δεχθεί άλλες εντολές.

Το Κέλυφος

Το κέλυφος ενεργεί ως διεπαφή (ενδιάμεσος) μεταξύ του χρήστη και του πυρήνα. Όταν ο χρήστης εισέλθει στο σύστημα, το πρόγραμμα που είναι υπεύθυνο για την αναγνώριση του (το login) ελέγχει το όνομα χρήστη και τον κωδικό του και κατόπιν ξεκινάει το πρόγραμμα που είναι γνωστό ως κέλυφος. Το κέλυφος είναι ένα πρόγραμμα ερμηνείας εντολών (command line interpreter ή CLI). Ερμηνεύει τις εντολές που γράφει ο χρήστης και κανονίζει για την εκτέλεση τους. Οι εντολές του χρήστη είναι και αυτές προγράμματα: όταν τερματίζουν, το κέλυφος δείχνει ξανά την προτροπή (`%` ή

\$) στο χρήστη δείχνοντας ότι μπορεί να συνεχίσει με την εκτέλεση άλλων εντολών.

Όποιος χρήστης διαθέτει ικανές γνώσεις μπορεί να προσαρμόσει τη συμπεριφορά του κελύφους του. Υπάρχει ακόμα η δυνατότητα χρήσης διαφορετικού κελύφους ακόμα και σε ένα μόνο μηχάνημα. Στο σχολείο μας το κέλυφος που χρησιμοποιούμε από προεπιλογή είναι το **bash**.

Το bash έχει κάποια χαρακτηριστικά που βοηθούν το χρήστη όταν εισάγει εντολές:

Αυτόματη συμπλήρωση ονόματος αρχείου - Πληκτρολογώντας μόνο ένα μέρος μιας εντολής ή ενός ονόματος αρχείου και πιέζοντας το **Tab**, το κέλυφος θα συμπληρώσει αυτόματα το υπόλοιπο μέρος. Αν το κέλυφος βρει περισσότερα από ένα ονόματα που να ταιριάζουν με τα γράμματα που έχουν πληκτρολογηθεί, θα ακουστεί ένας χαρακτηριστικός ήχος (beep) που σημαίνει ότι πρέπει να γράψετε ακόμα μερικά γράμματα και να πιέσετε ξανά το tab.

Ιστορικό - Το κέλυφος διατηρεί μια λίστα των εντολών που έχετε πληκτρολογήσει. Αν πρέπει να επαναλάβετε μια εντολή μπορείτε να χρησιμοποιήσετε τα πάνω και κάτω βελάκια στο πληκτρολόγιο για να δείτε και να εκτελέσετε ξανά τις προηγούμενες εντολές. Μπορείτε επίσης να γράψετε την εντολή history για να δείτε μια λίστα με τις προηγούμενες εντολές σας.

Αρχεία και Διεργασίες

Στο UNIX τα πάντα είναι είτε αρχείο είτε διεργασία.

Μια διεργασία είναι ένα πρόγραμμα που εκτελείται τη δεδομένη στιγμή και μπορεί να αναγνωριστεί από ένα μοναδικό αριθμό διεργασίας, PID (process identifier).

Ένα αρχείο είναι μια συλλογή δεδομένων. Τα αρχεία δημιουργούνται από τους χρήστες με την βοήθεια συντακτών κειμένου, μεταγλωττιστών κ.λ.π.

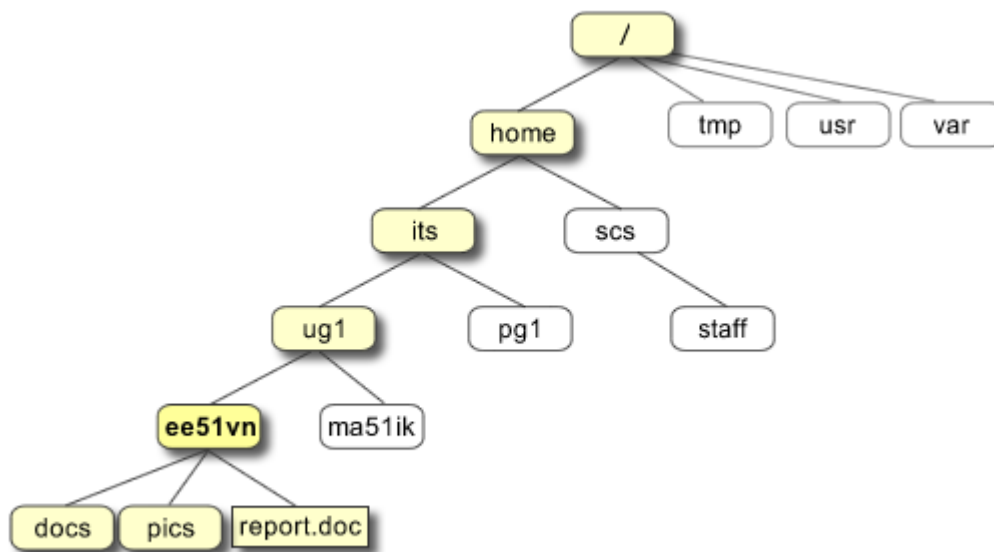
Παραδείγματα αρχείων:

- Ένα έγγραφο (αναφορά, άρθρο κ.λ.π.)
- Το κείμενο ενός προγράμματος όπως έχει γραφεί σε μια γλώσσα προγραμματισμού υψηλού επιπέδου
- Οδηγίες που είναι απευθείας κατανοητές από το μηχάνημα αλλά ακατανόητες από ένα φυσιολογικό χρήστη. Για παράδειγμα μια συλλογή από δυαδικά ψηφία (ένα εκτελέσιμο ή δυαδικό αρχείο).
- Ένας κατάλογος που περιέχει πληροφορίες σχετικά με τα περιεχόμενα του. Τα περιεχόμενα μπορεί να είναι ένα μείγμα από αρχεία και άλλους καταλόγους (υποκαταλόγους).

Δομή Καταλόγων

Τα αρχεία ομαδοποιούνται σε μια δομή καταλόγων. Το σύστημα αρχείων έχει ιεραρχική δομή, όπως ένα ανάστροφο δέντρο. Η κορυφή της ιεραρχίας κατά παράδοση ονομάζεται **root** (και γράφεται ως

για κάθετος /)

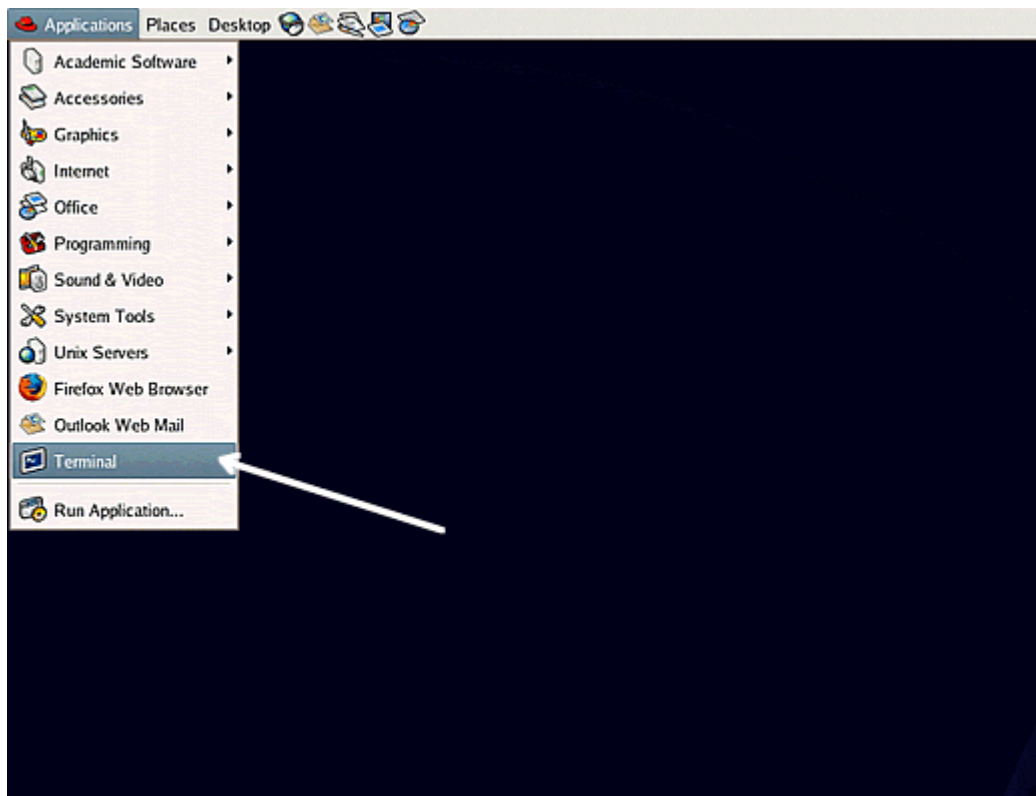


Στο παραπάνω διάγραμμα, βλέπουμε ότι ο προσωπικός κατάλογος (home) του προπτυχιακού φοιτητή **"ee51vn"** περιέχει δύο υποκαταλόγους (**docs** και **pics**) και ένα αρχείο, το **report.doc**.

Η πλήρης διαδρομή για το αρχείο **report.doc** είναι **"/home/its/ug1/ee51vn/report.doc"**

Ξεκινώντας Ένα Τερματικό του UNIX

Για να ανοίξετε ένα τερματικό του UNIX, κάντε click στο εικονίδιο "Terminal" icon στο μενού Applications/Accessories (Η ακριβής τοποθεσία μπορεί να είναι διαφορετική ανάλογα με το γραφικό περιβάλλον που χρησιμοποιείτε).



Θα εμφανιστεί ένα παράθυρο τερματικού με μια προτροπή (% ή \$) στο οποίο μπορείτε να αρχίσετε να γράφετε εντολές.



UNIX: Πρώτο Μάθημα

1.1 Εμφάνιση Αρχείων και Καταλόγων

ls (list)

Μόλις εισέλθετε στο σύστημα, ο αρχικός σας κατάλογος είναι ο προσωπικός σας κατάλογος, γνωστός και ως home directory. Ο προσωπικός σας κατάλογος έχει το ίδιο όνομα με το όνομα χρήστη σας, για παράδειγμα **ee91ab** και σε αυτόν αποθηκεύονται τα αρχεία και οι υποκατάλογοι που δημιουργείτε.

Για να δείτε τι υπάρχει στον προσωπικό σας κατάλογο, πληκτρολογήστε:

```
% ls
```

Η εντολή **ls** (τα μικρά γράμματα L και S) δείχνει μια λίστα των περιεχομένων του τρέχοντος καταλόγου.




Τη δεδομένη στιγμή ίσως να μην υπάρχουν ορατά αρχεία στον κατάλογο σας, και σε αυτή την περίπτωση θα δείτε απλώς ξανά την προτροπή του κελύφους σας. Ίσως όμως ο διαχειριστής συστήματος να έχει ήδη βάλει κάποια αρχεία στο λογαριασμό σας, την ώρα που τον δημιούργησε.

Η **ls** στην πραγματικότητα δεν δείχνει όλα τα αρχεία του καταλόγου σας, αλλά μόνο αυτά που το όνομα τους δεν αρχίζει με τελεία (.). Τα αρχεία που ξεκινάνε με τελεία (.) είναι γνωστά και ως κρυφά αρχεία και συνήθως περιέχουν σημαντικές ρυθμίσεις προγραμμάτων. Είναι κρυφά γιατί δεν πρέπει να τα αλλάζετε, εκτός αν είστε εξοικειωμένος με το UNIX!!!

Για να δείτε όλα τα αρχεία στον κατάλογο σας, συμπεριλαμβανομένων και αυτών που ξεκινάνε με την τελεία, πληκτρολογήστε:

```
% ls -a
```

Όπως μπορείτε να δείτε, το **ls -a** δείχνει και τα αρχεία που κανονικά είναι κρυφά.



```
penguin01
% ls -a
./  core/  Documents/  .login*  mail/  .mailbox*
../  .cshrc*  .hushlogin*  .logout*  Mail/
%
```

Η εντολή `ls` είναι ένα καλό παράδειγμα εντολής που μπορεί να δεχθεί προαιρετικά ορίσματα: το `-a` είναι ένα τέτοιο παράδειγμα. Τα προαιρετικά ορίσματα ή παράμετροι αλλάζουν τη συμπεριφορά μιας εντολής. Το UNIX διαθέτει ένα ενσωματωμένο σύστημα βοήθειας το οποίο μπορεί να σας πει ποιες παραμέτρους δέχεται κάθε εντολή και πως κάθε μια από αυτές μεταβάλλουν τη συμπεριφορά της εντολής (θα το δούμε αργότερα στο μάθημα).

1.2 Δημιουργία Καταλόγων

[mkdir \(make directory\)](#)

Θα δημιουργήσουμε τώρα ένα υποκατάλογο μέσα στον προσωπικό σας κατάλογο, για να κρατάμε τα αρχεία τα οποία θα δημιουργήσετε κατά τη διάρκεια αυτών των μαθημάτων. Για να δημιουργήσετε ένα υποκατάλογο με το όνομα `unixstuff` στον τρέχοντα κατάλογο, πληκτρολογήστε:

```
% mkdir unixstuff
```

Για να δείτε τον κατάλογο που μόλις δημιουργήσατε, πληκτρολογήστε:

```
% ls
```

1.3 Αλλαγή σε Διαφορετικό Κατάλογο

[cd \(change directory\)](#)

Η εντολή `cd directory` αλλάζει τον τρέχοντα κατάλογο σε αυτόν που αναφέρεται ως *'directory'*. Μπορείτε να σκεφτείτε τον τρέχοντα κατάλογο ως αυτόν που βρίσκεστε τη δεδομένη στιγμή (τη θέση σας δηλ. μέσα στο δέντρο του συστήματος αρχείων που είδαμε προηγουμένως).

Για να μετακινηθείτε στον κατάλογο που μόλις δημιουργήσατε, γράψτε:

```
% cd unixstuff
```


Γράψτε `ls` για να δείτε τα περιεχόμενα του (δεν θα δείτε τίποτα γιατί είναι άδειος!)

Άσκηση 1a

Φτιάξτε ένα άλλο κατάλογο μέσα στον **unixstuff**, με το όνομα **backups**

1.4 Οι Κατάλογοι `.` και `..`

Ενώ βρίσκεστε στον κατάλογο **unixstuff** πληκτρολογήστε:

```
% ls -a
```

Όπως μπορείτε να δείτε, στον κατάλογο **unixstuff** (αλλά και σε όλους τους άλλους καταλόγους), υπάρχουν δύο ειδικοί κατάλογοι με τις ονομασίες `.` και `..`

Ο Τρέχων Κατάλογος (`.`)

Στο UNIX, η `.` σημαίνει τον τρέχοντα κατάλογο, έτσι αν πληκτρολογήσετε

```
% cd .
```

Σημείωση: υπάρχει ένα κενό μεταξύ της εντολής `cd` και της τελείας `.`

σημαίνει ότι θα μείνετε στον κατάλογο που είστε (στον **unixstuff**).

Αρχικά αυτό μάλλον δεν θα σας φανεί χρήσιμο, αλλά θα δείτε αργότερα ότι χρησιμοποιώντας το `.` ως το όνομα του τρέχοντος καταλόγου, θα σας γλυτώσει από άσκοπη πληκτρολόγηση.

Ο Γονικός Κατάλογος (`..`)

Οι δύο τελείες `..` σημαίνουν το γονικό του τρέχοντος καταλόγου. Η εντολή:

```
% cd ..
```

θα σας πάει ένα κατάλογο επάνω στην ιεραρχία (στη συγκεκριμένη περίπτωση, στον προσωπικό σας κατάλογο). Δοκιμάστε το τώρα.

Σημείωση: Αν πληκτρολογήσετε `cd` χωρίς κανένα όρισμα, θα γυρίσετε πάντα στον προσωπικό σας κατάλογο. Αυτό είναι χρήσιμο αν έχετε χαθεί μέσα στο σύστημα αρχείων.

1.5 Διαδρομές

`pwd` (print working directory)

Οι διαδρομές σας επιτρέπουν να ανακαλύψετε που βρίσκεστε σε σχέση με όλο το σύστημα αρχείων.

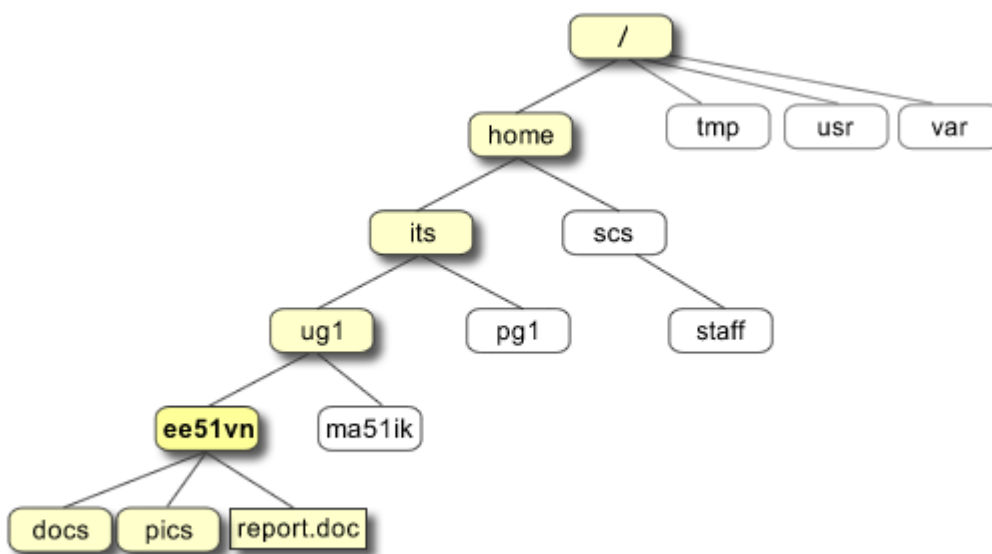
Για παράδειγμα, για να βρείτε την απόλυτη διαδρομή του προσωπικού σας καταλόγου, γράψτε `cd` για να επιστρέψετε σε αυτόν, και έπειτα πληκτρολογήστε:

```
% pwd
```

Η πλήρης διαδρομή θα είναι κάτι σαν το παρακάτω:

```
/home/its/ug1/ee51vn
```

το οποίο σημαίνει ότι ο **ee51vn** (ο προσωπικός σας κατάλογος) βρίσκεται μέσα στον υποκατάλογο **ug1** (τον κατάλογο της ομάδας), ο οποίος με τη σειρά του βρίσκεται μέσα στον υποκατάλογο **its** που βρίσκεται στον υποκατάλογο **home** ο οποίος βρίσκεται μέσα στο ριζικό κατάλογο, γνωστό ως **/**.



Άσκηση 1b

Χρησιμοποιήστε τις εντολές `cd`, `ls` and `pwd` για να εξερευνήσετε το σύστημα αρχείων.

(Θυμηθείτε ότι αν χαθείτε, μπορείτε να χρησιμοποιήσετε την εντολή `cd` χωρίς ορίσματα για να επιστρέψετε άμεσα στον προσωπικό σας κατάλογο)

1.6 Περισσότερες Πληροφορίες για Καταλόγους και Διαδρομές

Κατανόηση των Διαδρομών

Πληκτρολογήστε πρώτα `cd` για να επιστρέψετε στον προσωπικό σας κατάλογο, και έπειτα:

```
% ls unixstuff
```

για να δείτε τα περιεχόμενα του καταλόγου `unixstuff`

Πληκτρολογήστε τώρα:

```
% ls backups
```

Θα δείτε ένα μήνυμα όπως το παρακάτω:

```
backups: No such file or directory
```

Ο λόγος που βλέπετε αυτό το μήνυμα, είναι ότι το **backups** δεν βρίσκεται στον τρέχον κατάλογο σας. Για να χρησιμοποιήσετε μια εντολή σε ένα αρχείο (ή κατάλογο) που δεν βρίσκεται στον τρέχοντα κατάλογο (σε αυτόν που είστε μέσα), θα πρέπει είτε να χρησιμοποιήσετε την εντολή **cd** για να μετακινηθείτε στο σωστό κατάλογο, ή να καθορίσετε την πλήρη διαδρομή του. Για να δείτε τα περιεχόμενα του καταλόγου **backups**, θα πρέπει να γράψετε:

```
% ls unixstuff/backups
```

~ (ο προσωπικός σας κατάλογος)

Μπορείτε να χρησιμοποιήσετε τον χαρακτήρα **~** για να αναφερθείτε στον προσωπικό σας κατάλογο. Μπορείτε να χρησιμοποιήσετε αυτό το χαρακτήρα για να προσδιορίσετε διαδρομές που ξεκινούν από τον κατάλογο σας. Αν πληκτρολογήσετε:

```
% ls ~/unixstuff
```

θα δείτε τα περιεχόμενα του καταλόγου σας **unixstuff**, άσχετο με το που βρίσκεστε τη δεδομένη στιγμή μέσα στο σύστημα αρχείων.

Τι πιστεύετε ότι θα δείξει η εντολή:

```
% ls ~
```

Τι πιστεύετε ότι θα δείξει η εντολή:

```
% ls ~/..
```

Περίληψη

Εντολή	Σημασία
ls	Δείχνει αρχεία και καταλόγους
ls -a	Δείχνει όλα τα αρχεία και τους καταλόγους

<code>mkdir</code>	Δημιουργεί ένα κατάλογο
<code>cd directory</code>	Σας μεταφέρει στον κατάλογο που αναγράφεται
<code>cd</code>	Σας μεταφέρει απευθείας στον προσωπικό σας κατάλογο
<code>cd ~</code>	Σας μεταφέρει στον προσωπικό σας κατάλογο
<code>cd ..</code>	Σας μεταφέρει στον προηγούμενο στην ιεραρχία κατάλογο (γονικό)
<code>pwd</code>	Δείχνει την πλήρη διαδρομή του τρέχοντος καταλόγου

UNIX: Δεύτερο Μάθημα

2.1 Αντιγραφή Αρχείων

cp (copy)

Η εντολή `cp file1 file2` δημιουργεί ένα αντίγραφο του **file1** στον τρέχοντα κατάλογο και το ονομάζει **file2**.

Τώρα, θα αντιγράψουμε ένα αρχείο που υπάρχει σε κάποιο σημείο του συστήματος αρχείων (και στο οποίο έχουμε πρόσβαση) μέσα στον κατάλογο `unixstuff` που έχουμε δημιουργήσει, χρησιμοποιώντας την εντολή `cp`.

Αρχικά χρησιμοποιήστε την εντολή `cd` για να μετακινηθείτε στον κατάλογο **unixstuff**.

```
% cd ~/unixstuff
```

Έπειτα πληκτρολογήστε στην προτροπή:

```
% cp /vol/examples/tutorial/science.txt .
```

Σημείωση: Μην ξεχάσετε την τελεία `.` στο τέλος. Θυμηθείτε ότι η τελεία στο UNIX συμβολίζει τον τρέχοντα κατάλογο.

Η παραπάνω εντολή σημαίνει "αντέγραψε το αρχείο **science.txt** στον τρέχοντα κατάλογο, διατηρώντας το ίδιο όνομα".

Άσκηση 2a

Δημιουργήστε ένα αντίγραφο ασφαλείας του αρχείου **science.txt**, αντιγράφοντας το σε ένα αρχείο με όνομα **science.bak**

2.2 Μεταφορά Αρχείων

mv (move)

Η εντολή `mv file1 file2` μεταφέρει (ή μετονομάζει) το **file1** στο **file2**.

Για να μετακινήσετε ένα αρχείο από μια θέση σε μια άλλη, χρησιμοποιήστε την εντολή `mv`. Αυτό έχει ως αποτέλεσμα την μετακίνηση αντί για την αντιγραφή του αρχείου, έτσι στο τέλος θα έχετε ένα αντίγραφο του αρχείου αντί για δύο.

Μπορείτε επίσης να χρησιμοποιήσετε την εντολή `mv` για να μετονομάσετε ένα αρχείο. Αυτό γίνεται αν το μετακινήσετε στον ίδιο κατάλογο αλλά με διαφορετικό όνομα.

Θα μετακινήσουμε τώρα το αρχείο **science.bak** στον κατάλογο **backup** που δημιουργήσαμε προηγουμένως.

Αρχικά μετακινηθείτε στον κατάλογο `unixstuff` (ελπίζουμε να θυμόσαστε πως γίνεται). Έπειτα, καθώς είστε μέσα στον κατάλογο **unixstuff**, πληκτρολογήστε:

```
% mv science.bak backups/.
```

Πληκτρολογήστε `ls` και `ls backups` για να δείτε αν η εντολή λειτούργησε κανονικά.

2.3 Διαγραφή Αρχείων και Καταλόγων

rm (remove), rmdir (remove directory)

Για να διαγράψετε ένα αρχείο, χρησιμοποιήστε την εντολή `rm`. Ως παράδειγμα, θα δημιουργήσουμε ένα αντίγραφο του αρχείου **science.txt** και έπειτα θα το διαγράψουμε.

Μέσα στον κατάλογο **unixstuff**, πληκτρολογήστε:

```
% cp science.txt tempfile.txt
% ls
% rm tempfile.txt
% ls
```

Μπορείτε να χρησιμοποιήσετε την εντολή `rmdir` για να διαγράψετε ένα κατάλογο (βεβαιωθείτε πρώτα ότι είναι άδειος). Προσπαθήστε να διαγράψετε τον κατάλογο **backups**. Δεν θα μπορέσετε να το κάνετε, καθώς το UNIX δεν θα σας αφήσει να διαγράψετε ένα κατάλογο που δεν είναι άδειος.

Άσκηση 2b

Δημιουργήστε ένα κατάλογο με το όνομα **tempstuff** χρησιμοποιώντας την εντολή `mkdir`, και έπειτα διαγράψτε τον χρησιμοποιώντας την εντολή `rmdir`.

2.4 Εμφάνιση Περιεχομένων Αρχείου στην Οθόνη

clear (καθαρισμός οθόνης)

Πριν ξεκινήσετε με την επόμενη ενότητα, ίσως θέλετε να καθαρίσετε το παράθυρο του τερματικού από τις προηγούμενες εντολές, ώστε η έξοδος των εντολών που ακολουθούν να είναι ξεκάθαρη.

Στην προτροπή, πληκτρολογήστε:

```
% clear
```

Αυτό θα καθαρίσει το παράθυρο και θα σας αφήσει μόνο με την προτροπή \$ στο πάνω μέρος.

cat (concatenate)

Μπορείτε να χρησιμοποιήσετε την εντολή `cat` για να δείτε τα περιεχόμενα ενός αρχείου στην οθόνη σας. Πληκτρολογήστε:

```
% cat science.txt
```

Όπως μπορείτε να δείτε, το αρχείο είναι μεγαλύτερο από το μέγεθος του παραθύρου και έτσι σύντομα κυλάει προς τα πάνω με αποτέλεσμα να μην μπορούμε να το δούμε ολόκληρο.

less

Η εντολή `less` γράφει τα περιεχόμενα ενός αρχείου στην οθόνη, μια σελίδα κάθε φορά. Πληκτρολογήστε:

```
% less science.txt
```

Στο τέλος κάθε σελίδας πιάστε το **[space-bar]** αν θέλετε να δείτε την επόμενη σελίδα, ή το **[q]** για να τερματίσετε την ανάγνωση. Όπως καταλαβαίνετε, η `less` είναι προτιμότερη από την `cat` για μεγάλα αρχεία.

head

Η εντολή `head` γράφει τις δέκα πρώτες γραμμές ενός αρχείου στην οθόνη.

Καθαρίστε πρώτα την οθόνη και μετά πληκτρολογήστε:

```
% head science.txt
```

Έπειτα πληκτρολογήστε:

```
% head -5 science.txt
```

Τι διαφορά προκάλεσε το -5 στην εντολή head;

tail

Η εντολή **tail** εμφανίζει τις δέκα τελευταίες γραμμές ενός αρχείου στην οθόνη.

Καθαρίστε την οθόνη και πληκτρολογήστε:

```
% tail science.txt
```

Ερώτηση: Πως μπορείτε να δείτε τις 15 τελευταίες γραμμές ενός αρχείου;

2.5 Αναζήτηση στα Περιεχόμενα Ενός Αρχείου

Απλή Αναζήτηση με τη Χρήση της less

Χρησιμοποιώντας την εντολή **less**, μπορείτε να ψάξετε το αρχείο για μια λέξη κλειδί. Για παράδειγμα, για να ψάξετε το αρχείο **science.txt** για τη λέξη **'science'**, πληκτρολογήστε:

```
% less science.txt
```

και καθώς εκτελείται η **less**, πληκτρολογήστε μια κάθετο **[/]** ακολουθούμενη από τη λέξη που ψάχνετε

```
/science
```

Όπως μπορείτε να δείτε η, **less** βρίσκει και φωτίζει τη λέξη. Πληκτρολογήστε **[n]** για να βρείτε το επόμενο σημείο που εμφανίζεται η λέξη.

grep (μη ρωτήσετε γιατί τη λένε έτσι)

Η εντολή **grep** αποτελεί ένα από τα τυποποιημένα βοηθητικά εργαλεία του UNIX. Ψάχνει αρχεία για καθορισμένες λέξεις ή patterns. Αρχικά καθαρίστε την οθόνη και έπειτα πληκτρολογήστε:

```
% grep science science.txt
```

Όπως μπορείτε να δείτε, η **grep** εμφάνισε κάθε γραμμή που περιέχει την λέξη **science**.

Η μήπως όχι;

Δοκιμάστε το παρακάτω:

```
% grep Science science.txt
```

Η εντολή **grep** διαχωρίζει μικρά-κεφαλαία. Έτσι το **Science** είναι διαφορετικό από το **science**.

Για να μην γίνεται διαχωρισμός μικρών-κεφαλαίων, χρησιμοποιήστε την επιλογή **-i**, δηλ. πληκτρολογήστε:

```
% grep -i science science.txt
```

Για να ψάξετε για μια φράση ή ένα **pattern**, θα πρέπει να το περικλείσετε σε μονά εισαγωγικά. Για παράδειγμα, για να ψάξετε για τη φράση **spinning top**, πληκτρολογήστε:

```
% grep -i 'spinning top' science.txt
```

Κάποιες ακόμα επιλογές της **grep** είναι:

- v** δείχνει τις γραμμές που ΔΕΝ ταιριάζουν
- n** δείχνει τον αριθμό γραμμής μπροστά από κάθε γραμμή
- c** τυπώνει μόνο τον αριθμό των γραμμών που ταίριαξαν

Δοκιμάστε κάποιες από αυτές τις επιλογές και δείτε τα διαφορετικά αποτελέσματα. Μην ξεχνάτε ότι μπορείτε να χρησιμοποιήσετε περισσότερες από μια επιλογές κάθε φορά. Για παράδειγμα, βρείτε τον αριθμό των γραμμών που δεν περιέχουν τις λέξεις **science** ή **Science**.

```
% grep -ivc science science.txt
```

wc (word count)

Ένα ιδιαίτερα βοηθητικό μικρό πρόγραμμα είναι το **wc** το οποίο μετράει λέξεις. Για να μετρήσετε τις λέξεις στο αρχείο **science.txt**, πληκτρολογήστε:

```
% wc -w science.txt
```

Για να βρείτε πόσες γραμμές έχει το αρχείο, πληκτρολογήστε:

```
% wc -l science.txt
```

Περίληψη

Εντολή	Σημασία
<code>cp file1 file2</code>	Αντέγραψε το file1 και ονόμασε το αντίγραφο file2
<code>mv file1 file2</code>	Μετακίνησε ή μετονόμασε το file1 σε file2
<code>rm file</code>	Διέγραψε ένα αρχείο
<code>rmdir directory</code>	Διέγραψε ένα κατάλογο
<code>cat file</code>	Εμφάνισε ένα αρχείο
<code>less file</code>	Εμφάνισε ένα αρχείο μια σελίδα κάθε φορά
<code>head file</code>	Εμφάνισε τις πρώτες γραμμές ενός αρχείου
<code>tail file</code>	Εμφάνισε τις τελευταίες γραμμές ενός αρχείου
<code>grep 'keyword' file</code>	Ψάξε ένα αρχείο για μια συγκεκριμένη λέξη
<code>wc file</code>	Μέτρηση γραμμών/λέξεων/χαρακτήρων ενός αρχείου

UNIX: Τρίτο Μάθημα

3.1 Ανακατεύθυνση

Οι περισσότερες διεργασίες που ξεκινούν μέσω εντολών του UNIX γράφουν στην τυποποιημένη έξοδο (δηλ. στην οθόνη ή παράθυρο του τερματικού). Πολλές από αυτές τις εντολές παίρνουν την είσοδο τους από την τυποποιημένη είσοδο (δηλ. διαβάζουν από το πληκτρολόγιο). Εκτός από την τυποποιημένη είσοδο και έξοδο υπάρχει επίσης και η τυποποιημένη έξοδος σφάλματος όπου οι διεργασίες εμφανίζουν τα μηνύματα λάθους τους. Από προεπιλογή η έξοδος σφάλματος κατευθύνεται στην οθόνη ή παράθυρο του τερματικού.

Έχουμε ήδη δει τη χρήση της εντολής `cat` με την οποία μπορούμε να εμφανίσουμε τα περιεχόμενα ενός αρχείου στην οθόνη.

Γράψτε τώρα την εντολή `cat` χωρίς να προσδιορίσετε κάποιο αρχείο που να διαβάσει:

```
% cat
```

Γράψτε έπειτα κάποιες λέξεις στο πληκτρολόγιο και πιέστε το πλήκτρο **[Return]**.

Τέλος, πιέστε το πλήκτρο **[Ctrl]** και καθώς το κρατάτε, πιέστε το **[d]** (αυτό εν' συντομία γράφεται **^D**) για να τερματίσετε την είσοδο.

Τι συνέβη;

Αν εκτελέσετε την εντολή `cat` χωρίς να καθορίσετε κάποιο αρχείο για να διαβάσει, διαβάζει την

τυποποιημένη είσοδο (το πληκτρολόγιο) και μόλις λάβει το χαρακτήρα τέλους αρχείου (το **^D**), αντιγράφει την είσοδο στην τυποποιημένη έξοδο (την οθόνη).

Στο UNIX έχουμε τη δυνατότητα να ανακατευθύνουμε τόσο την είσοδο όσο και την έξοδο των εντολών.

3.2 Ανακατεύθυνση της Εξόδου

Χρησιμοποιούμε το σύμβολο **>** για να ανακατευθύνουμε την έξοδο μιας εντολής. Για παράδειγμα, για να δημιουργήσουμε ένα αρχείο με όνομα **list1** το οποίο να περιέχει μια λίστα από φρούτα, πληκτρολογούμε:

```
% cat > list1
```

Γράψτε έπειτα τα ονόματα μερικών φρούτων. Πιέστε **[Return]** μετά από κάθε όνομα.

```
pear
banana
apple
^D {Αυτό σημαίνει [Ctrl] και [d] για τέλος}
```

Αυτό που συμβαίνει, είναι ότι η εντολή **cat** διαβάζει την τυποποιημένη είσοδο (το πληκτρολόγιο) και το **>** ανακατευθύνει την έξοδο - που φυσιολογικά είναι η οθόνη - σε ένα αρχείο με το όνομα **list1**

Για να διαβάσετε τα περιεχόμενα του αρχείου, πληκτρολογήστε:

```
% cat list1
```

Άσκηση 3a

Χρησιμοποιώντας την παραπάνω μέθοδο, δημιουργήστε ένα αρχείο με το όνομα **list2** που να περιέχει τα παρακάτω φρούτα: πορτοκάλι, μανταρίνι, μάνγκο, γκρέιπφρουτ. Χρησιμοποιήστε την κατάλληλη εντολή για να διαβάσετε τα περιεχόμενα του **list2**

3.2.1 Προσθήκη Περιεχομένων σε Ένα Αρχείο

Χρησιμοποιώντας τη μορφή **>>** μπορείτε να προσθέσετε την τυποποιημένη έξοδο στο τέλος ενός ήδη υπάρχοντος αρχείου, χωρίς να διαγράψετε τα περιεχόμενα που έχει ήδη. Έτσι, για να προσθέσετε περισσότερες καταχωρίσεις στο αρχείο **list1**, πληκτρολογήστε:

```
% cat >> list1
```

Και έπειτα πληκτρολογήστε τα ονόματα περισσότερων φρούτων :)

```
peach
grape
orange
^D (Control D για τέλος εισόδου)
```

Για να διαβάσετε τα περιεχόμενα του αρχείου, πληκτρολογήστε:

```
% cat list1
```

Θα πρέπει να έχετε τώρα δύο αρχεία. Το ένα περιέχει έξι φρούτα, το άλλο περιέχει τέσσερα φρούτα.

Θα χρησιμοποιήσουμε τώρα την εντολή `cat` για να ενώσουμε (concatenate) τα δύο αρχεία, το `list1` και το `list2` σε ένα νέο αρχείο με το όνομα `biglist`. Πληκτρολογήστε:

```
% cat list1 list2 > biglist
```

Η εντολή αυτή διαβάζει τα περιεχόμενα των αρχείων `list1` και `list2` με τη σειρά που φαίνονται και γράφει την έξοδο στο αρχείο `biglist`

Για να διαβάσετε τα περιεχόμενα του νέου αρχείου, πληκτρολογήστε:

```
% cat biglist
```

3.3 Ανακατεύθυνση της Εισόδου

Θα χρησιμοποιήσουμε το σύμβολο `<` για να ανακατευθύνουμε την είσοδο της εντολής.

Η εντολή `sort` ταξινομεί αλφαβητικά ή αριθμητικά μια λίστα. Πληκτρολογήστε:

```
% sort
```

Γράψτε έπειτα τα ονόματα μερικών ζώων. Πιέστε [Return] μετά από κάθε όνομα.

```
dog
cat
bird
ape
^D (control d για να σταματήσετε)
```

Η έξοδος θα μοιάζει με την παρακάτω:

```
ape
bird
cat
```

```
dog
```

Μπορείτε να χρησιμοποιήσετε το `<` για να ανακατευθύνετε την είσοδο ώστε να προέρχεται από ένα αρχείο αντί για το πληκτρολόγιο. Για παράδειγμα, για να ταξινομήσετε τη λίστα με τα φρούτα, πληκτρολογήστε:

```
% sort < biglist
```

και θα δείτε την ταξινομημένη λίστα στην οθόνη σας.

Για να γράψετε την ταξινομημένη λίστα σε ένα αρχείο, πληκτρολογήστε:

```
% sort < biglist > slist
```

Χρησιμοποιήστε την εντολή `cat` για να διαβάσετε τα περιεχόμενα του αρχείου `slist`

3.4 Σωληνώσεις

Για να δείτε ποιος άλλος είναι συνδεδεμένος στο σύστημα μαζί με εσάς, πληκτρολογήστε:

```
% who
```

Μια μέθοδος για να δείτε αυτή τη λίστα ονομάτων ταξινομημένη, είναι να πληκτρολογήσετε:

```
% who > names.txt  
% sort < names.txt
```

Αυτό είναι κάπως αργό, και θα πρέπει επίσης να θυμηθείτε να διαγράψετε το προσωρινό αρχείο `names.txt` όταν τελειώσετε. Αυτό που στην πραγματικότητα θέλετε να κάνετε, είναι να ενώσετε με κάποιο τρόπο την έξοδο της εντολής `who` με την είσοδο της εντολής `sort`. Αυτό ακριβώς κάνουν οι σωληνώσεις. Το σύμβολο της σωλήνωσης είναι η κάθετη μπάρα `|`

Για παράδειγμα, πληκτρολογήστε:

```
% who | sort
```

και θα πάρετε το ίδιο αποτέλεσμα με το παραπάνω, αλλά πιο γρήγορα και καθαρά.

Για να βρείτε πόσοι χρήστες βρίσκονται αυτή τη στιγμή στο σύστημα, πληκτρολογήστε:

```
% who | wc -l
```

Άσκηση 3b

Χρησιμοποιώντας σωληνώσεις, δείξτε όλες τις γραμμές των αρχείων `list1` και `list2` που περιέχουν το

γράμμα 'r' και ταξινομήστε το αποτέλεσμα.

Περίληψη

Εντολή	Σημασία
<code>command > file</code>	Ανακατεύθυνση τυποποιημένης εξόδου σε αρχείο
<code>command >> file</code>	Προσθήκη τυποποιημένης εξόδου στο τέλος υπάρχοντος αρχείου
<code>command < file</code>	Ανακατεύθυνση τυποποιημένης εισόδου από αρχείου
<code>command1 command2</code>	Σωλήνωση της εξόδου της <code>command1</code> στην είσοδο της <code>command2</code>
<code>cat file1 file2 > file0</code>	Συνένωση των αρχείων <code>file1</code> και <code>file2</code> στο <code>file0</code>
<code>sort</code>	Ταξινόμηση δεδομένων
<code>who</code>	Λίστα χρηστών που είναι συνδεδεμένοι στο σύστημα τη δεδομένη στιγμή

UNIX: Τέταρτο Μάθημα

4.1 Χαρακτήρες Μπαλαντέρ

Ο Χαρακτήρας *

Ο χαρακτήρας `*` ονομάζεται μπαλαντέρ, και ταιριάζει με οποιοδήποτε πλήθος χαρακτήρων (από μηδέν) σε ένα όνομα αρχείου ή καταλόγου. Για παράδειγμα, στον κατάλογο `unixstuff` πληκτρολογήστε:

```
% ls list*
```

Αυτό θα δείξει όλα τα αρχεία του τρέχοντος καταλόγου που ξεκινάνε με τα γράμματα `list....`

Δοκιμάστε να πληκτρολογήσετε:

```
% ls *list
```

Αυτό θα δείξει όλα τα αρχεία του τρέχοντος καταλόγου που τελειώνουν με τα γράμματα `....list`

Ο Χαρακτήρας ?

Ο χαρακτήρας `?` ταιριάζει με ένα ακριβώς χαρακτήρα.

Έτσι το `?ouse` θα ταιριάζει με αρχεία όπως το `house` και το `mouse`, αλλά όχι το `grouse`.

Δοκιμάστε να πληκτρολογήσετε:

```
% ls ?list
```

4.2 Συμβάσεις στα Ονόματα Αρχείων

Θα πρέπει να σημειώσουμε εδώ ότι ένας κατάλογος είναι απλώς ένας ειδικός τύπος αρχείου. Έτσι οι συμβάσεις που ισχύουν για την ονοματολογία των αρχείων, ισχύουν και για τους καταλόγους.

Όταν ονομάζουμε αρχεία, πρέπει να αποφεύγουμε χαρακτήρες που έχουν ειδική σημασία όπως `/`, `*`, `&` και `%`. Επίσης καλό είναι να αποφεύγετε τα κενά διαστήματα στα ονόματα. Ο ασφαλέστερος τρόπος να ονομάζετε αρχεία, είναι με γράμματα και αριθμούς του λατινικού αλφάβητου, καθώς και με τα σύμβολα `_` (κάτω παύλα) και `.` (τελεία).

Καλά ονόματα	Κακά ονόματα
project.txt	project
my_big_program.c	my big program.c
fred_dave.doc	fred & dave.doc

Κατά παράδοση, τα ονόματα αρχείων ξεκινούν με ένα μικρό γράμμα και μπορεί να τελειώνουν με μια τελεία που ακολουθείται από μια ομάδα γραμμάτων που χαρακτηρίζουν τα περιεχόμενα του αρχείου. Για παράδειγμα, όλα τα αρχεία που περιέχουν κώδικα σε γλώσσα C μπορεί να τελειώνουν με την κατάληξη `.c`, π.χ. `prog1.c`. Έπειτα, για να δείτε μια λίστα όλων των αρχείων που περιέχουν κώδικα σε C στον κατάλογο σας, χρειάζεται απλώς να γράψετε `ls *.c` όταν βρίσκεστε σε αυτό τον κατάλογο.

4.3 Αναζήτηση στη Βοήθεια

Ενσωματωμένες Σελίδες Βοήθειας

Το σύστημά σας περιέχει ενσωματωμένες σελίδες βοήθειας που μπορούν να σας δώσουν πληροφορίες για τις περισσότερες εντολές. Οι σελίδες βοήθειας θα σας πληροφορήσουν σχετικά με τις επιλογές που έχει κάθε εντολή και πως αυτές μεταβάλλουν τη συμπεριφορά της. Πληκτρολογήστε `man command` για να δείτε τη σελίδα βοήθειας (manual) μιας συγκεκριμένης εντολής.

Για παράδειγμα, για να μάθετε λεπτομέρειες για την εντολή `wc` (word count), πληκτρολογήστε:

```
% man wc
```

Εναλλακτικά, μπορείτε να γράψετε:

```
% whatis wc
```

και θα πάρετε μια σύντομη περιγραφή (σε μια γραμμή) της εντολής, χωρίς τις λεπτομέρειες σχετικά με τις επιλογές κ.λ.π.

Apropos

Αν δεν γνωρίζετε το ακριβές όνομα μιας εντολής, δοκιμάστε:

```
% apropos keyword
```

Η εντολή αυτή θα σας δώσει τις εντολές που περιέχουν τη λέξη κλειδί (keyword) στην επικεφαλίδα της σελίδας βοήθειας τους. Για παράδειγμα:

```
% apropos copy
```

Summary

Εντολή	Σημασία
*	Ταίριασμα με οποιοδήποτε πλήθος χαρακτήρων
?	Ταίριασμα με ένα μόνο χαρακτήρα
<i>man command</i>	Σελίδα βοήθειας για μια εντολή
<i>whatis command</i>	Σύντομη περιγραφή μιας εντολής
<i>apropos keyword</i>	Εύρεση των εντολών που περιέχουν το keyword στην περίληψη τους

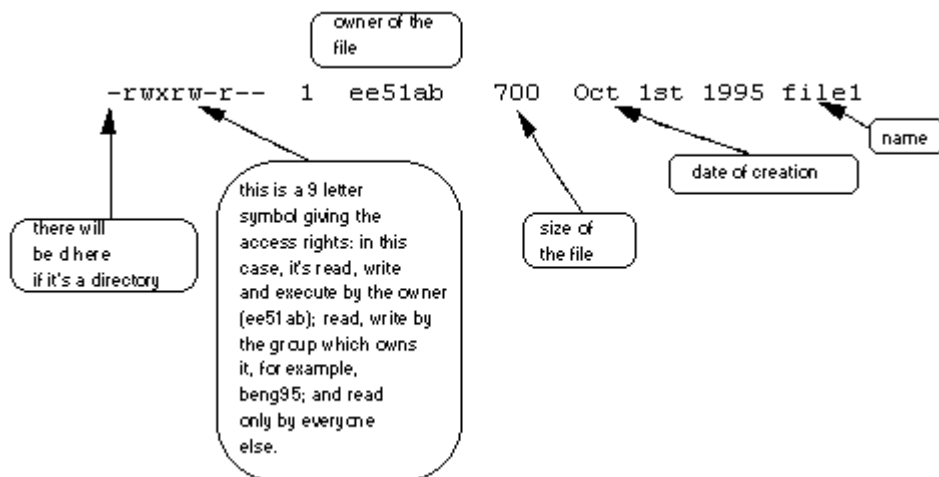
UNIX: Πέμπτο Μάθημα

5.1 Ασφάλεια Συστήματος Αρχείων (Δικαιώματα Πρόσβασης)

Με τρέχοντα κατάλογο τον **unixstuff**, πληκτρολογήστε:

```
% ls -l (l for long listing!)
```

Θα παρατηρήσετε ότι βλέπετε τώρα πολλές περισσότερες λεπτομέρειες σχετικά με τα περιεχόμενα του καταλόγου σας. Θα μοιάζουν με το παράδειγμα που βλέπετε παρακάτω:



Κάθε αρχείο (και κατάλογος) έχει κάποια δικαιώματα πρόσβασης που του αναλογούν και τα οποία μπορείτε να δείτε γράφοντας `ls -l`. Επίσης, η εντολή `ls -lg` δίνει πρόσθετες πληροφορίες, όπως για παράδειγμα το όνομα της ομάδας στην οποία ανήκει το αρχείο (στο παράδειγμα μας, `beng95`):

```
-rwxrw-r-- 1 ee51ab beng95 2450 Sept29 11:52 file1
```

Στην πιο αριστερή στήλη, υπάρχει μια σειρά από 10 σύμβολα στα οποία εμφανίζονται τα γράμματα `d`, `r`, `w`, `x`, `-` και περιστασιακά το `s` ή το `S`. Αν υπάρχει το `d`, θα είναι εντελώς αριστερά και συμβολίζει ότι το όνομα που βλέπουμε είναι στην πραγματικότητα κατάλογος. Αν είναι αρχείο, στη θέση του `d` θα υπάρχει μια παύλα (-).

Τα 9 σύμβολα που απομένουν, δείχνουν τις άδειες ή δικαιώματα πρόσβασης και ομαδοποιούνται σε τρεις ομάδες με τρία σύμβολα η κάθε μία

- Η αριστερή ομάδα των τριών συμβόλων δείχνει τα δικαιώματα για τον χρήστη ο οποίος είναι κάτοχος του αρχείου ή του καταλόγου (`ee51ab` στο παραπάνω παράδειγμα)
- Η μεσαία ομάδα δείχνει τα δικαιώματα για την ομάδα ανθρώπων στην οποία ανήκει το αρχείο ή κατάλογος (`eebeng95` στο παραπάνω παράδειγμα)
- Η δεξιά ομάδα δείχνει τα δικαιώματα για όλους τους υπόλοιπους χρήστες.

Τα σύμβολα `r`, `w`, κ.λ.π. έχουν ελαφρά διαφορετικό νόημα ανάλογα με το αν αναφέρονται σε ένα απλό αρχείο ή σε ένα κατάλογο.

Δικαιώματα Πρόσβασης σε Αρχεία

- Το `r` (ή `-`), δείχνει δικαίωμα ανάγνωσης (ή όχι). Με λίγα λόγια, αν υπάρχει μπορούμε να διαβάσουμε και να αντιγράψουμε το αρχείο.
- Το `w` (ή `-`), δείχνει δικαίωμα εγγραφής (ή όχι). Με λίγα λόγια αν μπορούμε να γράψουμε ή με κάποιο τρόπο να αλλάξουμε το αρχείο.

- Το x (ή -), δείχνει δικαίωμα εκτέλεσης (ή όχι). Με λίγα λόγια αν μπορούμε να εκτελέσουμε αυτό το αρχείο (Αν αυτό φυσικά έχει νόημα, π.χ. αν το αρχείο είναι κάποιο εκτελέσιμο πρόγραμμα).

Δικαιώματα Πρόσβασης σε Κατάλογους

- Το r επιτρέπει σε χρήστες να βλέπουν τη λίστα των αρχείων ενός καταλόγου.
- Το w επιτρέπει σε χρήστες να διαγράφουν αρχεία από τον κατάλογο ή να μετακινούν αρχεία μέσα σε αυτόν.
- Το x επιτρέπει την πρόσβαση σε αρχεία του καταλόγου. Αυτό υπονοεί ότι έχετε τη δυνατότητα να διαβάζετε αρχεία στον κατάλογο, με την προϋπόθεση ότι έχει δικαιώματα στα συγκεκριμένα αρχεία.

Έτσι, για να μπορείτε να διαβάσετε ένα αρχείο, θα πρέπει να έχετε δικαιώματα εκτέλεσης στον κατάλογο στον οποίο βρίσκεται και άρα και στον κατάλογο που τον περιέχει κ.ο.κ.

Μερικά Παραδείγματα

-rwxrwxrwx	Ένα αρχείο το οποίο μπορεί καθένας να διαβάσει, να αλλάξει και να διαγράψει.
-rw-----	Ένα αρχείο το οποίο μπορεί μόνο ο ιδιοκτήτης να γράψει και να διαβάσει, και κανείς άλλος δεν έχει οποιοδήποτε δικαίωμα. Κανείς (ούτε ο ιδιοκτήτης) έχει δικαίωμα εκτέλεσης. Για παράδειγμα, μπορεί να είναι το αρχείο με τα email σας.

5.2 Αλλαγή Δικαιωμάτων Πρόσβασης

chmod (αλλαγή κατάστασης ενός αρχείου)

Μόνο ο ιδιοκτήτης ενός αρχείου μπορεί να χρησιμοποιήσει την `chmod` για να αλλάξει τα δικαιώματα ενός αρχείου. Οι επιλογές της εντολής `chmod` είναι οι παρακάτω:

Σύμβολο	Σημασία
u	χρήστης
g	ομάδα
o	υπόλοιποι
a	όλοι
r	ανάγνωση
w	εγγραφή (και διαγραφή)

x	εκτέλεση (και πρόσβαση στον κατάλογο)
+	προσθήκη δικαιώματος
-	διαγραφή δικαιώματος

Για παράδειγμα, για να αφαιρέσετε τα δικαιώματα εγγραφής και εκτέλεσης στο αρχείο **biglist** για την ομάδα και τους υπόλοιπους χρήστες:

```
% chmod go-rwx biglist
```

Η παραπάνω εντολή δεν θα πειράξει τα υπόλοιπα δικαιώματα (π.χ. του ιδιοκτήτη).

Για να δώσετε δικαιώματα ανάγνωσης και εγγραφής στο αρχείο **biglist** σε όλους:

```
% chmod a+rw biglist
```

Άσκηση 5a

Δοκιμάστε να αλλάξετε δικαιώματα στο αρχείο **science.txt** και στον κατάλογο **backups**.

Χρησιμοποιήστε την εντολή `ls -l` για να ελέγξετε ότι τα δικαιώματα έχουν πραγματικά αλλάξει.

5.3 Διεργασίες και Jobs

Μια διεργασία είναι ένα πρόγραμμα που εκτελείται και αναγνωρίζεται από ένα μοναδικό αναγνωριστικό διεργασίας, PID. Για να δείτε πληροφορίες για τις διεργασίες σας και το αναγνωριστικό τους, γράψτε:

```
% ps
```

Μια διεργασία μπορεί να βρίσκεται στο προσκήνιο, στο παρασκήνιο, ή σε παύση. Σε γενικές γραμμές το κέλυφος δεν επιστρέφει την προτροπή εντολής μέχρι να τερματιστεί η τρέχουσα διεργασία.

Μερικές διεργασίες χρειάζονται πολύ χρόνο για να εκτελεστούν και στο μεταξύ εμποδίζουν τη χρήση του τερματικού. Αν μεταφέρετε μια τέτοια διεργασία στο παρασκήνιο, θα μπορέσετε να έχετε άμεσα διαθέσιμη την προτροπή εντολών. Μπορείτε έτσι να συνεχίσετε να εκτελείτε άλλες εντολές καθώς η αρχική διεργασία συνεχίζει να εκτελείται.

Εκτέλεση Εντολών στο Παρασκήνιο

Για να ξεκινήσετε μια διεργασία στο παρασκήνιο, προσθέστε ένα **&** στο τέλος της γραμμής εντολής. Για παράδειγμα, η εντολή `sleep` εισάγει μια αναμονή, για όσα δευτερόλεπτα της δώσουμε. Πληκτρολογήστε:

```
% sleep 10
```

Η παραπάνω εντολή περιμένει 10 δευτερόλεπτα πριν επιστρέψει η προτροπή του κελύφους. Μέχρι να δείτε ξανά την προτροπή δεν μπορείτε να κάνετε τίποτα εκτός από το να περιμένετε.

Για να εκτελέσετε την `sleep` στο παρασκήνιο, γράψτε:

```
% sleep 10 &
```

```
[1] 6259
```

Με το `&` η εντολής εκτελείται στο παρασκήνιο και η προτροπή επιστρέφει αμέσως. Μπορείτε να εκτελέσετε άλλα προγράμματα καθώς περιμένετε αυτή την εντολή να τελειώσει.

Η πρώτη γραμμή στο παραπάνω παράδειγμα, έχει πληκτρολογηθεί από τον χρήστη. Η επόμενη γραμμή που αναφέρει το αριθμό της διεργασίας και του `job`, προέρχεται από το μηχάνημα. Ο χρήστης ειδοποιείται για τον αριθμό του `job` (ξεκινάει από τον αριθμό 1) ο οποίος περικλείεται σε αγκύλες και για τον αριθμό της διεργασίας. Επίσης ειδοποιείται όταν τελειώσει η εκτέλεση μιας διεργασίας που εκτελείται στο παρασκήνιο. Η εκτέλεση στο παρασκήνιο είναι χρήσιμη για εργασίες που κάνουν αρκετό χρόνο να ολοκληρωθούν.

Αποστολή στο Παρασκήνιο μιας Διεργασίας που εκτελείται στο Προσκήνιο

Στην προτροπή, πληκτρολογήστε:

```
% sleep 1000
```

Μπορείτε να βάλετε την διεργασία που εκτελείται στο προσκήνιο σε αναμονή, πληκτρολογώντας `^Z`, δηλ. κρατώντας πιεσμένο το πλήκτρο `[Ctrl]` και πιέζοντας το `[z]`. Για να θέσετε κατόπιν τη διεργασία στο παρασκήνιο, πληκτρολογήστε:

```
% bg
```

Σημείωση: Μη στέλνετε στο παρασκήνιο διεργασίες που χρειάζονται είσοδο από τον χρήστη, π.χ. το `vi`

5.4 Λίστα Διεργασιών που Έχουν Διακοπεί ή Εκτελούνται στο Παρασκήνιο

Όταν μια διεργασία εκτελείται στο παρασκήνιο ή έχει διακοπεί, βρίσκεται σε μια λίστα με ένα αντίστοιχο αριθμό `job`. Για να δείτε αυτή τη λίστα, γράψτε:

```
% jobs
```

Θα δείτε μια έξοδο όπως την παρακάτω:

```
[1] Suspended sleep 1000
[2] Running netscape
[3] Running matlab
```

Για να ξεκινήσετε ξανά (στο προσκήνιο) μια διεργασία που έχει διακοπεί, πληκτρολογήστε:

```
% fg %αριθμός job
```

Για παράδειγμα, για να ξεκινήσετε ξανά την εντολή `sleep 1000`, πληκτρολογήστε:

```
% fg %1
```

Αν γράψετε `fg` χωρίς να δώσετε αριθμό `job`, θα επαναφέρετε στο προσκήνιο την τελευταία διεργασία που διακόψατε.

5.5 Τερματίζοντας μια Διεργασία

kill (τερματισμός ή αποστολή σήματος σε μια διεργασία)

Μερικές φορές είναι απαραίτητο να τερματίσουμε μια διεργασία (όταν για παράδειγμα εκτελεί ένα ατέρμονο βρόχο).

Για να τερματίσετε μια διεργασία που εκτελείται στο προσκήνιο, πιέστε **^C** (control c). Για παράδειγμα, εκτελέστε:

```
% sleep 100
^C
```

Για να τερματίσετε μια διεργασία που βρίσκεται στο παρασκήνιο ή έχει διακοπεί, γράψτε:

```
% kill %αριθμός job
```

Για παράδειγμα, εκτελέστε:

```
% sleep 100 &
% jobs
```

Αν για παράδειγμα ο αριθμός του `job` είναι 4, γράψτε:

```
% kill %4
```

Για να ελέγξετε αν το παραπάνω λειτούργησε, ελέγξτε ξανά τη λίστα των `jobs` για να δείτε αν έχει αφαιρεθεί η διεργασία.

ps (κατάσταση διεργασίας)

Εναλλακτικά, μπορείτε να τερματίσετε μια διεργασία χρησιμοποιώντας τον αναγνωριστικό αριθμό της (PID) και την εντολή `kill PID_number`

```
% sleep 1000 &
% ps
```

```
PID TT S TIME COMMAND
20077 pts/5 S 0:05 sleep 1000
21563 pts/5 T 0:00 netscape
21873 pts/5 S 0:25 nedit
```

Για να τερματίσετε τη διεργασία **sleep 1000**, πληκτρολογήστε:

```
% kill 20077
```

και έπειτα πληκτρολογήστε **ps** ξανά για να δείτε αν έχει διαγραφεί από την λίστα.

Αν μια διεργασία αρνείται να τερματίσει, χρησιμοποιήστε την επιλογή **-9**:

```
% kill -9 20077
```

Σημείωση: Δεν μπορείτε να τερματίσετε τις διεργασίες άλλων χρηστών !!!

Περίληψη

Εντολή	Σημασία
<code>ls -lag</code>	Δείχνει τα δικαιώματα πρόσβασης για όλα τα αρχεία
<code>chmod [options] file</code>	Αλλάζει τα δικαιώματα πρόσβασης του συγκεκριμένου αρχείου
<code>command &</code>	Εκτελεί την εντολή στο παρασκήνιο
<code>^C</code>	Τερματίζει την διεργασία που εκτελείται στο προσκήνιο
<code>^Z</code>	Διακόπτει τη διεργασία που εκτελείται στο προσκήνιο
<code>bg</code>	Θέτει στο παρασκήνιο την διεργασία που έχει διακοπεί
<code>jobs</code>	Δείχνει λίστα με τα τρέχοντα jobs
<code>fg %1</code>	Επαναφέρει στο προσκήνιο το job με αριθμό 1

<code>kill %1</code>	Τερματίζει το job με αριθμό 1
<code>ps</code>	Λίστα των τρεχόντων διεργασιών
<code>kill 26152</code>	Τερματίζει την διεργασία με αριθμό 26152

UNIX: Έκτο Μάθημα

Άλλες Χρήσιμες Εντολές του UNIX

quota

Σε ένα περιβάλλον σχολικού εργαστηρίου, συνήθως οι σπουδαστές έχουν διαθέσιμο περιορισμένο χώρο στο δίσκο για τα προσωπικά τους αρχεία. Το μέγεθος ποικίλει, αλλά μια τυπική ελάχιστη τιμή είναι τα 100 Mb. Αν ξεπεράσετε τον προκαθορισμένο αυτό χώρο (που ονομάζεται quota) έχετε διαθέσιμες 7 μέρες για να σβήσετε τα περιττά αρχεία.

Για να ελέγξετε το μέγιστο διαθέσιμο σας χώρο καθώς και πόσο έχετε καταναλώσει, πληκτρολογήστε:

```
% quota -v
```

df

Η εντολή `df` τον ελεύθερο χώρο που έχει απομείνει στο σύστημα αρχείων. Για παράδειγμα, για να δείτε πόσος διαθέσιμος χώρος υπάρχει στο μηχάνημα σας γράψτε:

```
% df -h
```

du

Η εντολή `du` δείχνει σε kilobytes το χώρο που καταλαμβάνει κάθε υποκατάλογος. Είναι χρήσιμη αν έχετε ξεπεράσει το διαθέσιμο χώρο που έχει καθοριστεί για το λογαριασμό σας και θέλετε να δείτε ποιος κατάλογος είναι αυτός που περιέχει τα περισσότερα αρχεία. Στο προσωπικό σας κατάλογο, πληκτρολογήστε:

```
% du -s *
```

Η επιλογή `-s` θα δείξει μόνο την περίληψη (το συνολικό μέγεθος) ενώ το `*` σημαίνει ότι θα ληφθούν υπόψη όλα τα αρχεία και οι κατάλογοι.

gzip

Το πρόγραμμα αυτό συμπιέζει αρχεία, μειώνοντας έτσι το χώρο που καταλαμβάνουν στο δίσκο, και

μας εξοικονομεί χώρο. Για παράδειγμα, πληκτρολογήστε:

```
% ls -l science.txt
```

και σημειώστε το μέγεθος του αρχείου όπως αναφέρεται από την εντολή `ls -l`. Για να συμπιέσετε έπειτα το `science.txt`, πληκτρολογήστε:

```
% gzip science.txt
```

Αυτό θα το συμπιέσει και θα το τοποθετήσει σε ένα αρχείο με το όνομα **science.txt.gz**

Για να δείτε την αλλαγή στο μέγεθος, χρησιμοποιήστε ξανά την εντολή `ls -l`.

Για να αποσυμπιέσετε το αρχείο, χρησιμοποιήστε την εντολή `gunzip`:

```
% gunzip science.txt.gz
```

zcat

Η εντολή `zcat` απεικονίζει αρχεία συμπιεσμένα με το `gzip`, χωρίς να χρειάζεται να τα αποσυμπιέσει πρώτα.

```
% zcat science.txt.gz
```

Αν το κείμενο προχωράει πολύ γρήγορα για σας, περάστε την έξοδο μέσω της εντολής `less`:

```
% zcat science.txt.gz | less
```

file

Η εντολή `file` αναγνωρίζει το είδος του αρχείου ανάλογα με το περιεχόμενό του, για παράδειγμα κείμενο (ASCII), φωτογραφία, συμπιεσμένα δεδομένα κ.α. Για να πάρετε μια αναφορά για όλα τα είδη αρχείων που υπάρχουν στον κατάλογο σας, πληκτρολογήστε:

```
% file *
```

diff

Η εντολή αυτή συγκρίνει τα περιεχόμενα δύο αρχείων και δείχνει τις διαφορές μεταξύ τους. Υποθέστε ότι για παράδειγμα έχετε το αρχείο `file1`, αλλάζετε κάποιο τμήμα του, και το αποθηκεύετε ως `file2`. Για να δείτε τις διαφορές μεταξύ τους, γράψτε:

```
% diff file1 file2
```

Οι γραμμές που ξεκινάνε με `<` αναφέρονται στο `file1`, ενώ αυτές που ξεκινάνε με `>` δείχνουν το `file2`.

find

Η εντολή αυτή ψάχνει στους καταλόγους, για αρχεία και καταλόγους που ταιριάζουν με ένα όνομα, ημερομηνία, μέγεθος ή οποιαδήποτε άλλη παράμετρο έχετε δώσει. Πρόκειται για απλή εντολή, αλλά έχει πάρα πολλές επιλογές - μπορείτε να τις δείτε στην αντίστοιχη σελίδα βοήθειας με την εντολή `man find`.

Για να ψάξετε για όλα τα αρχεία με την κατάληξη `.txt`, ξεκινώντας από τον τρέχοντα κατάλογο (`.`) και συνεχίζοντας σε όλους τους υποκαταλόγους, τυπώνοντας το όνομα του αρχείου στην οθόνη, πληκτρολογήστε:

```
% find . -name "*.txt" -print
```

Για να βρείτε όλα τα αρχεία με μέγεθος μεγαλύτερο από 1 Mb και να δείτε τα αποτελέσματα ως λίστα με λεπτομερείς πληροφορίες, πληκτρολογήστε:

```
% find . -size +1M -ls
```

history

Το κέλυφος κρατάει μια λίστα (χρονικά ταξινομημένη) με τις εντολές που έχετε ήδη εκτελέσει. Κάθε εντολή παίρνει ένα αριθμό ανάλογα με τη σειρά εκτέλεσης της:

```
% history (λίστα ιστορικού εντολών)
```

Αν χρησιμοποιείται το κέλυφος C (`cs`h), μπορείτε να χρησιμοποιήσετε το θαυμαστικό (!) για να ανακαλέσετε τις εντολές με εύκολο τρόπο.

```
% !! (ανάκληση της τελευταίας εντολής)
```

```
% !-3 (ανάκληση της τρίτης πιο πρόσφατης εντολής)
```

```
% !5 (ανάκληση της πέμπτης εντολής από τη λίστα)
```

```
% !grep (ανάκληση της τελευταίας εντολής που ξεκινάει με grep)
```

Μπορείτε να αυξήσετε το μέγεθος της προσωρινής αποθήκευσης του ιστορικού, πληκτρολογώντας:

```
% set history=200
```

UNIX: Έβδομο Μάθημα

7.1 Μεταγλώττιση Πακέτων Λογισμικού στο UNIX

Ένα σύστημα UNIX μπορεί να περιέχει τόσο ελεύθερο όσο και εμπορικό (ιδιόκτητο) λογισμικό. Γενικά είναι δυνατόν ένας χρήστης να κατεβάσει και να μεταγλωττίσει μικρά πακέτα λογισμικού στο δικό του κατάλογο. Αυτό το λογισμικό πιθανόν είναι χρήσιμο μόνο για συγκεκριμένα άτομα.

Για να εγκαταστήσετε λογισμικό από τον πηγαίο κώδικα, είναι απαραίτητα κάποια βήματα.

- Πρέπει να εντοπίσετε και να κατεβάσετε τον πηγαίο κώδικα (συνήθως δίνεται σε συμπιεσμένη μορφή)
- Πρέπει να αποσυμπιέσετε τον πηγαίο κώδικα
- Πρέπει να μεταγλωττίσετε τον πηγαίο κώδικα
- Πρέπει να εγκαταστήσετε το εκτελέσιμο αρχείο που θα προκύψει
- Πρέπει να φτιάξετε κάποιες ρυθμίσεις (path) ώστε να μπορεί να βρεθεί το εκτελέσιμο

Από όλα τα παραπάνω βήματα, πιθανώς το πιο δύσκολο είναι η μεταγλώττιση του πηγαίου κώδικα.

Μεταγλώττιση του Πηγαίου Κώδικα

Όλες οι γλώσσες υψηλού επιπέδου πρέπει να μετατραπούν σε μια μορφή που μπορεί να καταλάβει ο υπολογιστής. Για παράδειγμα, ο πηγαίος κώδικας σε γλώσσα C μετατρέπεται σε μια γλώσσα χαμηλού επιπέδου που ονομάζεται assembly. Η assembly που παράγεται από αυτό το πρώτο στάδιο μετατρέπεται στον αντικειμενικό κώδικα (object code) ο οποίος είναι κομμάτια κώδικα τα οποία ο υπολογιστής μπορεί να καταλάβει άμεσα. Το τελευταίο στάδιο στην μεταγλώττιση περιλαμβάνει την σύνδεση (link) του αντικειμενικού κώδικα με βιβλιοθήκες κώδικα (libraries) οι οποίες περιέχουν συγκεκριμένες ενσωματωμένες συναρτήσεις (functions). Το τελευταίο αυτό στάδιο παράγει ένα εκτελέσιμο πρόγραμμα.

Η εκτέλεση όλων αυτών των βημάτων με το χέρι είναι πολύπλοκη και πέρα από τις δυνατότητες του τυπικού χρήστη. Ευτυχώς, οι προγραμματιστές έχουν φτιάξει μια σειρά από εργαλεία και βοηθητικά προγράμματα που απλοποιούν την διαδικασία.

Η Εντολή make και το Αρχείο Makefile

Η εντολή **make** επιτρέπει στους προγραμματιστές να διαχειριστούν μεγάλα προγράμματα ή ομάδες προγραμμάτων. Βοηθάει στην ανάπτυξη μεγάλων προγραμμάτων καταγράφοντας ποια κομμάτια του προγράμματος έχουν αλλάξει και μεταγλωττίζοντας μόνο τα τμήματα που έχουν μεταβληθεί σε σχέση με την τελευταία μεταγλώττιση.

Το πρόγραμμα **make** διαβάζει τους κανόνες που θα ακολουθήσει για την μεταγλώττιση από ένα αρχείο κειμένου που ονομάζεται **Makefile** και το οποίο βρίσκεται στον ίδιο κατάλογο με τα αρχεία του πηγαίου κώδικα. Περιέχει πληροφορίες σχετικά με τον τρόπο μεταγλώττισης του λογισμικού, π.χ. το επίπεδο βελτιστοποίησης, αν θέλουμε να συμπεριληφθούν μέσα στο εκτελέσιμο οδηγίες αποσφαλμάτωσης (debugging) κ.λ.π. Περιέχει επίσης πληροφορίες για την τοποθεσία

εγκατάστασης του τελικού εκτελέσιμου αρχείου, των σελίδων βοήθειας (manual pages), των αρχείων δεδομένων, τυχόν βιβλιοθηκών που χρησιμοποιούνται από το εκτελέσιμο, αρχείων ρυθμίσεων κ.α.

Μερικά πακέτα απαιτούν να αλλάξετε το Makefile χειροκίνητα για να ρυθμίσετε τον τελικό κατάλογο εγκατάστασης και πιθανόν και άλλες παραμέτρους. Ωστόσο, πολλά πακέτα διανέμονται πλέον με το βοηθητικό πρόγραμμα GNU configure.

configure

Καθώς αυξάνονταν οι παραλλαγές συστημάτων UNIX, έγινε δυσκολότερη η συγγραφή προγραμμάτων που να μπορούν να εκτελεστούν σε όλες τις διανομές. Οι προγραμματιστές συχνά δεν είχαν πρόσβαση σε κάθε διαθέσιμο σύστημα, και τα χαρακτηριστικά μερικών συστημάτων άλλαζαν από έκδοση σε έκδοση. Το σύστημα ρύθμισης και μεταγλώττισης GNU απλοποιεί την μεταγλώττιση προγραμμάτων που διανέμονται με τη μορφή πηγαίου κώδικα. Όλα τα προγράμματα μεταγλωττίζονται με τη βοήθεια μια απλής και τυποποιημένης διαδικασίας δύο βημάτων. Ο χρήστης που θα μεταγλωττίσει το πρόγραμμα δεν χρειάζεται να εγκαταστήσει ειδικά εργαλεία για να εκτελέσει αυτή την εργασία.

Η εντολή `configure` είναι ένα script για το κέλυφος το οποίο προσπαθεί να βρει τις σωστές τιμές που απαιτούνται σε διάφορες μεταβλητές του συστήματος και χρησιμοποιούνται κατά τη μεταγλώττιση. Χρησιμοποιεί αυτές τις τιμές για να δημιουργήσει ένα αρχείο **Makefile** σε κάθε κατάλογο του πακέτου.

Ο απλούστερος τρόπος για τη μεταγλώττιση ενός πακέτου, είναι:

1. `cd` στον κατάλογο που περιέχει τον πηγαίο κώδικα του πακέτου.
2. Πληκτρολογήστε `./configure` για να γίνει ρύθμιση του πακέτου για το σύστημα σας.
3. Πληκτρολογήστε `make` για να μεταγλωττίσετε το πακέτο.
4. Προαιρετικά, πληκτρολογήστε `make check` για να τρέξετε τυχόν διαγνωστικά (προγράμματα ελέγχου) που έρχονται μαζί με το πακέτο.
5. Πληκτρολογήστε `make install` για να εγκαταστήσετε τα προγράμματα, τα αρχεία δεδομένων και την τεκμηρίωση.
6. Προαιρετικά, πληκτρολογήστε `make clean` για να διαγράψετε τα αρχεία του εκτελέσιμου και αντικειμενικού κώδικα από τον κατάλογο με το πηγαίο κώδικα του προγράμματος.

Το βοηθητικό πρόγραμμα configure υποστηρίζει μια μεγάλη ποικιλία επιλογών. Μπορείτε συνήθως να χρησιμοποιήσετε την επιλογή `--help` για να δείτε μια λίστα με ενδιαφέρουσες επιλογές για το συγκεκριμένο configure script που θα εκτελέσετε.

Οι μόνες γενικής φύσεως επιλογές που πρόκειται πιθανόν να χρησιμοποιήσετε είναι οι `--prefix` και

--exec-prefix. Οι επιλογές αυτές χρησιμοποιούνται για να καθορίσουν τους καταλόγους εγκατάστασης.

Ο κατάλογος που δίνεται στην επιλογή **--prefix** θα περιέχει τα αρχεία που δεν έχουν σχέση με την αρχιτεκτονική του μηχανήματος, όπως αρχεία δεδομένων, τεκμηρίωσης και ρυθμίσεων

Ο κατάλογος που δίνεται στην επιλογή **--exec-prefix**, (ο οποίος τυπικά είναι υποκατάλογος του καταλόγου που δίνεται στο **--prefix**), θα περιέχει αρχεία που εξαρτώνται από το συγκεκριμένο μηχάνημα, όπως τα εκτελέσιμα.

7.2 Κατέβασμα του Πηγαίου Κώδικα

Για το συγκεκριμένο παράδειγμα, θα κατεβάσουμε ένα μικρό ΕΛΛΑΚ πρόγραμμα, το οποίο μετατρέπει μεταξύ διαφορετικών μονάδων μέτρησης.

Αρχικά, δημιουργήστε ένα κατάλογο για να αποθηκεύσετε τον πηγαίο κώδικα:

```
% mkdir download
```

[Κατεβάστε από εδώ τον πηγαίο κώδικα](#) και αποθηκεύστε τον στον κατάλογο που δημιουργήσατε.

7.3 Αποσυμπιέστε τον Πηγαίο Κώδικα

Μετακινηθείτε στον κατάλογο **download** και δείτε τα περιεχόμενα του.

```
% cd download
% ls -l
```

Όπως μπορείτε να δείτε, το όνομα του αρχείου έχει κατάληξη `tar.gz`. Η εντολή `tar` μπορεί να αποθηκεύσει αρκετά αρχεία και καταλόγους σε ένα μοναδικό αρχείο `tar`. Το αρχείο που προκύπτει συμπιέζεται έπειτα με τη χρήση του προγράμματος `gzip`, δημιουργώντας έτσι ένα αρχείο `tar.gz`.

Αρχικά, αποσυμπιέστε το αρχείο χρησιμοποιώντας την εντολή `gunzip`. Θα δημιουργήσετε έτσι ένα αρχείο με κατάληξη `.tar`.

```
% gunzip units-1.74.tar.gz
```

Προχωρήστε έπειτα σε εξαγωγή των αρχείων που περιέχονται στο αρχείο `.tar`.

```
% tar -xvf units-1.74.tar
```

Δείτε ξανά τα περιεχόμενα του καταλόγου **download** και έπειτα μετακινηθείτε στον υποκατάλογο **units-1.74**.

```
% cd units-1.74
```

7.4 Ρύθμιση και Δημιουργία του Makefile

Το πρώτο πράγμα που πρέπει να κάνετε είναι να διαβάσετε προσεκτικά τα αρχεία **README** και **INSTALL**. Πρόκειται για απλά αρχεία κειμένου, και μπορείτε να τα δείτε εύκολα με την εντολή `less`. Τα αρχεία αυτά περιέχουν σημαντικές πληροφορίες σχετικά με τη μεταγλώττιση και εκτέλεση του λογισμικού.

Το πακέτο `units` που μεταγλωττίζουμε χρησιμοποιεί το σύστημα GNU `configure` για την μεταγλώττιση του πηγαίου κώδικα. Θα πρέπει να καθορίσουμε τον κατάλογο εγκατάστασης, καθώς ο προεπιλεγμένος θα βρίσκεται στους κύριους καταλόγους του συστήματος όπου δεν έχετε δυνατότητα εγγραφής. Θα πρέπει να δημιουργήσετε ένα κατάλογο εγκατάστασης μέσα στον προσωπικό σας κατάλογο.

```
% mkdir ~/units174
```

Εκτελέστε έπειτα το βοηθητικό πρόγραμμα `configure`, θέτοντας ταυτόχρονα και τον κατάλογο που θα γίνει η εγκατάσταση

```
% ./configure --prefix=$HOME/units174
```

Σημείωση: Η μεταβλητή **\$HOME** είναι ένα παράδειγμα μεταβλητής περιβάλλοντος. Η τιμή του **\$HOME** είναι το πλήρες μονοπάτι για τον προσωπικό σας κατάλογο. Για να το διαπιστώσετε, απλώς πληκτρολογήστε:

```
% echo $HOME
```

για να δείτε τα περιεχόμενα αυτής της μεταβλητής. Θα μιλήσουμε περισσότερο για τις μεταβλητές περιβάλλοντος σε επόμενο μάθημα.

Αν εκτελεστεί σωστά το `configure`, θα δημιουργήσει ένα αρχείο `Makefile` με όλες τις απαραίτητες επιλογές. Μπορείτε να δείτε το `Makefile` αν θέλετε (χρησιμοποιήστε και πάλι την εντολή `less`), αλλά μην αλλάξετε τα περιεχόμενα του.

7.5 Μεταγλώττιση του Πακέτου

Μπορείτε τώρα να ξεκινήσετε τη μεταγλώττιση του πακέτου χρησιμοποιώντας την εντολή `make`.

```
% make
```

Μετά από ένα - δύο λεπτά (εξαρτάται από την ταχύτητα του υπολογιστή σας), θα δημιουργηθούν τα εκτελέσιμα. Μπορείτε να ελέγξετε αν η μεταγλώττιση έγινε σωστά, πληκτρολογώντας:

```
% make check
```

Αν όλα πήγαν καλά, μπορείτε τώρα να εγκαταστήσετε το πακέτο.

```
% make install
```

Η εντολή αυτή θα εγκαταστήσει το πακέτο στον κατάλογο `~/units174` που δημιουργήσαμε προηγουμένως.

7.6 Εκτέλεση του Λογισμικού

Είμαστε πλέον έτοιμοι να εκτελέσουμε το λογισμικό (θεωρώντας ότι όλα λειτουργήσαν).

```
% cd ~/units174
```

Αν δείτε τα περιεχόμενα του καταλόγου `units`, θα βρείτε ένα αριθμό από υποκαταλόγους.

<code>bin</code>	Τα εκτελέσιμα αρχεία
<code>info</code>	Τεκμηρίωση μορφοποιημένη κατάλληλα για το πρόγραμμα GNU <code>info</code>
<code>man</code>	Σελίδες βοήθειας (<code>man pages</code>)
<code>share</code>	Κοινόχρηστα αρχεία δεδομένων

Για να εκτελέσετε το πρόγραμμα, μετακινηθείτε στον κατάλογο `bin` και γράψτε

```
% ./units
```

Για παράδειγμα, μετατρέψτε 6 πόδια σε μέτρα.

```
You have: 6 feet  
You want: metres
```

```
* 1.8288
```

Αν πάρετε την απάντηση 1.8288, συγχαρητήρια! Το πρόγραμμα λειτουργεί κανονικά.

Για να δείτε ποιες μονάδες μπορεί να μετατρέψει, κοιτάξτε το αρχείο δεδομένων στον κατάλογο `share` (η λίστα είναι σχετικά μεγάλη).

Για να διαβάσετε την πλήρη τεκμηρίωση, μετακινηθείτε στον κατάλογο `info` και πληκτρολογήστε:

```
% info --file=units.info
```

7.7 Απογύμνωση του Κώδικα από Άχρηστα Τμήματα

Κατά την ανάπτυξη ενός πακέτου λογισμικού, είναι χρήσιμο για τον προγραμματιστή να συμπεριλάβει πληροφορίες αποσφαλμάτωσης στο εκτελέσιμο αρχείο που προκύπτει. Με τον τρόπο

αυτό, αν εμφανιστούν προβλήματα κατά την εκτέλεση του προγράμματος, ο προγραμματιστής μπορεί να φορτώσει το εκτελέσιμο σε ένα κατάλληλο πρόγραμμα και να βρει τα προβλήματα (bugs).

Αυτό είναι χρήσιμο για τον προγραμματιστή, αλλά άχρηστο για τον χρήστη. Μπορούμε να θεωρήσουμε ότι το πακέτο, από τη στιγμή που έχει τελειώσει και είναι διαθέσιμο για κατέβασμα έχει ήδη δοκιμαστεί και διορθωθεί. Ωστόσο, όταν μεταγλωττίσαμε το παραπάνω πακέτο, συμπεριλάβαμε στο τελικό εκτελέσιμο και τις πληροφορίες αποσφαλμάτωσης. Μια και είναι απίθανο να τις χρειαστούμε ως τελικοί χρήστες του προγράμματος, μπορούμε να τις βγάλουμε από το εκτελέσιμο. Θα έχουμε έτσι το πλεονέκτημα ενός μικρότερου εκτελέσιμου αρχείου που θα εκτελείται και λίγο ταχύτερα.

Θα δούμε το μέγεθος του εκτελέσιμου αρχείου πριν και μετά τη διαδικασία. Μετακινηθείτε στον υποκατάλογο **bin** του καταλόγου εγκατάστασης του προγράμματος **units**:

```
% cd ~/units174/bin
% ls -l
```

Όπως μπορείτε να δείτε, το αρχείο είναι πάνω από 100 kbytes σε μέγεθος. Μπορείτε να δείτε περισσότερες πληροφορίες για το αρχείο χρησιμοποιώντας την εντολή **file**:

```
% file units
```

```
units: ELF 32-bit LSB executable, Intel 80386, version 1,
dynamically linked (uses shared libs), not stripped
```

Για να διαγράψετε από το εκτελέσιμο τις πληροφορίες αποσφαλμάτωσης (debug) και αριθμών γραμμών από το δυαδικό αρχείο, χρησιμοποιήστε την εντολή **strip**:

```
% strip units
% ls -l
```

Όπως μπορείτε να δείτε, το αρχείο τώρα είναι μόλις 36 kbytes - το ένα τρίτο του αρχικού του μεγέθους! Δύο τρίτα του εκτελέσιμου αρχείου καταλαμβάνονταν από τις πληροφορίες αποσφαλμάτωσης.

Ελέγξτε ξανά τις πληροφορίες του αρχείου.

```
% file units
```

```
units: ELF 32-bit LSB executable, Intel 80386, version 1,
dynamically linked (uses shared libs), stripped
```

Σε μερικές περιπτώσεις, μπορείτε να χρησιμοποιήσετε την εντολή **make** για να εγκαταστήσετε δυαδικά αρχεία τα οποία από την αρχή να μην περιέχουν πληροφορίες αποσφαλμάτωσης. Αντί να γράψετε **make install**, πληκτρολογήστε απλώς **make install-strip**

UNIX: Όγδοο Μάθημα

8.1 Μεταβλητές στο UNIX

Οι μεταβλητές είναι ένας τρόπος να περάσουμε πληροφορίες από το κέλυφος στα προγράμματα όταν τα εκτελούμε. Τα προγράμματα κοιτάζουν "το περιβάλλον" για συγκεκριμένες μεταβλητές και αν τις βρουν, χρησιμοποιούν τις τιμές τους. Μερικές από τις μεταβλητές ορίζονται από το σύστημα, άλλες από εσάς και άλλες από το κέλυφος ή οποιοδήποτε πρόγραμμα φορτώνει ένα άλλο πρόγραμμα.

Οι τυποποιημένες μεταβλητές του UNIX χωρίζονται σε δύο κατηγορίες, τις μεταβλητές περιβάλλοντος και τις μεταβλητές κελύφους. Σε γενικές γραμμές, οι μεταβλητές κελύφους εφαρμόζονται μόνο στο κέλυφος που εκτελείται τη δεδομένη στιγμή και χρησιμοποιούνται για να θέσουν συνθήκες εργασίας για κάποιο μικρό χρονικό διάστημα. Οι μεταβλητές περιβάλλοντος έχουν μεγαλύτερη επίδραση, και αυτές που τίθενται κατά την είσοδο σας στο σύστημα είναι έγκυρες για όσο διάστημα παραμένετε συνδεδεμένοι. Κατά σύμβαση, οι μεταβλητές περιβάλλοντος έχουν ονόματα με ΚΕΦΑΛΑΙΑ ΓΡΑΜΜΑΤΑ ενώ οι μεταβλητές κελύφους με μικρά.

8.2 Μεταβλητές Περιβάλλοντος

Ένα παράδειγμα μεταβλητής περιβάλλοντος είναι η μεταβλητή `OSTYPE`. Η τιμή της είναι το λειτουργικό σύστημα που χρησιμοποιείτε. Πληκτρολογήστε:

```
% echo $OSTYPE
```

Μερικά ακόμα παραδείγματα μεταβλητών περιβάλλοντος είναι:

- `USER` (το όνομα χρήστη σας)
- `HOME` (η πλήρης διαδρομή για τον προσωπικό σας κατάλογο)
- `HOST` (το όνομα του υπολογιστή που χρησιμοποιείτε)
- `ARCH` (η αρχιτεκτονική επεξεργαστή του υπολογιστή σας)
- `DISPLAY` (το όνομα της οθόνης που χρησιμοποιείται από τα X Windows)
- `PRINTER` (ο εκτυπωτής που δέχεται εργασίες εκτύπωσης από προεπιλογή)
- `PATH` (οι κατάλογοι που ψάχνει το κέλυφος για να εντοπίσει μια εντολή)

Εύρεση των Τρεχόντων Τιμών των Μεταβλητών

Οι μεταβλητές τύπου ENVIRONMENT (περιβάλλοντος) μπορούν να τεθούν με την `setenv` και να απεικονιστούν με τη χρήση της `printenv` ή `env`. Μπορούν να διαγραφούν με την χρήση της `unsetenv`.

Για να δείτε τις τιμές όλων αυτών των μεταβλητών, πληκτρολογήστε:

```
% printenv | less
```

8.3 Μεταβλητές Κελύφους

Ένα παράδειγμα μεταβλητής του κελύφους, είναι η μεταβλητή που σχετίζεται με την τήρηση ιστορικού. Η τιμή αυτής της μεταβλητής αναφέρεται στο πόσες εντολές θα θυμάται το κέλυφος, επιτρέποντας έτσι στο χρήστη να ανατρέξει στις εντολές που έχει ήδη εκτελέσει. Πληκτρολογήστε:

```
% echo $history
```

Περισσότερα παραδείγματα μεταβλητών κελύφους είναι:

- `cwd` (ο τρέχον κατάλογος σας)
- `home` (η διαδρομή για τον προσωπικό σας κατάλογο)
- `path` (οι κατάλογοι που θα ψάξει το κέλυφος για να βρει μια εντολή)
- `prompt` (το κείμενο που δείχνει ως προτροπή το κέλυφος σας για να εισάγετε μια εντολή)
- `shell` (το κέλυφος με το οποίο κάνατε login στο σύστημα)

Εύρεση Τρεχόντων Τιμών Αυτών των Μεταβλητών

Οι μεταβλητές του κελύφους μπορούν να τεθούν και να απεικονιστούν χρησιμοποιώντας την εντολή `set`. Μπορούν επίσης να διαγραφούν με την εντολή `unset` command.

Για να δείτε τις τιμές όλων αυτών των μεταβλητών, πληκτρολογήστε:

```
% set | less
```

Τελικά ποια είναι η διαφορά μεταξύ path και PATH;

Σε γενικές γραμμές, οι μεταβλητές περιβάλλοντος και κελύφους που έχουν το ίδιο όνομα (εκτός από το γεγονός ότι οι μεν είναι κεφαλαία και οι δε μικρά) είναι χωριστές και ανεξάρτητες εκτός από το γεγονός ότι πιθανόν έχουν τις ίδιες αρχικές τιμές. Ωστόσο υπάρχουν και εξαιρέσεις.

Κάθε φορά που αλλάζουν οι μεταβλητές κελύφους `home`, `user` και `term` ανανεώνονται και οι αντίστοιχες μεταβλητές περιβάλλοντος `HOME`, `USER` και `TERM`. Ωστόσο αν ανανεωθούν πρώτα οι μεταβλητές περιβάλλοντος, οι αντίστοιχες του κελύφους δεν αλλάζουν τιμή.

Οι μεταβλητές PATH και path καθορίζουν τους καταλόγους που ανιχνεύονται για την εύρεση εντολών και προγραμμάτων. Και οι δύο μεταβλητές πάντοτε αντιπροσωπεύουν την ίδια λίστα καταλόγων - όποια από τις δύο και αν αλλάξει, η άλλη ανανεώνεται αυτόματα.

8.4 Χρησιμοποιώντας και Θέτοντας Μεταβλητές

Κάθε φορά που εισέρχεστε στο UNIX σύστημα σας, το σύστημα ψάχνει στον προσωπικό σας κατάλογο για αρχεία αρχικοποίησης. Οι πληροφορίες σε αυτά τα αρχεία χρησιμοποιούνται για να ρυθμίσουν τις παραμέτρους του προσωπικού σας περιβάλλοντος. Τα κελύφη C και TC χρησιμοποιούν δύο αρχεία, τα `.login` και `.cshrc` (προσέξτε ότι και τα δύο αρχεία ξεκινούν με τελεία).

Κατά την είσοδο σας, το κέλυφος C διαβάζει πρώτα το `.cshrc` και μετά το `.login`

Το `.login` χρησιμοποιείται για να θέσει ρυθμίσεις που ισχύουν για όλη την σύνοδο (session) και για να εκτελέσει λειτουργίες που απαιτούνται μόνο στη διαδικασία εισόδου.

Το `.cshrc` χρησιμοποιείται για να θέσει ρυθμίσεις και να εκτελέσει λειτουργίες που σχετίζονται με το κέλυφος και πρέπει να γίνονται κάθε φορά που ξεκινά - άσχετα αν είναι το κέλυφος εισόδου ή όχι).

Σε γενικές γραμμές, είναι σωστότερο να ρυθμίζονται οι μεταβλητές ENVIRONMENT (περιβάλλοντος) στο αρχείο `.login` και οι μεταβλητές SHELL (κελύφους) στο `.cshrc`.

ΠΡΟΣΟΧΗ: ΠΟΤΕ μη βάλετε εντολές που ενεργοποιούν το γραφικό περιβάλλον στα αρχεία `.cshrc` ή `.login`.

8.5 Θέτοντας Μεταβλητές Περιβάλλοντος στο Αρχείο `.cshrc`

Αν για παράδειγμα θέλετε να αλλάξετε τον αριθμό των εντολών κελύφους που αποθηκεύονται στη λίστα ιστορικού, θα πρέπει να θέσετε μια νέα τιμή στη μεταβλητή του κελύφους που ονομάζεται `history`. Η προεπιλεγμένη τιμή είναι 100, αλλά μπορείτε να την αυξήσετε αν θέλετε.

```
% set history = 200
```

Ελέγξτε ότι η παραπάνω εντολή λειτούργησε:

```
% echo $history
```

Το παραπάνω ωστόσο θέτει την μεταβλητή μόνο για όση ώρα εκτελείται το τρέχον κέλυφος. Αν ανοίξετε ένα νέο παράθυρο `xterm` (για παράδειγμα), θα έχει την παλιά τιμή της μεταβλητής. Για να αλλάξετε αυτή την τιμή ΜΟΝΙΜΑ θα πρέπει να προσθέσετε την εντολή `set` μέσα στο αρχείο `.cshrc`.

Ανοίξτε το αρχείο `.cshrc` σε ένα συντάκτη κειμένου. Μπορείτε να χρησιμοποιήσετε κάποιο συντάκτη που σας παρέχει το γραφικό περιβάλλον σας ή ένα εύκολο συντάκτη για την κονσόλα (π.χ. τον `ee`).

```
% ee ~/.cshrc
```

Προσθέστε την παρακάτω γραμμή META τη λίστα των άλλων εντολών.

```
set history = 200
```

Αποθηκεύστε το αρχείο και αναγκάστε το κέλυφος να ξαναδιαβάσει το αρχείο `.cshrc` χρησιμοποιώντας την εντολή του κελύφους `source`.

```
% source .cshrc
```

Ελέγξτε τη σωστή λειτουργία, πληκτρολογώντας:

```
% echo $history
```

8.6 Ρύθμιση του path

Όταν πληκτρολογείτε μια εντολή, η μεταβλητή `path` (ή `PATH`) ορίζει σε ποιους καταλόγους θα ψάξει το κέλυφος για να την βρει. Αν το σύστημα επιστρέψει ένα μήνυμα του τύπου "command: Command not found", σημαίνει είτε ότι η εντολή δεν υπάρχει καθόλου στο σύστημα, ή απλά ότι δεν υπάρχει στο `path` σας.

Για παράδειγμα, για να εκτελέσετε το πρόγραμμα `units` πρέπει είτε να ορίσετε απευθείας τη πλήρη διαδρομή προς αυτός (`~/units174/bin/units`) ή να έχετε προσθέσει στο `path` σας τον κατάλογο `~/units174/bin`.

Μπορείτε να τον προσθέσετε στο τέλος του υπάρχοντος `path` (το οποίο αντιπροσωπεύεται από το `$path`) χρησιμοποιώντας την παρακάτω εντολή:

```
% set path = ($path ~/units174/bin)
```

Ελέγξτε τη λειτουργία του παραπάνω, προσπαθώντας να εκτελέσετε το `units` ενώ βρίσκεστε σε οποιοδήποτε κατάλογο.

```
% cd  
% units
```

Για να κάνετε αυτή την αλλαγή στο `path` ΜΟΝΙΜΗ, προσθέστε την παρακάτω γραμμή στο αρχείο `.cshrc` META τη λίστα των υπόλοιπων εντολών.

```
set path = ($path ~/units174/bin)
```