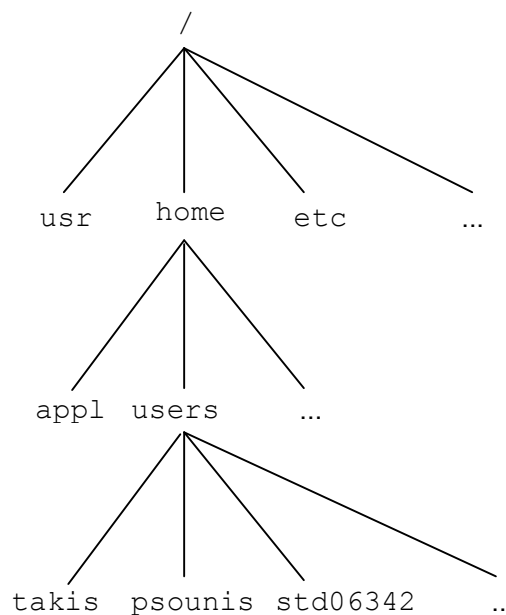


## ΕΡΓΑΣΤΗΡΙΟ 2: Unix Tutorial

Σκοπός του εργαστηρίου αυτού είναι να έλθουμε σε επαφή με βασικές εντολές του Unix και την μεταγλώττιση προγραμμάτων C (με χρήση του μεταγλωττιστή `gcc`).

Για να οργανώσουμε τα αρχεία μας, τα τοποθετούμε σε καταλόγους. Κάθε κατάλογος μπορεί να περιέχει είτε αρχεία, είτε άλλους καταλόγους που τους λέμε και υποκαταλόγους του αρχικού. Όλα ξεκινούν από έναν αρχικό κατάλογο που λέγεται και ρίζα `/`. Για παράδειγμα, η δόμηση του συστήματος αρχείων στο μηχάνημα που έχετε συνδεθεί είναι ως εξής:



Όταν εργαζόμαστε σε ένα σύστημα, βρισκόμαστε σε έναν κατάλογο που θεωρείται ο τρέχων κατάλογος. Για παράδειγμα όταν συνδεόμαστε σε ένα μηχάνημα μέσω του Putty, αυτόματα βρισκόμαστε στον αρχικό μας κατάλογο: `/home/users/stdXXYY`. Για να αναφερθούμε τώρα σε κάποιον άλλο κατάλογο έχουμε δύο τρόπους:

- Είτε με απόλυτο μονοπάτι, ξεκινώντας από την ρίζα, για παράδειγμα: `/home/users/std06342`
- Είτε με σχετικό μονοπάτι, σε σχέση με τον τρέχοντα κατάλογο. Για παράδειγμα, αν ο τρέχων κατάλογος είναι ο `/home`, τότε σχετικό μονοπάτι σε αυτόν είναι το `users/std06342`. Επίσης σε σχέση με τον τρέχοντα κατάλογο, υπάρχουν και οι ειδικοί καταλόγοι:
  - `..` : Είναι συνώνυμο του γονικού καταλόγου (Για παράδειγμα, αν τρέχων κατάλογος είναι ο `/home/users/std06342`, τότε με `..` αναφερόμαστε στον κατάλογο `/home/users`).
  - `.` : Είναι συνώνυμο του τρέχοντος καταλόγου

Εδώ να τονίσουμε ότι τα ονόματα που χρησιμοποιούνται στους καταλόγους και στα αρχεία είναι *case-sensitive*, δηλαδή το όνομα καταλόγου «`Psounis`» είναι διαφορετικό από το όνομα καταλόγου «`psoUnis`». Αυτό είναι ένα γενικότερο χαρακτηριστικό του Unix, βέβαια. Όλα τα ονόματα που χρησιμοποιούμε, όχι μόνο για καταλόγους και αρχεία, αλλά και για εντολές του λειτουργικού ή εντολές μέσα σε διάφορα βοηθητικά προγράμματα (π.χ. κειμενογράφους) είναι *case-sensitive*.

Θα μελετήσουμε τώρα ένα πακέτο εντολών του Unix, που θα μας φανεί χρήσιμο για να διαχειριστούμε τα αρχεία μας, να γράψουμε προγράμματα C, να τα μεταγλωττίσουμε και να τα εκτελέσουμε.

Για τον λόγο αυτό θα διαβάζουμε την σύνταξη των εντολών<sup>1</sup> και θα τις χρησιμοποιούμε αναλόγως με το ζητούμενο στόχο. Ακολουθήστε βήμα-βήμα το παρακάτω σενάριο:

1. Εμφανίστε τα περιεχόμενα του καταλόγου `/usr/include`.

Η εντολή `ls` εμφανίζει τα περιεχόμενα καταλόγων.

Σύνταξη:

- `ls` : Εμφανίζει τα περιεχόμενα του τρέχοντος καταλόγου
- `ls ονομα_καταλόγου`: Εμφανίζει τα περιεχόμενα του καταλόγου με όνομα `ονομα_καταλόγου`, που έχει δοθεί είτε απόλυτα (από `root`) είτε σχετικά (από τρέχοντα κατάλογο)

Χρήσιμες επιλογές:

- `-a` : Εμφανίζει τα κρυφά αρχεία (αρχεία που αρχίζουν από τελεία)
- `-l` : Εμφανίζει εκτεταμένες πληροφορίες για τα περιεχόμενα του καταλόγου (δικαιώματα, μέγεθος, ημερομηνία τροποποίησης κ.λ.π.)

Χρήση μεταχαρακτήρων (ταιριάζουν τμήμα ονόματος αρχείου ή καταλόγου με τα διαθέσιμα)

- `*` : Ταιριάζει με κανέναν ή περισσότερους χαρακτήρες
- `?` : Ταιριάζει με ακριβώς έναν χαρακτήρα

Παραδείγματα χρήσης μεταχαρακτήρων:

- `ls *t` : Εμφάνισε τα αρχεία ή περιεχόμενα καταλόγων που το όνομα τους τελειώνει σε "t".
- `ls ??z8` : Εμφάνισε τα αρχεία ή περιεχόμενα καταλόγων που το όνομά τους έχει 4 γράμματα και τα δύο τελευταία είναι "z8"

2. Εμφανίστε το πλήρες όνομα του τρέχοντος καταλόγου.

Η εντολή `pwd` εμφανίζει το πλήρες όνομα του τρέχοντος καταλόγου.

3. Δημιουργήστε κάτω από τον τρέχοντα κατάλογο ένα νέο κατάλογο με όνομα `myWork`.

Η εντολή `mkdir` δημιουργεί τον κατάλογο που της δίνουμε σαν παράμετρο.

Συνταξη:

`mkdir ονομα_καταλόγου`

---

<sup>1</sup> Ένα ιδιαίτερα χρήσιμο κείμενο, με περαιτέρω τρόπους σύνταξης εντολών, πέρα από αυτές που θα αξιοποιήσουμε σήμερα, μπορούν να βρεθούν εδώ: <http://cgi.di.uoa.gr/~ip/Unix.pdf>

4. Μεταβείτε σε αυτόν τον νέο κατάλογο.

Η εντολή `cd` χρησιμοποιείται για την μετάβαση σε έναν κατάλογο.

Σύνταξη:

- `cd cat_name`: Μεταβαίνει στον κατάλογο `cat_name` κάτω από τον τρέχοντα κατάλογο
- `cd cat_path`: Μεταβαίνει στον κατάλογο που ορίζεται από το απόλυτο ή σχετικό μονοπάτι `cat_path`

Ειδική χρήση:

- `cd`: Μεταβαίνει στον αρχικό μας κατάλογο

5. Αντιγράψτε στον κατάλογο αυτό το πηγαίο αρχείο `~iphw/samples/mysin.c`

Η εντολή `cp` χρησιμοποιείται για την αντιγραφή αρχείων ή καταλόγων

Σύνταξη:

```
cp όνομα νέο_όνομα  
cp όνομα κατάλογος_προορισμού
```

Επιλογές:

- `-i`: Εμφανίζει ένα μήνυμα ελέγχου, όταν το αρχείο υπάρχει ήδη στον προορισμό και πρέπει να αντικατασταθεί.
- `-R`: Αντιγράφει τον κατάλογο `όνομα` μαζί με όλα τα περιεχόμενα του (αρχεία, υποκαταλόγους) σε κατάλογο με όνομα `νέο_όνομα`.

6. Δημιουργήστε το εκτελέσιμο αρχείο `mysin` από το πηγαίο αρχείο που μόλις αντιγράψατε.

Ο μεταγλωττιστής `gcc` χρησιμοποιείται για τη μεταγλώττιση πηγαίων αρχείων σε εκτελέσιμα.

Σύνταξη:

```
gcc -o εκτελέσιμο πηγαίο_αρχείο
```

Επιλογές:

- `-lm`: Ενσωμάτωση της μαθηματικής βιβλιοθήκης

7. Τρέξτε το εκτελέσιμο αυτό.

8. Αντιγράψτε στον τρέχοντα κατάλογο το πηγαίο αρχείο `~iphw/samples/mymain.c`

9. Αντιγράψτε στον τρέχοντα κατάλογο το αντικειμενικό αρχείο `~iphw/samples/myfunct.o`

10. Μεταγλωττίστε το πηγαίο αρχείο `mymain.c` στο αντίστοιχο αντικειμενικό αρχείο.

Ο μεταγλωττιστής `gcc` χρησιμοποιείται για την μεταγλώττιση πηγαίων αρχείων σε αντικειμενικά

Σύνταξη:

```
gcc -c πηγαίο_αρχείο.c : Παράγει το αντικειμενικό αρχείο με  
όνομα πηγαίο_αρχείο.o
```

11. Συνδέστε το αντικειμενικό αρχείο που κατασκευάσατε και το αντικειμενικό αρχείο `myfunct.o` δημιουργώντας το εκτελέσιμο αρχείο `myprog`.

Ο μεταγλωττιστής `gcc` χρησιμοποιείται για την σύνδεση αντικειμενικών αρχείων σε ένα εκτελέσιμο

Σύνταξη:

```
gcc -o εκτελέσιμο αντικ_αρχείο1 αντικ_αρχείο2 ....
```

12. Εκτελέστε το `myprog` και ακολουθήστε τις οδηγίες που θα εκτυπωθούν.

13. Διαγράψτε όλα τα αρχεία που δημιουργήσατε καθώς και τον κατάλογο `myWork`.

Η εντολή `rm` χρησιμοποιείται για τη διαγραφή αρχείων ή καταλόγων

Σύνταξη για διαγραφή αρχείων:

```
rm όνομα_αρχείου (χρήση και μεταχαρακτήρων)
```

Σύνταξη για διαγραφή καταλόγων (μαζί με όλα τα περιεχόμενά τους):

```
rm -r όνομα_καταλόγου
```

Επιλογές:

- `-i`: Εμφανίζει μήνυμα επιβεβαίωσης διαγραφής του αρχείου ή του καταλόγου.

14. Αντιγράψτε στον τρέχοντα κατάλογο το αρχείο `stdio.h` από τον κατάλογο `/usr/include`

15. Μετονομάστε το αρχείο που αντιγράψατε σε `Mystdio.h`

Η εντολή `mv` χρησιμοποιείται για τη μετονομασία ή μετακίνηση αρχείων ή καταλόγων

Σύνταξη:

```
mv αρχικό_όνομα τελικό_όνομα
```

```
mv αρχικό_όνομα καταλογος_προορισμού
```

Επιλογές:

- `-i`: Εμφανίζει μήνυμα ελέγχου εφόσον υπάρχει ήδη το αρχείο στον προορισμό και πρέπει να αντικατασταθεί.

16. Εμφανίστε στην οθόνη τα περιεχόμενα του αρχείου αυτού.

Η εντολή `cat` χρησιμοποιείται για την εμφάνιση των περιεχομένων αρχείων

Σύνταξη:

```
cat όνομα_αρχείου
```

Επιλογές:

- `-n` : Εμφανίζει αρίθμηση στις γραμμές του αρχείου

17. Δημιουργήστε στον τρέχοντα κατάλογο ένα κενό αρχείο με όνομα `.my_file` (προσέξτε την τελεία στην αρχή του ονόματος).

Η εντολή `touch` χρησιμοποιείται για τη δημιουργία κενών αρχείων.

Σύνταξη:

```
touch όνομα_αρχείου
```

18. Εμφανίστε τα δικαιώματα προστασίας του αρχείου αυτού και ό,τι άλλες χρήσιμες πληροφορίες μας δίνει το λειτουργικό σύστημα για αυτό.

19. Αλλάξτε τα δικαιώματα προστασίας έτσι ώστε εσείς (ο ιδιοκτήτης του) να έχει όλα τα δικαιώματα προστασίας (ανάγνωσης, εγγραφής και εκτέλεσης), τα μέλη της ομάδας στην οποία ανήκει το αρχείο να έχουν δικαιώματα ανάγνωσης και εκτέλεσης και οι υπόλοιποι να έχουν μόνο δικαίωμα ανάγνωσης.

Η εντολή `chmod` χρησιμοποιείται για τον καθορισμό των δικαιωμάτων προστασίας ενός αρχείου

Σύνταξη:

```
chmod XYZ όνομα_αρχείου
```

όπου X, Y, Z είναι οκταδικοί αριθμοί από το 0 έως το 7 που καθορίζουν τα δικαιώματα του ιδιοκτήτη, της ομάδας και των υπολοίπων αντίστοιχα.

Τα δικαιώματα προκύπτουν αν γράψουμε τον αριθμό σε δυαδική μορφή, όπου το 1<sup>ο</sup> bit καθορίζει το δικαίωμα ανάγνωσης, το 2<sup>ο</sup> bit το δικαίωμα εγγραφής και το 3<sup>ο</sup> bit το δικαίωμα εκτέλεσης.

20. Εμφανίστε πάλι τα δικαιώματα προστασίας του αρχείου `.my_file` (και τις άλλες πληροφορίες).

21. Εμφανίστε όλα τα αρχεία του τρέχοντος καταλόγου, μαζί με τα δικαιώματα προστασίας (και τις άλλες πληροφορίες), μη συμπεριλαμβανομένων των αρχείων που το όνομά τους αρχίζει από τελεία.

22. Το ίδιο με το προηγούμενο, αλλά να συμπεριληφθούν και τα αρχεία που το όνομά τους αρχίζει από τελεία.

23. Μέσω της εντολής `man` ενημερωθείτε σχετικά με τις δυνατότητες της εντολής `grep`.

Η εντολή `man` εμφανίζει πληροφορίες για την σύνταξη εντολών.

Σύνταξη:  
`man όνομα_εντολής`

24. Χρησιμοποιείστε την εντολή `grep` για να βρείτε μέσα στο αρχείο `/usr/include/stdlib.h` τις γραμμές που περιλαμβάνουν το κείμενο `size`.

25. Επαναλάβετε την προηγούμενη εντολή αλλά να κάνετε ανακατεύθυνση της εξόδου της στο αρχείο `grepout.txt`

Η ανακατεύθυνση εξόδου σε αρχείο γίνεται με την ακόλουθη σύνταξη:

Εντολή > ονομα\_αρχείου

26. Συνδυάστε με κάποιον τρόπο τις εντολές `ls` και `grep` για να βρείτε στον τρέχοντα κατάλογο αρχεία στα οποία ο ιδιοκτήτης έχει δικαιώματα ανάγνωσης και εγγραφής, αλλά όχι εκτέλεσης και όλοι οι υπόλοιποι δεν έχουν κανένα δικαίωμα.

Η σωλήνωση της εξόδου ενός προγράμματος στην είσοδο ενός άλλου, γίνεται με την ακόλουθη σύνταξη:

Πρόγραμμα | Πρόγραμμα

27. Αντιγράψτε στον τρέχοντα κατάλογο το πηγαίο αρχείο `~iphw/samples/capitalize.c` δημιουργήστε το αντίστοιχο εκτελέσιμο και τρέξτε το με ανακατεύθυνση της εισόδου από το ίδιο το πηγαίο αρχείο `capitalize.c` στέλνοντας την έξοδο (πάλι με ανακατεύθυνση) στο αρχείο `CAPITALIZE.c`

Η ανακατεύθυνση εισόδου και εξόδου γίνεται με την ακόλουθη σύνταξη:

Εντολή < Αρχείο\_για\_είσοδο > Αρχείο\_για\_έξοδο

28. Δοκιμάστε να μεταγλωττίσετε το αρχείο `CAPITALIZE.c`. Μην ανησυχείτε όμως αν δεν μπορεί να μεταγλωττιστεί (γιατί άραγε;)