# Μικρή εισαγωγή σε εντολές του Unix.

Το κείμενο αυτό έχει σαν σκοπό να σας βοηθήσει να μάθετε μερικές από τις βασικές εντολές του Unix. Θα πρέπει να το μελετήσετε καθισμένοι μπροστά από ένα Unix τερματικό. Διαβάστε μία παράγραφο, και στην συνέχεια δοκιμάστε την εντολή. Δούλεψε όπως θα έπρεπε; Αν όχι κυτάξτε προσεκτικά τι γράψετε. Το πιθανότερο ήταν να υπάρχει ένα λάθος στην γραφή της συγκεκριμένης εντολής.

Ας ξεκινήσουμε με μία απλή εντολή: θέλουμε ο υπολογιστής να μας πει την σημερινή μέρα. Να πως το κάνουμε:

```
[kontogar@Lykourgos kontogar]$ date
Sat Oct 26 17:59:16 EEST 2002
```

Δοκιμάστε την παραπάνω εντολή να δείτε πως δουλευεί και τι αποτελέσματα δίνει. Στο Unix είναι σημαντικό να έχετε υπόψην σας ότι παίζουν σημαντικό ρόλο αν γράφετε με κεφαλαία ή μικρά. Έτσι για παράδειγμα οι γραφές Unix,UnIx και unix δεν είναι ισοδύναμες! Δοκιμάστε να δώσετε στο τερματικό σας τις εντολές cal, cal 2002,whoami,finger Οι πρώτες δύο δίνουν το ημερολόγιο του τρέχοντος μήνα και έτους αντίστιχα, η τρίτη επιστρέφει το login name και η τελευταία μας δείχνει τους χρήστες που είναι συνδεμένοι την στιγμή αυτή με τον υπολογιστή.

Ας δούμε τώρα την εντολή echo Αν δώσω

```
echo PATH
PATH
```

λέω στον υπολογιστή να τυπώσει το κορδόνι με όνομα PATH. Αν αντιθέτως πληκτρολογήσω

```
echo $PATH
/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/bin:/usr/local/bin:
/usr/local/sbin:/usr/bin:/usr/ucb:/home/lectures/kontogar/bin
```

Λέω στον υπολογιστή να τυπώσει την μεταβλητή συστήματος PATH. Στην μεταβλητή αυτή είναι αποθηκευμένα τα μέρη του συστήματος αρχείων στα οποία βρίσκονται προγράμμτα τα οποία μπορείτε να εκτελέστε.

Αν θέλετε να προσθέσετε άλλο ένα μέρος του συστήματος αρχείων σας στην μεταβλητή PATH (για παράδειγμα το /usr/local/bin) θα πρέπει να το κάνετε δίνοντας

```
export PATH=$PATH:/usr/local/bin
```

Ας δούμε τώρα τον χειρισμό αρχείων: Το σύστημα αρχείων μιάζει με ένα μεγάλο κουτί στο οποίο βρίσκονται μεγάλα κουτιά directories μέσα στα οποία βρίσκονται μικρότερα κουτιά και αρχεία. Με την εντολή pwd μπορείτε να δείτε το όνομα του κουτιού στο οποίο βρίσκεστε

```
pwd
/home/kontogar
```

και το οποίο δεν είναι άλλο από το home directory σας στο οποίο θα γράφετε τα αρχεία σας. Προκειμενου να είναι αυτά τακτοποιημένα τα χωρίζουμε σε «κουτιά» τα οποία θα τα ονομάζουμε direcories. Για να δημιουργήσουμε ένα κουτί χρησιμοποιούμε την εντολή

```
mkdir pari-programs
```

Έτσι αν τώρα δώσουμε την εντολή ls η οποία μας δείχνει τα περιεχόμενα ενός κουτιού, τότε ανάμεσα στα αρχεία μας θα δούμε και το καινούριο directory με όνομα pari-programs. Με την εντολή cd pari-programs ανοίγουμε το directory αυτό, ενώ η εντολή cd .. μας ανεβάζει ένα direcory προς τα πάνω.

Ένα πρόγραμμα μπορεί να πάρει αρκετό χρόνο να μας δώσει αποτελέσμτα ανάλογα με την δυσκολία του την ταχύτητα του συστήματος και τον φόρτο του συστήματος μας. Δεν μπορούμε να κάτσουμε με τις ώρες μπροστά στο τερματικό μας ή στον υπολογιστή να το περιμένουμε. Έτσι αν θέλουμε να παραγοντοποιήσουμε τον πρώτο αριθμό $2^{100000} - 1$ θα δημιουργήσουμε ένα αρχείο με όνομα factor.in που θα έχει μόνο μια γραμμή την

```
factor(2^(100000)-1)
```

και θα τρέξουμε στην γραμμή εντολών

```
gp < factor.in > factor.out &
```

Το παραπάνω έχει το εξής νόημα: τρέξε το πρόγραμμα gp αλλά μην περιμένεις να σου πληκτρολογήσει ο χρήστης εντολές. Αυτές θα τις πάρεις από το αρχείο factor.in. Επίσης δεν θα εμφανίσεις τα αποτελέσματα στην οθόνη αλλά θα τα γράψεις στο αρχείο factor.out. Τέλος ο χαρακτήρας & λέει στον υπολογιστή να τρέξει το προγραμμά στο παρασκήνιο(backround. Με αυτό τον τρόπο μπορούμε να βάλουμε τον υπολογιστή να υπολογίζει να κλείσουμε την σύνδεση μας με αυτόν και να πάμε την άλλη μέρα να πάρουμε τα αποτελέσματα.

Θα δούμε τώρα μερικές βασικές εντολές που αφορούν το έλεγχο των εργασιών στο unix. Ένα σύστημα unix εκτελεί πολλές εργασίες ταυτόχρονα και εξηπηρετεί πολλούς χρήστες. Έτσι για παράδειμα η εντολή [1]

[1] στο linux αυτή η εντολή συντάσεται ώς ps aux

```
ps -ef
```

θα μας δώσει ένα μακρύ κατάλογο των ενολών που το σύστημα εκτελεί. Για να δούμε τις εντολές που τρέχουμε εμείς πρέπει το αποτέλεσμα της εντολής ps να το φιλτράτουμε μέσω της εντολής grep. Έτσι δίνοντας

```
ps -ef | grep kontogar
```

Βλέπουμε όλες τις εντολές τις οποίες τρέχει ο χρήστης με όνομα kontogar. Στον κατάλογο των εργασιών δίπλα απο το όνομα του χρήστη βλέπουμε κάποιους αριθμούς. Ο πρώτος από αυτούς είναι ο κωδικός της εργασιάς. Αν θέλουμε να τερματίσουμε μία εργασία του συστήματος γιατί κάτι δεν πάει καλά με αυτήν, μπορούμε να το κάνουμε δίνοντας

```
kill 21479
```

όπου το 21479 είναι ο αριθμός της συγκεκριμένης εργασίας.

Τρέξτε μια διαδικασία στο παρασκήνιο, για παράδειγμα την προηγούμενη παραγοντοποιήση και δοκιμάστε να δείτε αν εκτελείτε κάνοντας χρήση της ps. Αυτός είναι και ο τρόπος που θα διαπιστώσετε την επόμενη μέρα αν ο μακρύς υπολογισμός που ξεκινήσατε το προηγούμενο βράδυ έχει ολοκληρωθεί ή όχι.

Θα τελείωσουμε αυτή την μικρή αναφορά σε εντολές του unix αναφέροντας την εντολή man. Με αυτή την εντολή μπορούμε να διαβάσουμε τα εγχειρίδια των εντολών του συστήματος. Δώστε λοιπόν

```
man ls
```

για να διαβάσετε περισσότερες πληροφορίες σχετικά με την εντολή ls ή για οποιαδήποτε άλλη εντολή αναφέραμε εδώ.

## Compiling

Ένα πρόγραμμα σε C αποτελείται από τον πηγαίο κώδικα (source code) τον οποίο πρέπει να τον περάσουμε από τον compiler προκειμένου να παράγουμε το εκτελέσιμο αρχείο, το οποίο είναι κατανοητό από την μηχανή.

Ένας διαδεδομένος compiler για το unix είναι ο gcc. Αυτός είναι το προϊόν πολυετούς έρευνας του GNU-project το οποίο ξεκίνησε πρίν από 20 χρόνια ο R.Stallman.

Ας υποθέσουμε ότι έχουμε ένα πηγαίο αρχείο με όνομα c-code.c. Εκτελόντας την εντολή

```
gcc c-code.c
```

ενεργοποιούμε τον gcc compiler ο οποίος παράγει το εκτελέσιμο αρχείο a.out, το οποίο το εκτελούμε με την εντολή

```
./a.out
```

Αν θέλαμε το παραγόμενο εκτελέσιμο αρχείο να είχε όνομα runme αντί για το a.out θα έπρεπε να εκτελούσαμε τον gcc ως εξής:

```
gcc c-code.c -o runme
```

**Ασκήσεις**

1. Γράψτε από ένα βιβλίο *C* το περίφημο «καλήμερα κόσμε» πρόγραμμα μεταγλωτίστε το με τον gcc και εκτελέστε το παραγόμενο πρόγραμμα. Κάνοντας χρηση του τελεστή ανακατεύθυνσης > εκτελέστε το πρόγραμμα ώστε να εξάγει το αποτέλεσμα μέσα σε ένα αρχείο και ελένξτε με ένα text editor τα περιεχόμενα του αρχείου αυτού.

2. Στο unix η εντολή man δίνει πληροφορίες για εντολές του συστήματος. Δίνοντας man gcc δείτε το εγχειρίδιο χρήσης του gcc. Η man δίνει πληροφορίες και για τις εντολές της C. Δώστε man printf για να δείτε την χρήση της εντολής printf

Στο μάθημα θα χρειαστεί να χρησημοποιήσουμε βιβλιοθήκες της C και συγκεκριμένα την βιβλιοθήκη OpenGL. Για να μπορέσει ο gcc να χρησιμοποιήσει εξωτερικές βιβλιοθήκες θα πρέπει να του πούμε μέσω παραμέτρων την τοποθεσία των βιβλιοθηκών. Θα πούμε περισσότερες λεπτομέριες στην συνέχεια του μαθήματος, προς το παρόν θα μάθουμε να χρησιμοπούμε Makefiles τα οποία στην αρχή θα σας δίνονται έτοιμα.

Δώστε την εντολή

```
cp -r /home/lecturers/kontogar/CompGraph/CubeTEST .
```

για να αντιγράψετε τα περιεχόμενα του φακέλου CubeTEST από την προσωπική μου περιοχή στην δικιά σας και στην συνεχεία δώστε cd CubeTEST για να μπείτε στο κατάλογο CubeTEST. Εξετάστε τα περιεχόμενα των δύο αρχείων που βρίσκονται μέσα στον κατάλογο αυτό και τέλος δώστε make για να κάνετε compile το πρώτο σας OpenGL πρόγραμμα.