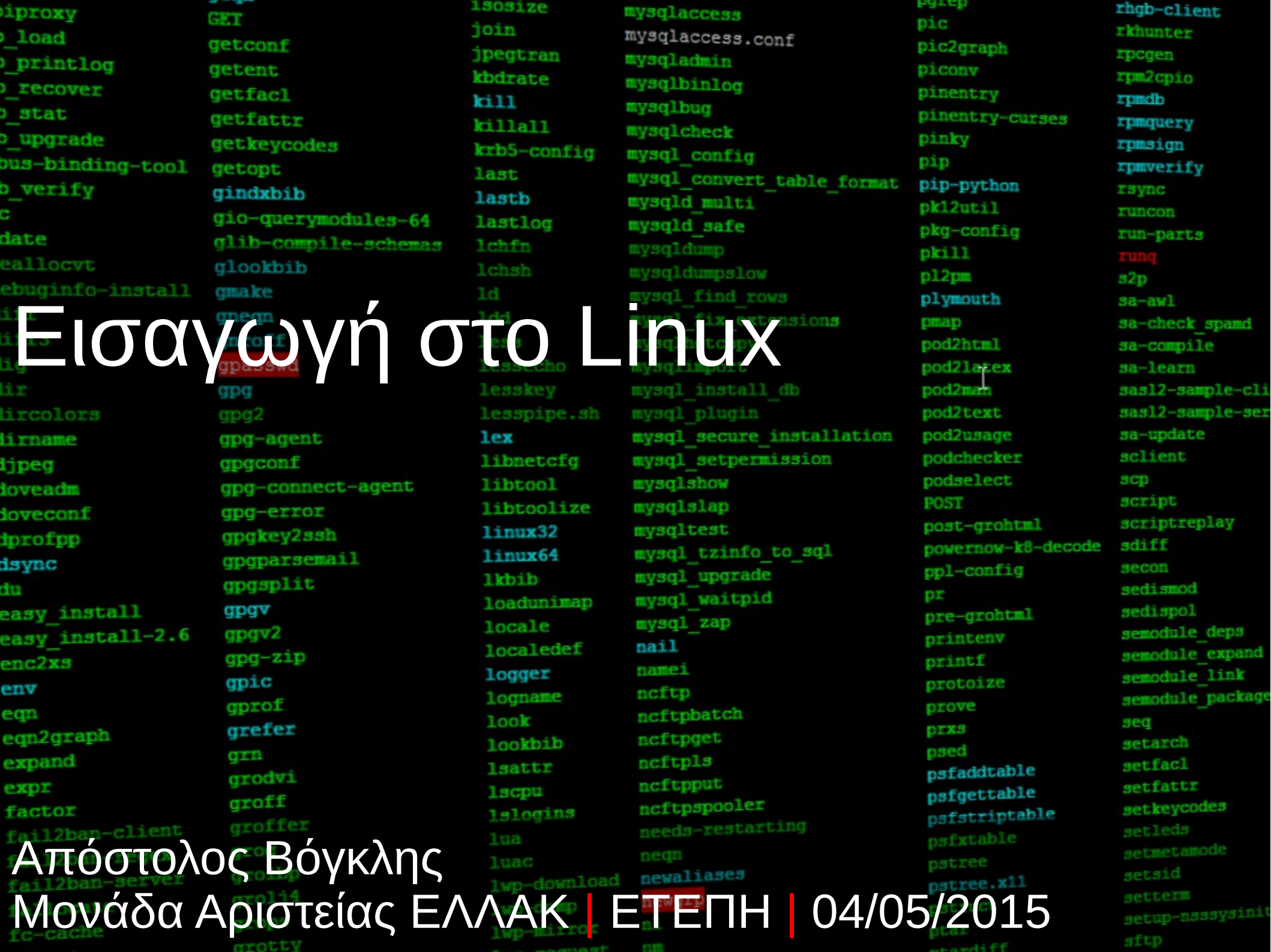




Εισαγωγή στο Linux

Απόστολος Βόγκλης

Μονάδα Αριστείας ΕΛΛΑΚ | ΕΤΕΠΗ | 04/05/2015

Τι είναι ένα Λειτουργικό Σύστημα

- Το λογισμικό πάνω στο οποίο στηρίζεται η διαχείριση όλων των διαδικασιών ενός Η/Υ.
- Χωρίς αυτό οι Η/Υ είναι πρακτικά αδύνατο να χρησιμοποιηθούν.
- Ευρέως διαδεδομένα λειτουργικά συστήματα: **MS-DOS, WINDOWS, MacOS, Unix, Linux.**
- Ορισμένα παλαιότερα λειτουργικά συστήματα: **NOS, COS, VMS, VM-CMS, PRIMOS** κλπ.

Χρηστές (users)

Κέλυφος (Shell) ή Γραφικό περιβάλλον (GUI)

**Εφαρμογές/Προγράμματα/
μεταφραστές
(applications/Software/
Compilers)**

**Εφαρμογές
διαχείρισης του
συστήματος/System
Utilities**

**Βιβλιοθήκες του συστήματος/
System Calls Libraries**

**Ο Πυρήνας του λειτουργικού
Συστήματος KERNEL**

**ΥΛΙΚΟ/HARDWARE
CPU RAM HDD NICS**

Η Ιστορία του Unix

- Δημιουργήθηκε από τον Kenneth Thompson το 1969 στα Bell Labs.
- Ήταν γραμμένο σε Assembly του Η/Υ PDP-7 της εταιρείας Digital (DEC).
- Υποστήριζε μόνο έναν χρήστη.
- Ξαναγράφηκε το 1971 σε Assembly του PDP-11 (DEC).

Η Ιστορία του Unix ...

- Ο Thompson το 1970 κατασκεύασε την γλώσσα B, ώστε να μην χρειάζεται να γράφεται όλο το σύστημα εκ νέου σε Assembly κάθε φορά που άλλαζε πλατφόρμα.
- Η γλώσσα B τελικά αποδείχτηκε ακατάλληλη.
- Το 1973 το Unix ξαναγράφηκε εξ ολοκλήρου στην γλώσσα C που δημιούργησε ο Dennis Ritchie στα Bell Labs και τότε άρχισε να υποστηρίζει πολλούς χρήστες.

Η Ιστορία του Unix ...

- Αναπτύχθηκαν διάφορες εκδόσεις του Unix
- Στο University of California at Berkeley η έκδοση BSD.
- Από την Microsoft το προϊόν Xenix που είχε επιπλέον, εμπορικές εφαρμογές.
- Το 1983 από την AT&T η έκδοση System-V
- Από το 1984 άρχισε μια προσπάθεια για την ανάδειξη μιάς Standard έκδοσης.

Εκδόσεις του Unix

- **Solaris** και **SunOS** της **Sun Microsystems**
- **AIX** της **IBM**
- **IRIX** της **Silicon Graphics**
- **HP/UX** της **Hewlett-Packard**
- **AUX** της **Apple**
- **Xenix** της **Microsoft**
- **Linux** (διατίθεται ελεύθερα)
- **Free BSD** από το **UC at Berkeley** (διατίθεται ελεύθερα)

Η Ιστορία του Linux ...

- 1991 : Ο Linus Torvalds ξεκίνησε ένα προσωπικό project για ένα νέο λογισμικό.
- Ο πυρήνας του Linux αποτελούνταν από μερικά αρχεία C.
- 2015 : Έκδοση 3.18. >18m γραμμές κώδικα κάτω από την GNU GPL
- Σταθερότητα και ασφάλεια.

Το λειτουργικό σύστημα Linux

- Το Linux υποστηρίζει ταυτόχρονα πολλούς χρήστες. (Multi-User, Time Sharing).
- Κάθε χρήστης μπορεί να εκτελεί ταυτόχρονα πολλά προγράμματα. (Multi-Tasking).
- Κάθε χρήστης έχει ένα «λογαριασμό».
- Συνήθως ο λογαριασμός έχει το όνομα του χρήστη.
- Κάθε λογαριασμός καθορίζει ένα «χώρο» όπου ο χρήστης διατηρεί τα αρχεία του.

Σχετικά με το Linux ...

- Έχει δοθεί βαρύτητα στην ασφάλεια και προστασία των δεδομένων.
- Το **Linux** έχει μεταφερθεί σε πολλές διαφορετικές μηχανές.
- Η «μεταφερσιμότητα» οφείλεται στο γεγονός ότι το **Linux** είναι γραμμένο κατά κύριο λόγο στην υψηλού επιπέδου γλώσσα **C**.
- Μόνο ένα μικρό μέρος είναι γραμμένο σε γλώσσα μηχανής (Assembly).
- Υπάρχει πληθώρα από διαφορετικά παραθυρικά περιβάλλοντα (KDE, GNOME κα)

Ορολογία

- **Σύστημα:** Εννοούμε το λειτουργικό σύστημα. Η έκφραση «μπήκα στο σύστημα» σημαίνει ότι έχω επιτυχώς ταυτοποιηθεί και μπορώ να χρησιμοποιήσω τους υπολογιστικούς πόρους που μου διατίθενται.
- **Τερματικό (terminal):** Ένα πρόγραμμα που εκτελείται σε παραθυρικό περιβάλλον και δίνει τη δυνατότητα στο χρήστη να αλληλεπιδράσει με το Σύστημα, μέσω του κελύφους, χρησιμοποιώντας το πληκτρολόγιο.
- **Κέλυφος ή φλοιός (shell):** Πρόγραμμα του λειτουργικού συστήματος που επιτρέπει στο χρήστη να αλληλεπιδράσει με εντολές από το πληκτρολόγιο.
- **Παραθυρικό περιβάλλον:** Πρόγραμμα που εκτελείται και επιτρέπει στο χρήστη να αλληλεπιδράσει με το ποντίκι (πχ. X-Windows)
- **Αρχείο:** Ένα αρχείο υπολογιστή είναι ένα σύνολο από πληροφορίες, δεδομένα ή και ένας πόρος.
 - Ένα αρχείο είναι διατηρητέο με την έννοια ότι ακόμα και όταν τερματίσουν να εκτελούνται τα προγράμματα που το δημιούργησαν ή το χρησιμοποιούν αυτό θα συνεχίσει να υπάρχει
 - Σε κάθε αρχείο υπολογιστή ενός υπολογιστή δίνεται ένα όνομα ώστε να ξεχωρίζει από όλα τα άλλα αρχεία.

Ορολογία

- **Σύστημα αρχείων:** Μέθοδος για την αποθήκευση και την οργάνωση των δεδομένων των αρχείων.
- **Κατάλογος:** ένα ειδικό αρχείο το οποίο περιέχει πληροφορίες για άλλα αρχεία και καταλόγους. Ένας κατάλογος που βρίσκεται καταχωρημένος σε έναν κατάλογο ονομάζεται υποκατάλογός του. Έτσι σχηματίζεται μια ιεραρχία καταλόγων, μια δενδροειδής δομή.
- **Διεργασία (process):** Σε έναν υπολογιστή είναι το στιγμιότυπο ενός προγράμματος που εκτελείται.
- **Χρήστης:**

Ενότητα 1: Πρόσβαση στο Linux

Λίστα εντολών

- exit – έξοδος από σύστημα ή τερματικό
- logout – έξοδος από το σύστημα
- passwd – Αλλαγή της κωδικής λέξης
- rlogin – απομακρυσμένη σύνδεση σε σύστημα Linux (μέσω δικτύου)
- slogin – κρυπτογραφημένη έκδοση της rlogin
- ssh – κρυπτογραφημένη δημιουργία απομακρυσμένου τερματικού
- telnet – είσοδος σε απομακρυσμένο σύστημα μέσω της υπηρεσίας telnet

Η διαδικασία login

- Το Linux ζητά το όνομα του «**Λογαριασμού**». Ενδεικτικά τυπώνει στην οθόνη την λέξη:

login:

- Ο χρήστης εδώ πρέπει να πληκτρολογήσει το όνομα του λογαριασμού του πχ: **login:** και στην συνέχεια να "πατήσει" το πλήκτρο ENTER

login ... συνέχεια

- Στην συνέχεια το Linux ζητά μια «λέξη-κλειδί» (password) από τον χρήστη. Ενδεικτικά τυπώνει στην οθόνη την λέξη:

password:

- Ο χρήστης τώρα πρέπει να πληκτρολογήσει την λέξη-κλειδί η οποία δεν εμφανίζεται στην οθόνη για λόγους «ασφαλείας» και *ENTER*

“login” ...

- Εάν όλα έχουν εισαχθεί σωστά, ο χρήστης συνδέεται και μπορεί να αρχίσει να εργάζεται.
- Εάν κάτι δεν έχει πάει καλά, το **Linux** ξεκινά ξανά την διαδικασία **login**.

Σύνδεση νέου χρήστη

- Υπάρχουν διάφορες παραλλαγές της διαδικασίας **login** την πρώτη φορά που χρησιμοποιεί κανείς τον λογαριασμό του.
- Αυτό εξαρτάται από το τι επιλογές έχει κάνει ο διαχειριστής του συστήματος. (**Administrator**)
- Π.χ. είναι δυνατόν να μη ζητηθεί password, αλλά να πρέπει να καθοριστεί στην συνέχεια από τον ίδιο τον χρήστη.

Πως αλλάζει η λέξη-κλειδί

- Η εντολή: **passwd** ξεκινά την διαδικασία αλλαγής του password.
- Το Linux αποκρίνεται: **enter old password:**
- Ο χρήστης πληκτρολογεί το ήδη ισχύον password που δεν εμφανίζεται στην οθόνη
- Το Linux αποκρίνεται: **enter new password:**
- Ο χρήστης πληκτρολογεί το νέο password.
- Το Linux αποκρίνεται: **reenter new password:**
- Ο χρήστης πληκτρολογεί ξανά το νέο password.
- Το πλήκτρο **ENTER** χρησιμοποιείται στο τέλος της κάθε πληκτρολόγησης.

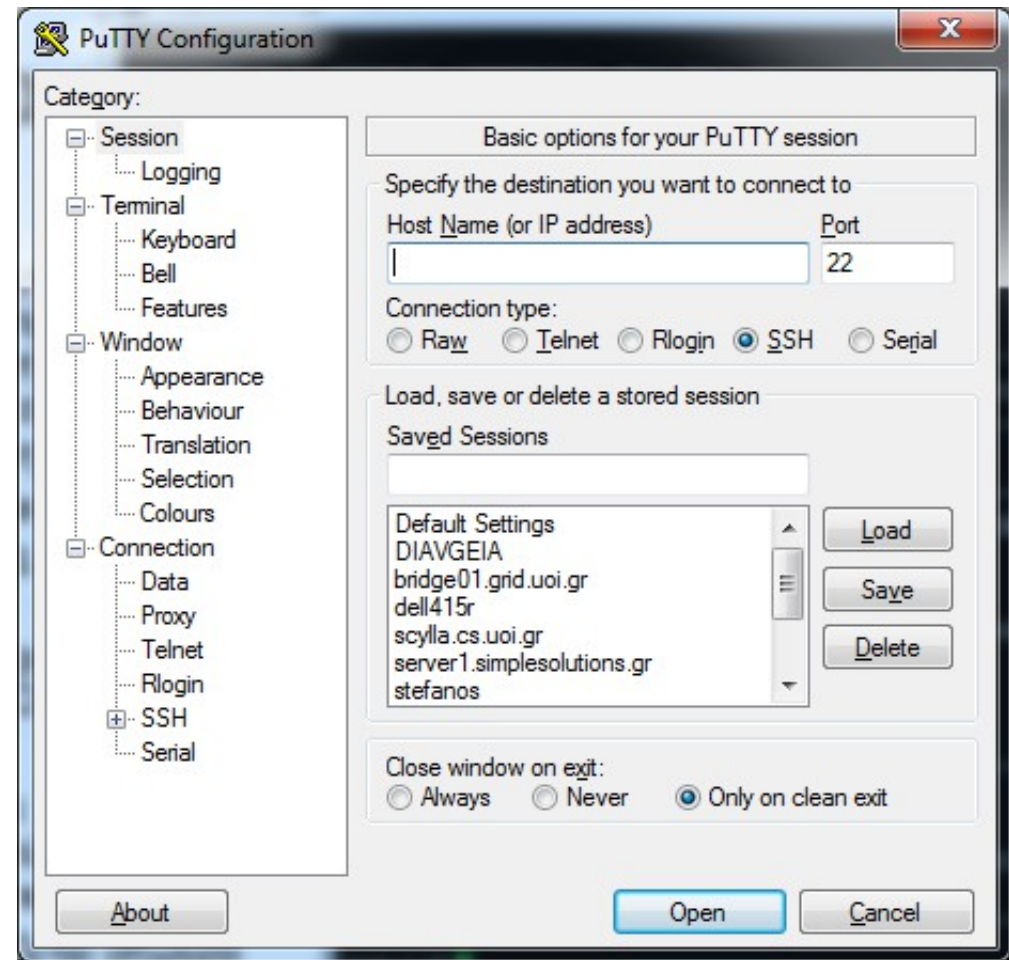
Η διαδικασία “logout”

Για να αποσυνδεθεί ο χρήστης υπάρχουν διάφοροι ισοδύναμοι τρόποι.

- Πληκτρολόγηση της λέξης: **exit** και **ENTER**
- Πληκτρολόγηση της λέξης: **logout** και **ENTER**
- Κρατάμε το πλήκτρο **Control** πατημένο και πληκτρολογούμε τον χαρακτήρα **D**.

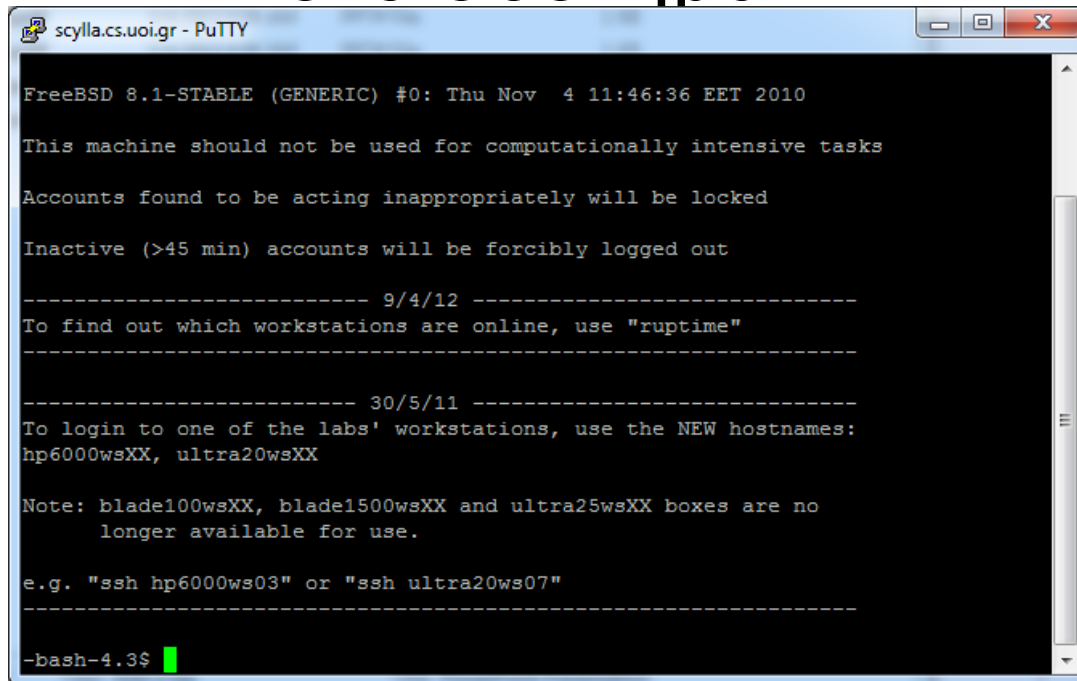
Απομακρυσμένη είσοδος

- ssh – Το ssh (secure shell) είναι πρόγραμμα για ασφαλή σύνδεση με άλλο υπολογιστή, για ασφαλή εκτέλεση εντολών σε απομακρυσμένο μηχάνημα
 - **ssh remotews01**
 - **ssh -l user1 remotews01**
 - Στα Windows putty, SSH Secure Shell Client, ...
- telnet – είσοδος σε απομακρυσμένο σύστημα μέσω της υπηρεσίας telnet
 - Δεν υποστηρίζεται σήμερα



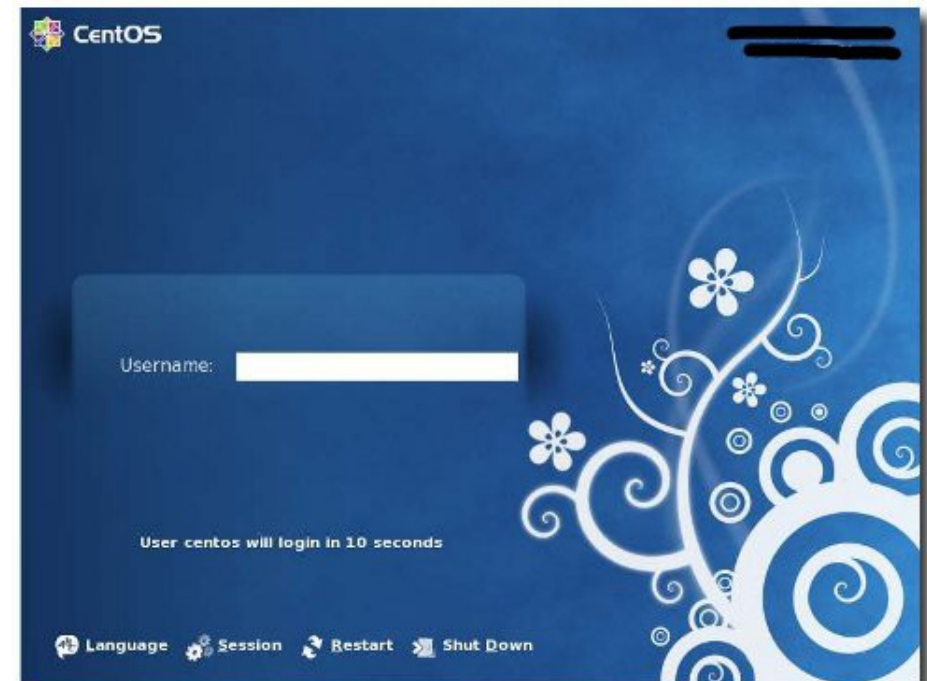
Ενότητα 2: Συνδεδεμένοι στο σύστημα

- Επιτυχημένη διαδικασία login μας «εισάγει» στο σύστημα



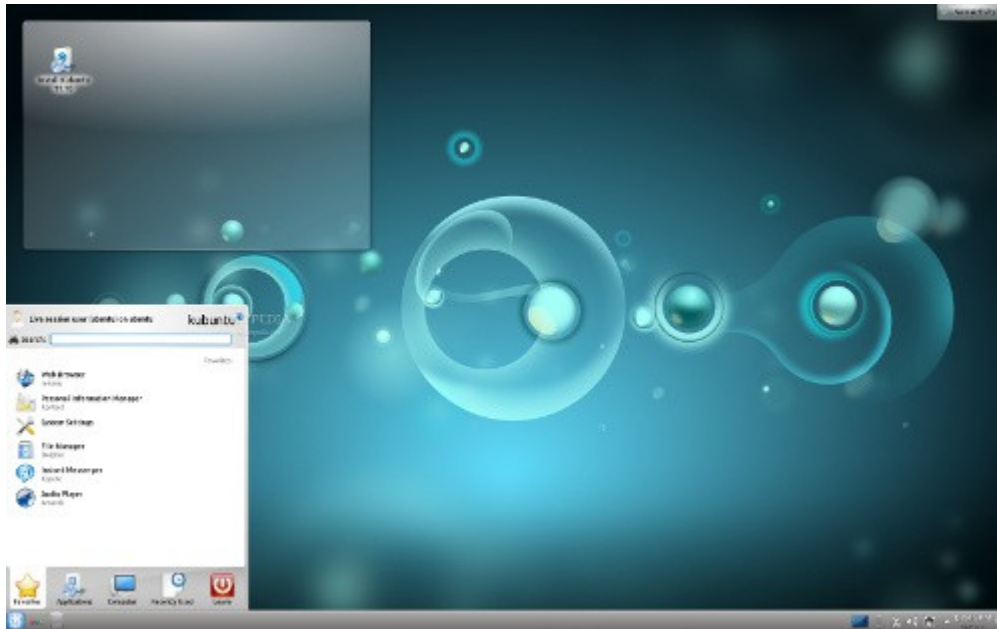
scylla.cs.uoi.gr - PuTTY

```
FreeBSD 8.1-STABLE (GENERIC) #0: Thu Nov  4 11:46:36 EET 2010  
This machine should not be used for computationally intensive tasks  
Accounts found to be acting inappropriately will be locked  
Inactive (>45 min) accounts will be forcibly logged out  
  
----- 9/4/12 -----  
To find out which workstations are online, use "ruptime"  
  
----- 30/5/11 -----  
To login to one of the labs' workstations, use the NEW hostnames:  
hp6000wsXX, ultra20wsXX  
  
Note: blade100wsXX, blade1500wsXX and ultra25wsXX boxes are no  
longer available for use.  
  
e.g. "ssh hp6000ws03" or "ssh ultra20ws07"  
  
-----  
-bash-4.3$
```

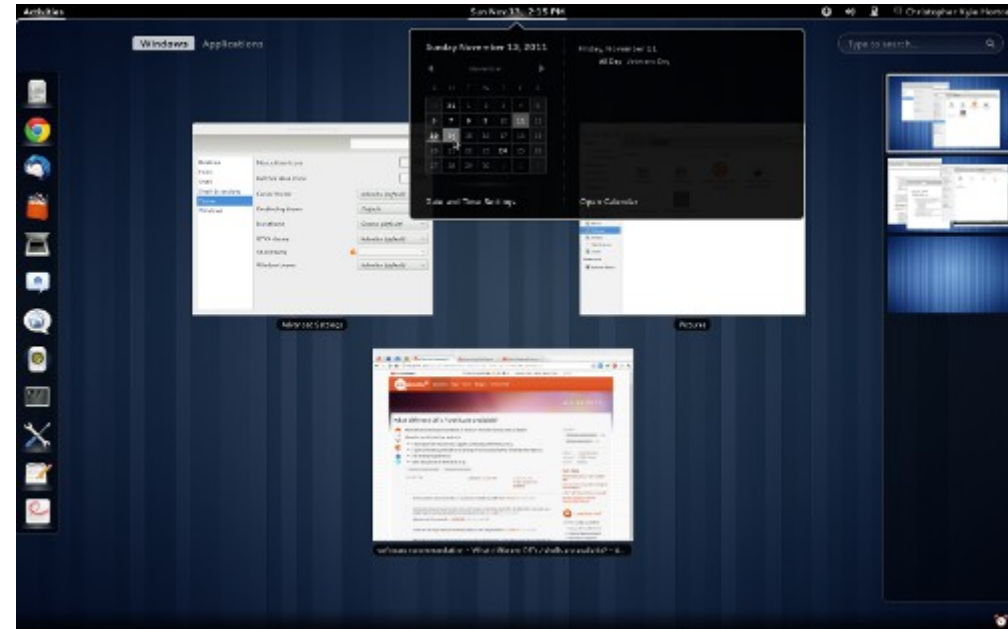


Εξοικείωση με το Περιβάλλον

- Υπάρχουν διάφορα περιβάλλοντα

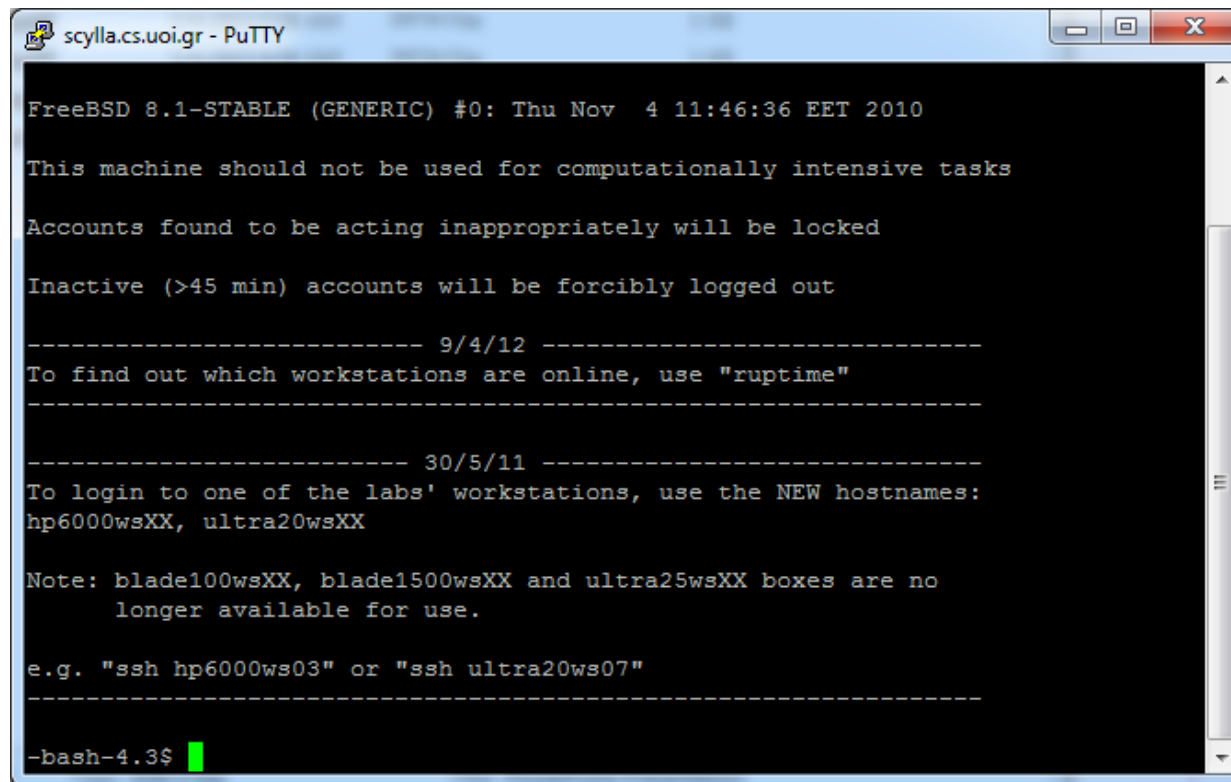


KDE



GNOME

Τερματικό - Πρόγραμμα



```
scylla.cs.uoi.gr - PuTTY

FreeBSD 8.1-STABLE (GENERIC) #0: Thu Nov  4 11:46:36 EET 2010

This machine should not be used for computationally intensive tasks

Accounts found to be acting inappropriately will be locked

Inactive (>45 min) accounts will be forcibly logged out

----- 9/4/12 -----
To find out which workstations are online, use "ruptime"
-----

----- 30/5/11 -----
To login to one of the labs' workstations, use the NEW hostnames:
hp6000wsXX, ultra20wsXX

Note: blade100wsXX, blade1500wsXX and ultra25wsXX boxes are no
longer available for use.

e.g. "ssh hp6000ws03" or "ssh ultra20ws07"

-----

-bash-4.3$
```

**Στο κάθε τερματικό εκτελείται ένα κέλυφος (shell) που
δέχεται την είσοδο του χρήστη και την «στέλνει» στο
σύστημα**

Τερματικό - Ιστορία



VT100



**Thin
client**



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Ταμείο
Περιφερειακής
Ανάπτυξης



ψηφιακή Ελλάδα
Όλα είναι δυνατά
Επικρατικό Πρόγραμμα
"Ψηφιακή Σύγκλιση"



Ενότητα 3: Πληροφορίες στο Linux – αλληλεπίδραση με το κέλυφος

Προηγούμενη αλληλεπίδραση

Command prompt

```
scylla.cs.uoi.gr - PuTTY
ls_output
mbox
output
ph3
ph4
pi_results.ods
project
public_html
readme.pdf
sfgdfgdfg
syntaktikos.c
test
test.c
test.ssss
test1.c
y7-lecture08.pdf
y7-lecture09.pdf
y7-lecture10+11.pdf
y7-lecture2.pdf
y7-lecture3.pdf
y7-lecture4.pdf
y7-lecture5+6.pdf
y7-lecture7.pdf
-bash-4.3$ ls -l
```

Cursor

Χρήστες και ομάδες

- Το Linux καθορίζει ομάδες χρηστών.
- Κάθε χρήστης μπορεί να ανήκει σε μια ομάδα ή περισσότερες ομάδες (groups).
- Κάθε χρήστης έχει ένα «προσωπικό χώρο» στο σύστημα
- Για να αλλάξει ένας χρήστης ομάδα πρέπει να επέμβει ο αναλυτής του συστήματος (**super-user ή root**).



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Ταμείο
Περιφερειακής
Ανάπτυξης



ψηφιακή Ελλάδα
Όλα είναι δυνατά
Ευρωπαϊκό Πρόγραμμα
"Ψηφιακή Σύγκλιση"



Διασυνδέοντας την Έρευνα και τ



ΕΣΠΑ
2007-2013
Πρόγραμμα για την ανάπτυξη
Ποιότητα ζωής για όλους



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΙΩΑΝΝΙΝΩΝ



Επιστημονικό &
Τεχνολογικό
Πάρκο
Ηπείρου



MEDICAL TECHNOLOGY
MedLab
INTELLIGENT
INFORMATION SYSTEMS

User ID – Group ID

- **id** : τυπώνει την ταυτότητα του χρηστή

uid=1003(user1) gid=513(None)

Ποιοι χρήστες χρησιμοποιούν το σύστημα

- Η εντολή **who** δίνει αυτού του είδους τις πληροφορίες.
- Συντάσσεται και με παραμέτρους.

\$ who

Τυπική έξοδος:

```
root      console  Apr 26    10:31
user11    tty01      Apr 26    12:34
user9801  tty04      Apr 26    14:25
```

who . . .

- Η πρώτη στήλη δείχνει το όνομα συνδεδεμένου χρήστη.
- Η δεύτερη στήλη τον τερματικό σταθμό από τον οποίο έγινε η σύνδεση.
- Οι υπόλοιπες στήλες δείχνουν την ημερομηνία και ώρα που έγινε η σύνδεση.

Η σύνταξη:

who am i

και

whoami

Δείχνει το όνομα του χρήστη και τον τερματικό σταθμό σύνδεσης.



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Ταμείο
Περιφερειακής
Ανάπτυξης



ψηφιακή Ελλάδα
Όλα είναι δυνατά
Ευρωπαϊκό Πρόγραμμα
"Ψηφιακή Σύγκλιση"



grnet

ΕΣΠΑ
2007-2013
Πρόγραμμα για την
Ποιότητα ζωής



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΙΩΑΝΝΙΝΩΝ



Επιστημονικό &
Τεχνολογικό
Πάρκο
Ηπείρου



MedLab
INTELLIGENT
INFORMATION SYSTEMS

Η εντολή: finger

Ζητά πληροφορίες για κάποιον χρήστη. Συντάσσεται και με παραμέτρους.

finger



Δίνει πληροφορίες για όλους τους συνδεδεμένους χρήστες.

finger name



Δίνει πληροφορίες για τους χρήστες που υποδεικνύονται από την λέξη: **name**

name	<i>Πληροφορίες</i>
jimmy	Μόνο για τον χρήστη jimmy
jimmy john	Για τους χρήστες jimmy και john jimmy@zeus
Για το χρήστη jimmy	στο μηχάνημα zeus @eros
Για όλους τους χρήστες στον H/Y	zeus

finger . . .

Η εντολή **finger** δίνει επιπλέον πληροφορίες που ο χρήστης έχει τοποθετήσει στα αρχεία:



.plan



.project

- Όλα τα περιεχόμενα του **.plan** εμφανίζονται. *Ο χρήστης εδώ μπορεί να έχει γράψει το τηλέφωνό του, την τοποθεσία του γραφείου του, τις ώρες που είναι διαθέσιμος, κλπ.*
- Μόνο η πρώτη γραμμή του **.project** εμφανίζεται.



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Ταμείο
Περιφερειακής
Ανάπτυξης



ψηφιακή Ελλάδα
Θάλα είναι δυνατά
Επικρατικό Πρόγραμμα
"Ψηφιακή Σύγκλιση"



Διασυνδέοντας την Έρευνα και τ



Ποιότητα ζωής για όλους



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΙΩΑΝΝΙΝΩΝ



Επιστημονικό &
Τεχνολογικό
Πάρκο
Ηπείρου



INTELLIGENT
INFORMATION SYSTEMS

Οι εντολές: **tty** και **date**

tty



Επιστρέφει το όνομα του τερματικού στο οποίο ο χρήστης είναι συνδεδεμένος.

/dev/pts/3

date



Επιστρέφει την ημερομηνία και ώρα σε προκαθορισμένη μορφή.

Mon Oct 12 09:00:00 2009



Ευρωπαϊκό Ταμείο
Περιφερειακής
Ανάπτυξης



Όλα είναι δυνατά
Επικρατικό Πρόγραμμα
"Ψηφιακή Σύγκλιση"



grnet
Διασυνδέοντας την Έρευνα και τ

2007-2013
Πρόγραμμα για την ανάπτυξη
Ποιότητα ζωής για όλους

ΠΑΝΕΠΙΣΤΗΜΙΟ
ΙΩΑΝΝΙΝΩΝ

Επιστημονικό &
Τεχνολογικό
Πάρκο
Ηπείρου

Med
lab
INTELLIGENT
INFORMATION SYSTEMS

Η εντολή cal

Παρουσιάζει πληροφορία ημερολογίου.

cal



Ημερολόγιο τρέχοντος μήνα

cal year



Ημερολόγιο ολόκληρου του έτους
year

cal month year



Ημερολόγιο του μήνα **month** του
έτους **year**

Η εντολή man ...

- man id
- man date
- man cal
- man finger
- man passwd

Βρίσκει και τυπώνει χρήσιμες πληροφορίες σχετικά με την ζητούμενη εντολή

Το Σύστημα Αρχείων του Linux

Περιεχόμενα διάλεξης

- Το σύστημα αρχείων του Linux
- Σημαντικοί κατάλογοι
- Σημαντικά αρχεία
- Σχετικά και απόλυτα μονοπάτια
- Εντολή pwd
- Εντολή ls
- Εντολή cd
- Εντολή df, du
- Εντολή quota
- Εντολή file

Το σύστημα αρχείων του Linux

- Το σύστημα αρχείων είναι μια λογική συλλογή αρχείων σε μια διαμόρφωση (partition) ενός σκληρού δίσκου.
- Ένας σκληρός δίσκος μπορεί να περιέχει περισσότερες από μια διαμορφώσεις, κάθε μια από τις οποίες «φιλοξενεί» ένα λειτουργικό σύστημα.
- Αυτό επιτρέπει την εύκολη λογική διαχείριση διαφορετικών συστημάτων αρχείων.
- Τα ΠΑΝΤΑ στο Linux θεωρούνται ότι είναι αρχεία και αυτό περιλαμβάνει φυσικές συσκευές όπως (DVD-ROMs, συσκευές USB, δισκέττες κ.ο.κ).
 - Συνηθισμένα αρχεία
 - Κατάλογοι
 - Ειδικά αρχεία συσχετιζόμενα με το υλικό
 - Ειδικά αρχεία για επικοινωνία μεταξύ προγραμμάτων

Συνηθισμένα αρχεία

- Προγράμματα – Εφαρμογές
 - Εκτελέσιμα αρχεία a.out που δημιουργεί ο compiler της C
- Αρχεία κείμενου
 - Ο πηγαίος κώδικας της C
- Αρχεία δεδομένων
 - Αρχεία που περιέχουν δεδομένα σε μη εκτυπώσιμη μορφή

Κατάλογοι

- Ειδικής μορφής αρχεία
- Σταθερό μέγεθος
- Περιέχουν αρχεία και άλλους καταλόγους
 - Ουσιαστικά περιέχουν μια λίστα με τα ονόματα των αρχείων και των υπο-καταλόγων.
- Περιέχουν δυο ειδικούς καταλόγους τον «.» και τον «..»

Αρχεία συσκευών – device files

- Τα αρχεία συσκευών χρησιμοποιούνται για να παρέχουν μια διεπαφή για έναν οδηγό μιας συσκευής (device driver)
- Επιτρέπουν στο λειτουργικό σύστημα να αλληλεπιδράσει με τον οδηγό της συσκευής χρησιμοποιώντας απλές κλήσεις εισόδου/εξόδου.
- Εκτυπωτές, σκληροί δίσκοι (/dev/sda0), CDROMS (/dev/cdrom), modem (/dev/modem)

Σύνδεσμοι σε αρχεία

hard links – soft (symbolic) links

Hard Links

- Παρέχουν τη δυνατότητα ύπαρξης πολλαπλών αντιγράφων ενός αρχείου σε διαφορετικούς υποκαταλόγους του ίδιου φυσικού μέσου.
- Τα αντίγραφα υλοποιούνται μόνο σε λογικό επίπεδο.
- Δεν δημιουργούνται πραγματικά αντίγραφα απλά σύνδεσμοι που ανακατευθύνουν οποιαδήποτε εντολή πάνω στον σύνδεσμο στο πραγματικό αρχείο.

Symbolic links

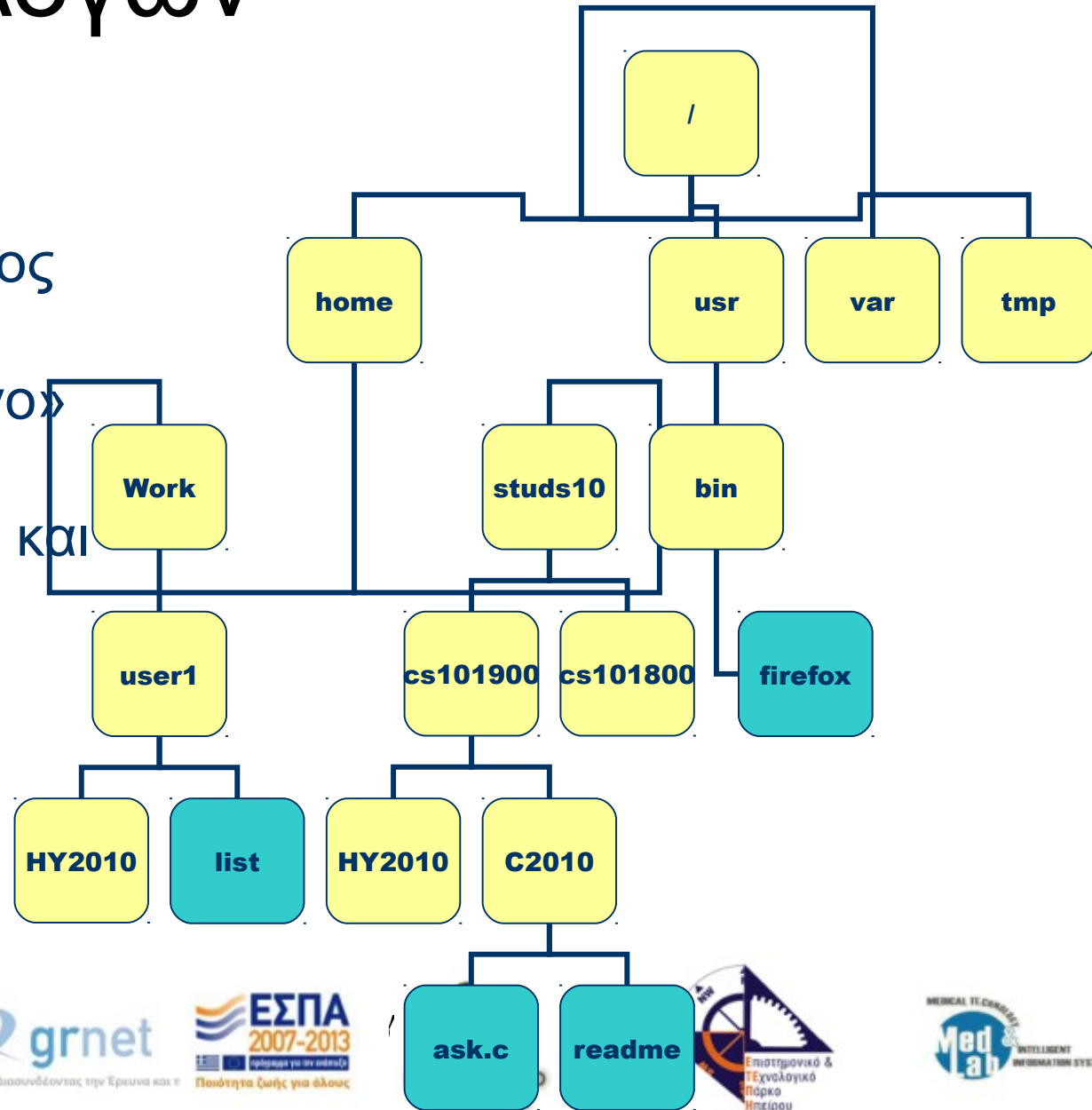
- Σύνδεσμοι οι οποίοι μπορούν να αναφέρονται και σε υποκαταλόγους αλλά και σε αρχεία διαφορετικών φυσικών μέσων αποθήκευσης
- Μπορούν να αναφέρονται και σε αρχεία που βρίσκονται σε διαφορετικούς Η/Υ στο τοπικό δίκτυο ή ακόμη και σε απομακρυσμένα δίκτυα (π.χ μέσω του Διαδικτύου).

Αρχεία επικοινωνίας – named pipes και sockets

- Ένας τρόπος επικοινωνίας μεταξύ διεργασιών στο Linux.
- Είναι ένα ειδικό αρχείο στο οποίο γράφουν και διαβάζουν οι διεργασίες προκειμένου να ανταλλάξουν δεδομένα

Δομή καταλόγων

- Ιεραρχική δομή
- Ο βασικός κατάλογος είναι ο «/»
- «Αναποδογυρισμένο» δέντρο
- Απεριόριστο βάθος και πλάτος
- Κάθε φορά «βρισκόμαστε» σε έναν κατάλογο

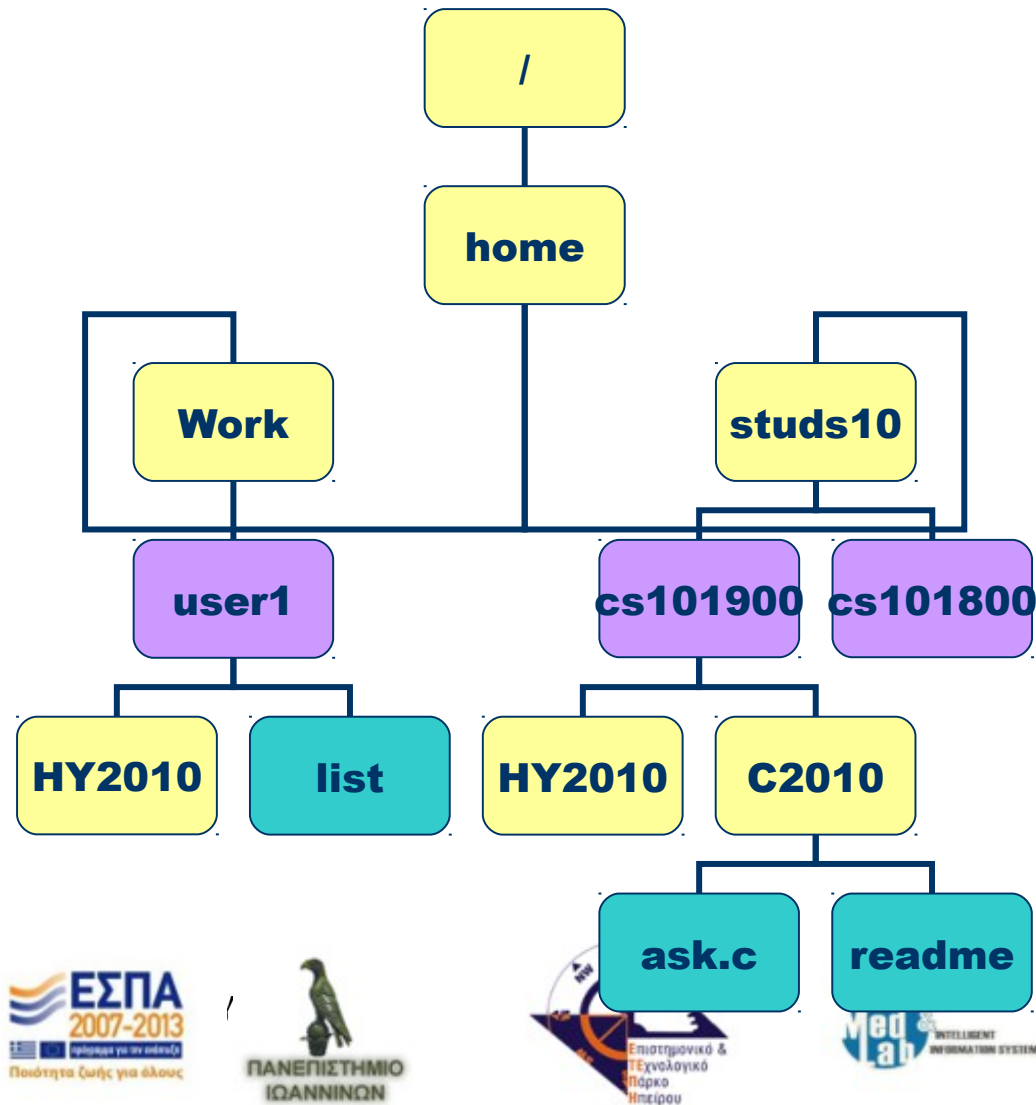


Τρέχων Κατάλογος – Κατάλογος Εργασίας – Working directory

- Πρόκειται για τον κατάλογο στον οποίο βρίσκεται ο χρήστης.
- Εντολή `pwd` : Τυπώνει το απόλυτο μονοπάτι
outputs the absolute path (more on this later) of your working directory
- Unless you specify another directory, commands will assume you want to operate on the working directory

Προσωπικός κατάλογος χρήστη – Home directory

- Ειδικό μέρος στο σύστημα αρχείων στο οποίο ο κάθε χρήστης διατηρεί τα προσωπικά αρχεία.
- Όταν εισέρχεται ένας χρήστης στο σύστημα ο προσωπικός κατάλογος γίνεται ο τρέχοντας κατάλογος.
- Ο προσωπικός κατάλογος συμβολίζεται με το «~» (tilde).
- Δηλαδή ο προσωπικός φάκελος του χρήστη user1 συμβολίζεται με το ~user1.



Σημαντικοί κατάλογοι

Φάκελος	Περιγραφή
/	Κατάλογος «ρίζα» - Αρχικός κατάλογος
/bin	Εδώ βρίσκεται μεγάλο μέρος των εντολών. Διαθέσιμες σε όλους τους χρήστες
/dev	Περιέχει αρχεία συσκευών
/etc	Περιέχει εντολές και αρχεία ρύθμισης του διαχειριστή του συστήματος
/lib	Περιέχει βιβλιοθήκες προγραμμάτων
/boot	Περιέχει εντολές για την εκκίνηση του συστήματος
/home	Περιέχει τους προσωπικούς καταλόγους όλων των χρηστών
/mnt	Χρησιμοποιείται για προσάρτηση προσωρινών συστημάτων αρχείων CD-ROM, δισκέτα, USB



Σημαντικοί κατάλογοι

Φάκελος	Περιγραφή
/proc	Περιέχει της διεργασίες που εκτελούνται (με μορφή αρχείου)
/tmp	Περιέχει προσωρινά αρχεία
/usr	Περιέχει αρχεία που χρησιμοποιούνται από πολλούς χρήστες
/var	Περιέχει αρχεία καταγραφής (log files)
/sbin	Περιέχει εκτελέσιμα αρχεία για τον διαχειριστή
/kernel	Περιέχει αρχεία του πυρήνα

Σημαντικά αρχεία

Κατάλογος /etc

- passwd Περιέχει όλους τους χρήστες
- motd Περιέχει το "Μήνυμα της Ημέρας"
- mtab Λίστα προσαρτήσεων
- utmp Λίστα συνδεδεμένων χρηστών
- termcap Ορισμοί διαφόρων τερματικών

Σημαντικά αρχεία

Κατάλογος /dev

- console Συλλέγει τα μηνύματα του Linux
- null Εικονικό αρχείο
- lp Υποστήριξη εκτυπωτή
- mt# Υποστήριξη μαγνητικής ταινίας
- tty# Υποστήριξη τερματικής οθόνης

Αναφορά σε αρχείο - κατάλογο

- Κάθε στοιχείο του συστήματος αρχείων καθορίζεται από:
 - Τη θέση του
 - Το όνομά του
- Η θέση του στοιχείου μπορεί να καθοριστεί με το λεγόμενο «μονοπάτι» (path):
 - Απόλυτο μονοπάτι: Η διαδρομή προς το στοιχείο από το ριζικό κατάλογο «/»
 - Σχετικό μονοπάτι: Η διαδρομή προς το στοιχείο από τον τρέχον κατάλογο
- Στην ίδια θέση δεν υπάρχουν στοιχεία με το ίδιο όνομα

Ονόματα αρχείων και καταλόγων

- Το όνομα ενός αρχείου μπορεί να περιέχει έως και 256 χαρακτήρες εκτός από τον /
- Οι ειδικοί χαρακτήρες και σύμβολα έχουν συγκεκριμένη σημασία στο κέλυφος του Linux . Θα ήταν σκόπιμο να αποφεύγονται στα ονόματα αρχείων και καταλόγων.

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

Πχ. evamvak lab list → evamvak_lab_list

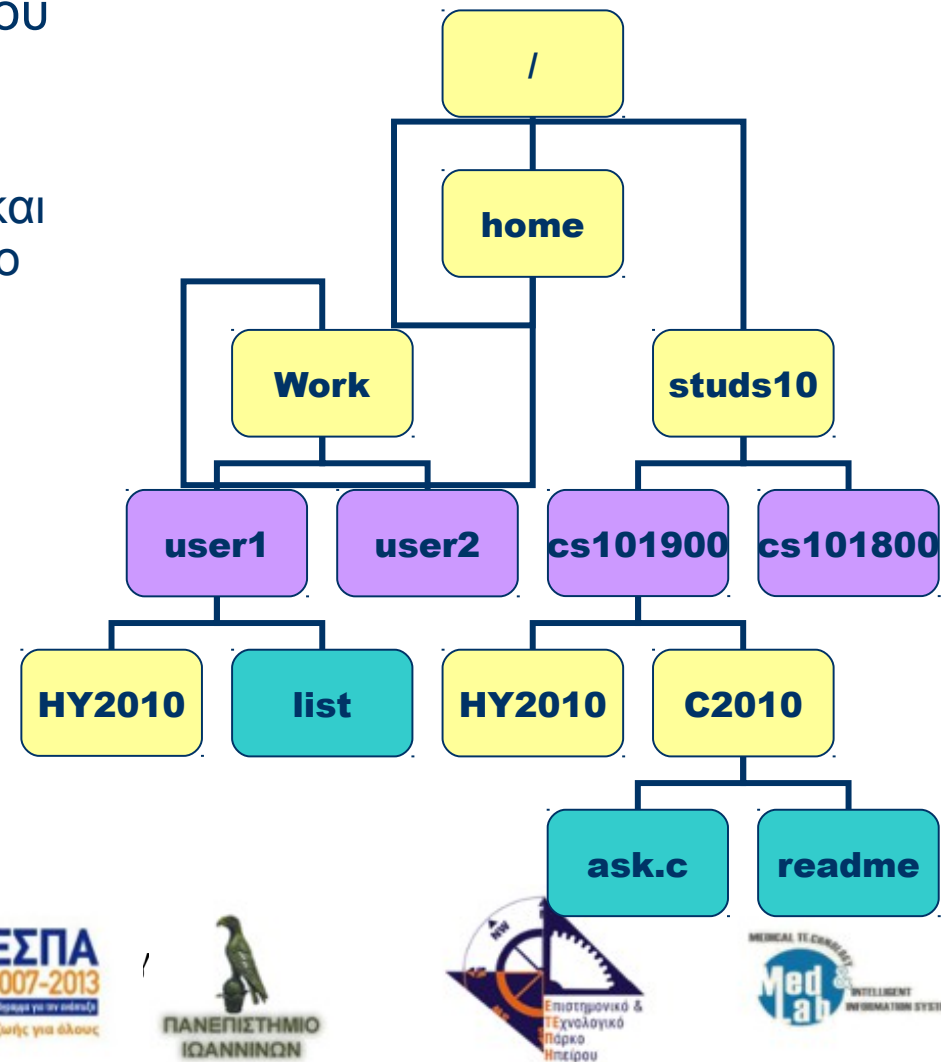
- Κεφαλαία – μικρά έχει διαφορά

Πχ. Costas ≠ costas

- Τα αρχεία που το όνομά τους ξεκινά με «.», λέγονται κρυφά.

Απόλυτα – σχετικά μονοπάτια

- Μονοπάτι: Ακολουθία καταλόγων που χωρίζονται με το «/»
- Απόλυτο μονοπάτι:
 - Ξεκινάει από τον αρχικό κατάλογο και ακολουθώ το δέντρο έως το στοιχείο
 - πχ. /home/Work/user1/list
 - πχ. /home/studs10/cs101900
- Σχετικό μονοπάτι:
 - Ξεκινάει από το τρέχοντα κατάλογο
 - Χρησιμοποιεί το «..» για να αναφερθεί στον παραπάνω κατάλογο, το «.» για τον τρέχων και το «~» για το home.
 - πχ. Είμαι στο home directory **user1**
 - ./list ../../studs10/cs101800
../user2



Απόλυτα – σχετικά μονοπάτια

- Στόχος το αρχείο ask2.c

Απόλυτο μονοπάτι:

/home/studs10/cs101900/C2010/ask.c

Σχετικό μονοπάτι:

Είμαι στο **cs101800**

../cs101900/C2010/ask.c

../../../../studs10/cs101900/C2010/ask.c

~cs101900/C2010/ask.c

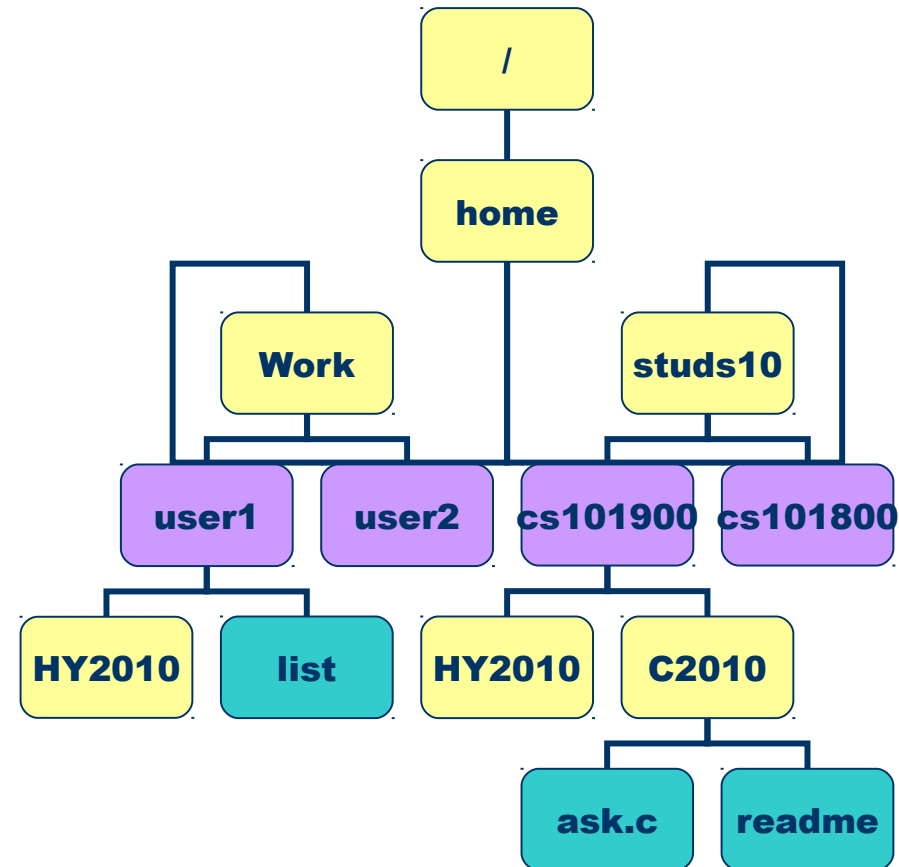
./C2010/ask.c

C2010/ask.c

Είμαι στο **user2**

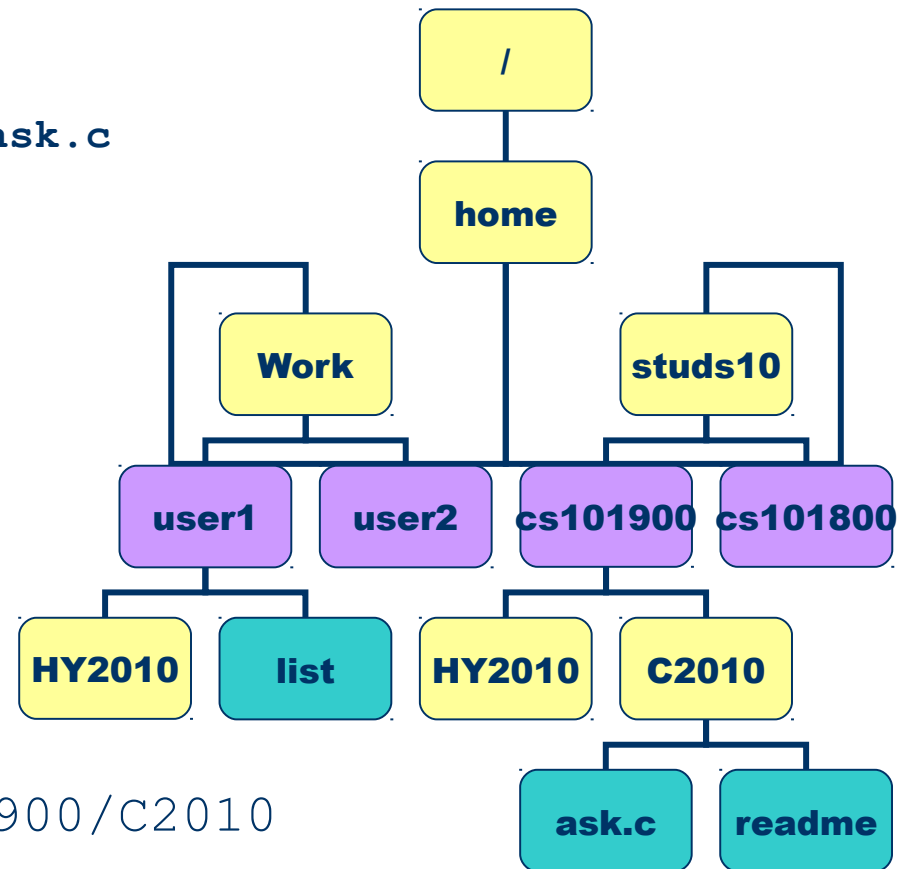
../../../../studs10/cs101900/C2010/ask.c

~cs101900/C2010/ask.c



Σχετικά μονοπάτια

- Τρέχων κατάλογος: `user1`
- Στόχος: `/home/studs10/cs101900/ask.c`
- Βήμα – βήμα
 - `cd ..`
 - `cd ..`
 - `cd studs10`
 - `cd cs101900`
 - `cd C2010`
- Όλο το μονοπάτι
- Από τον `home` κατάλογο
 - `cd /home/studs10/cs101900/C2010`



Εντολή pwd

- Εμφανίζει το όνομα του τρέχοντος καταλόγου.
- Απόλυτο μονοπάτι!

user1 : **/usr/home/grads/Work/user1**

cs101900 : **/usr/home/students/stud10/cs101900**

- Απαντάει στην ερώτηση: « -Πού είμαι;»

Εντολή ls

- Παρέχει πληροφορίες για αρχεία και καταλόγους.
- Συντάσσεται με παραμέτρους (flags).

```
$ ls
```

τυπώνει τα ονόματα των αρχείων του τρέχοντος καταλόγου

```
$ ls -l
```

τυπώνει πληροφορίες για όλα τα αρχεία του τρέχοντος καταλόγου

```
$ ls <dirname>
```

τυπώνει τα ονόματα των αρχείων του καταλόγου <dirname>

Εντολή ls (παράδειγμα)

- \$ ls
- \$ ls -l

Παράμετροι της ls

- `ls -l`: Προβολή λεπτομερειών για κάθε αρχείο

Δικαιώματα	Αριθμός συνδέσμων		Μέγεθος σε bytes		Όνομα αρχείου		
	Χρήστης - Ομάδα		Ημερομηνία τροποποίησης				
-rw-r--r--	1	tolvog guest	0	Dec 4 11:51	file1		
-rw-r--r--	1	tolvog guest	0	Dec 4 11:51	file2		
drwxr-xr-x	2	tolvog guest	4096	Dec 4 11:51	test1		
-rw-r--r--	1	tolvog guest	0	May 4 10:37	test4		

Παράμετροι της ls

- `ls -p`: Εισαγωγή «/» στο τέλος του ονόματος των καταλόγων

Σε κάποια
συστήματα

`ls -F`

```
hw4.pdf
lab01/
lab02/
lab6/
ls_output
mbox
output
ph3/
ph4/
pi_results.ods
project/
public_html/
readme.pdf
```



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Ταμείο
Περιφερειακής
Ανάπτυξης



ψηφιακή **Ρ**ημάδα
Όλα είναι δυνατά
Ευρωπαϊκό Πρόγραμμα
"Ψηφιακή Σύγκλιση"



Διασυνδέοντας την Έρευνα και τ



ΕΣΠΑ
2007-2013
Πρόγραμμα για την ανάπτυξη
Ποιότητα ζωής για όλους



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΙΩΑΝΝΙΝΩΝ



Επιστημονικό &
Τεχνολογικό
Πάρκο
Ηπείρου



MEDICAL TECHNOLOGY
INTELLIGENT
INFORMATION SYSTEMS

Παράμετροι της ls

- `ls <dirname>`: Προβάλλει τα περιεχόμενα του φακέλου `<dirname>` (σχετικό ή απόλυτο μονοπάτι)

```
-bash-4.3$ ls lab01/
Makefile      server      stoiba.h     utils.h~
Makefile~     server.c    stoiba.h~    utils.o
client        server.c~   stoiba.o
client.c      stoiba.c    utils.c
samples      stoiba.c~   utils.h
-bash-4.3$
```


Παράμετροι της ls

- `ls -a`:
Προβάλλει
τα κρυφά
αρχεία

```
-bash-4.3$ ls -a
.
..
.ICEauthority
.Xauthority
.adobe
.bash_history
.bash_profile
.bashrc
.cache
.checkgmail
.compiz
.config
.cshrc
.dbus
```

Παράμετροι της ls

- `ls -t`:
Ταξινόμησε
τα αρχεία
από το πιο
πρόσφατο
στο πιο
παλιό
- `ls -r`:
Αντιστρέφει τη
σειρά
προβολής των
αρχείων
- `ls -R`:
Προβάλλει
αναδρομικά τα
περιεχόμενα
των
υποκαταλόγων

Παράμετροι της ls

```
-bash-4.3$ ls -R
DEADLIER_10          de.c~
DEADLIER_100         de_best_1_sphere.c
DEADLIER_50          de_best_1_sphere.c~
J19.pdf              karagiannis.sql.gz
a.out                normal_DE
de.c

./DEADLIER_10:

./DEADLIER_100:

./DEADLIER_50:

./normal_DE:
DE_TP0_results.ods   de_best_1_sphere.txt
candlesticks.dat     de_best_1_sphere.txt[A
commands             de_best_1_sphere.txt~
de_best_1_sphere     de_rand_1_sphere
de_best_1_sphere.c   de_rand_1_sphere.c
de_best_1_sphere.c~  de_rand_1_sphere.txt
-bash-4.3$
```

Παράμετροι της ls

- Μπορώ να χρησιμοποιήσω περισσότερες από μια παραμέτρους
 - `ls -l -t`: Δείξε τις λεπτομέρειες των αρχείων αφού τα ταξινομήσεις από το πιο πρόσφατο στο πιο παλιό και
 - `ls -al`: Δείξε τις λεπτομέρειες **ΌΛΩΝ** των αρχείων (και των κρυφών)
- `ls -ltr ???`
- `ls -ta ???`

Εντολή cd (change directory)

- Με την εντολή αυτή καθορίζουμε τον τρέχοντα κατάλογο.
- Σύνταξη: `cd <dirname>`

`<dirname>`: σχετικό ή απόλυτο μονοπάτι

πχ. Δεδομένου ότι ο τρέχον κατάλογος είναι:

`/usr/home/vamvako/talks`

- Η εντολή: `cd ..` αλλάζει τον τρέχοντα κατάλογο στον αμέσως πιο πάνω ιεραρχικά ("γονικό" κατάλογο)
- Ο τρέχον κατάλογος πλέον θα είναι:

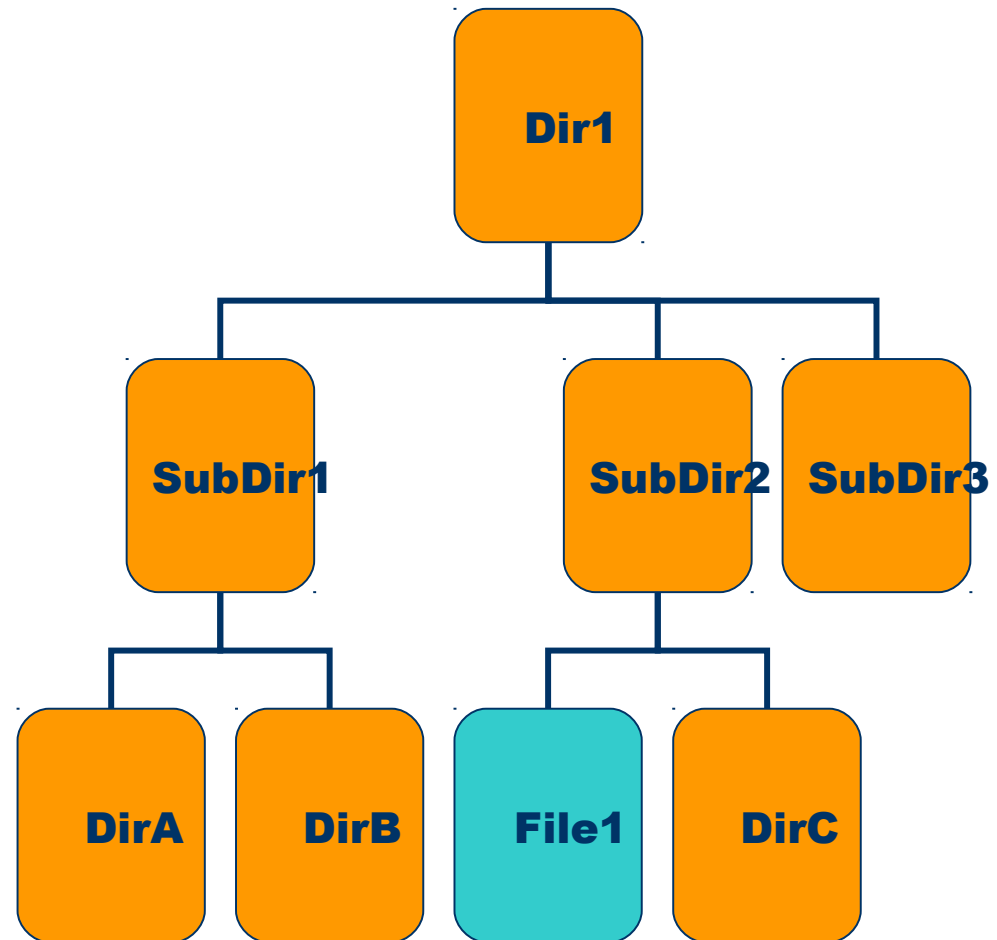
`/usr/home/vamvako`

- Μία τελεία παραπέμπει στον τρέχοντα κατάλογο, δηλαδή η εντολή:

`cd .` δεν αλλάζει κατάλογο.

Εντολή cd

- Dir1
- `cd SubDir1` → SubDir1
- `cd DirA` → DirA
- `cd ../../` → Dir1
- `cd SubDir/DirC` → DirC
 - `cd File1` ???
 - `cd ../SubDir3` ???
 - `cd ../../SubDir3` ???



Εντολή df

- df -k:
τυπώνει την
συνολική
(χρησιμοποιούμε
νη και διαθέσιμη)
ποσότητα
αποθηκευτικού
χώρου στο
σύστημα
αρχείων
(kbytes)
- df -h:
τυπώνει την ίδια
ποσότητα σε
MB και GB

```
Filesystem            1024-blocks      Used      Avail Capacity  Mounted on
/dev/mirror/sysmirror0s1a 10154158      883574      8458252      9%      /
devfs                   1              1              0     100%      /dev
/dev/mirror/sysmirror0s1e 428331170     4871084     389193594      1%      /usr
/dev/mirror/sysmirror0s1d 30462636      161930     27863696      1%      /var
procfs                   4              4              0     100%      /proc
uranus-vlan74:/usr/home/students 1965940896 117852812 1743230536      6%      /usr/home/students
uranus-vlan74:/usr/home/staff 2064244896 242875656 1716511692     12%      /usr/home/staff
uranus-vlan74:/usr/home/grads 2064244896 49864236 1909523112      3%      /usr/home/grads
gaia-vlan74:/var/mail 416971314 129674710 287296604     31%      /.amd_mnt/gaia-vlan74/host/var/mail
-bash-4.3$ df -h
Filesystem            Size      Used      Avail Capacity  Mounted on
/dev/mirror/sysmirror0s1a 9.7G      863M      8.1G      9%      /
devfs                   1.0K      1.0K        0B     100%      /dev
/dev/mirror/sysmirror0s1e 408G      4.6G      371G      1%      /usr
/dev/mirror/sysmirror0s1d 29G       158M      27G       1%      /var
procfs                   4.0K      4.0K        0B     100%      /proc
uranus-vlan74:/usr/home/students 1.8T      112G      1.6T      6%      /usr/home/students
uranus-vlan74:/usr/home/staff 1.9T      232G      1.6T     12%      /usr/home/staff
uranus-vlan74:/usr/home/grads 1.9T       48G      1.8T      3%      /usr/home/grads
gaia-vlan74:/var/mail 398G      124G      274G     31%      /.amd_mnt/gaia-vlan74/host/var/mail
-bash-4.3$
```


Εντολή du

Τυπώνει την χρησιμοποιούμενη ποσότητα του αποθηκευτικού χώρου στον τρέχοντα κατάλογο

■ **du -h :**

(αυτόματη προσαρμογή των μονάδων)
1024 (Kbyte) → 1 (Mbyte)

■ **du -s :** τυπώνει την συνολική
χρησιμοποιούμενη ποσότητα
του αποθηκευτικού χώρου
στον τρέχοντα κατάλογο

■ **du -sh /etc**

du -sh → 5M /etc

■ **du -a :** τυπώνει την
χρησιμοποιούμενη ποσότητα του
αποθηκευτικού χώρου για όλη την
δενδροειδή δομή των αρχείων και
καταλόγων που περιέχονται στον
τρέχοντα κατάλογο

du -sh → 245M .

Εντολή du

Παράδειγμα χρήσης:



```
-bash-4.3$ du -s project/
1792    project/
-bash-4.3$ du -h project/
4.0K    project/DEADLIER_10
4.0K    project/DEADLIER_50
4.0K    project/DEADLIER_100
460K    project/normal_DE
1.8M    project/
-bash-4.3$ du -skH project/
1792    project/
-bash-4.3$ du -skh project/
1.8M    project/
```



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Ταμείο
Περιφερειακής
Ανάπτυξης



ψηφιακή Ελλάδα
Όλα είναι δυνατά
Επικρατικό Πρόγραμμα
"Ψηφιακή Σύγκληση"



Διασυνδέοντας την Έρευνα και τ



Ποιότητα ζωής για όλους



Εντολή quota

- `quota -v`: Υπολογίζει και εμφανίζει πόσο χώρο επιτρέπεται να χρησιμοποιεί ο χρήστης στο filesystem.

```
-bash-4.3$ quota -v tolvog
Disk quotas for user tolvog (uid 10093):
```

Filesystem	usage	quota	limit	grace	files	quota	limit	grace
/usr/home/students	0	0	0		0	0	0	
/usr/home/staff	6838604	0	0		2482	0	0	
/usr/home/grads	0	0	0		0	0	0	

Εντολή file

- Τυπώνει τον τύπο του αρχείου

file myfile.txt → **myfile.txt: ASCII test**

file /etc → **/etc: directory**

file /dev/tty38 → **/dev/tty38: character special (4/38)**

Η εντολή file αναμένει τουλάχιστον ένα όρισμα
(αρχείο ή κατάλογο)

Διαχείριση Αρχείων και Καταλόγων

Περιεχόμενα διάλεξης

- Εντολές διαχείρισης καταλόγων
 - Δημιουργία καταλόγων
 - Εντολή mkdir
 - Διαγραφή καταλόγων
 - Εντολή rmdir
- Εντολές διαχείρισης αρχείων
 - Δημιουργία αρχείων
 - nedit, gedit, vi
 - Touch, cat
 - Διαγραφή αρχείων
 - Εντολή rm
 - Αντιγραφή αρχείων
 - Εντολή cp και ln
 - Μετακίνηση αρχείων/καταλόγων
 - Εντολή mv
- Μεταχαρακτήρες

Δημιουργία καταλόγων – Εντολή mkdir

- Σύνταξη
 - `mkdir [-m mode] [-p] <dirname> . . .`
- Περιγραφή
 - Δημιουργία του καταλόγου <dirname>
- Παράμετροι
 - **-m** Καθορίζει τα δικαιώματα δημιουργίας
 - **-p** Δημιουργήσει και τους ενδιάμεσους καταλόγους που ορίζει το <dirname>
- Παραδείγματα
 - `mkdir -m 700 letter`
 - `mkdir abc`
 - `mkdir -p ./abc/def/ghi`



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Ταμείο
Περιφερειακής
Ανάπτυξης



ψηφιακή Ελλάδα
Ολοκληρωμένο είναι το
Εθνικό Πρόγραμμα
"Ψηφιακή Σύγκλιση"



Διασυνδέοντας την Έρευνα και τ

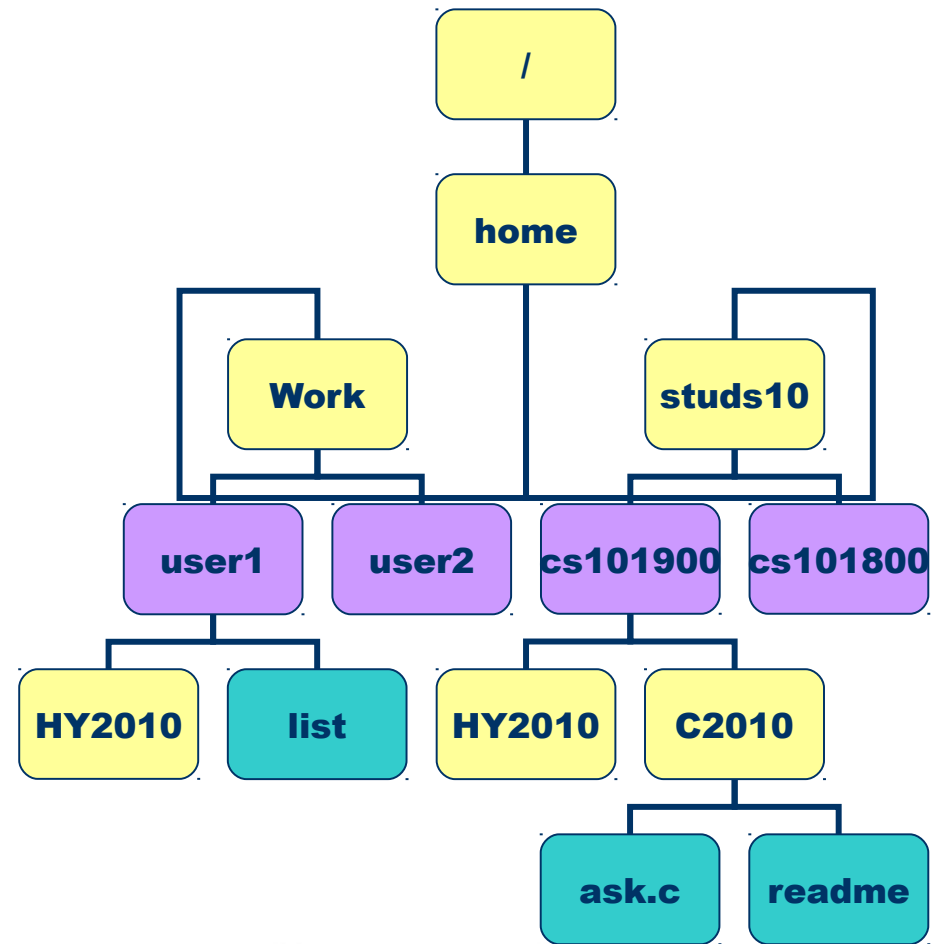


Ποιότητα ζωής για όλους



Παράδειγμα mkdir

- Βρίσκομαι στο
/home/Work/user1
\$ mkdir NewDir



Παράδειγμα mkdir

- Βρίσκομαι στο
/home/Work/user1

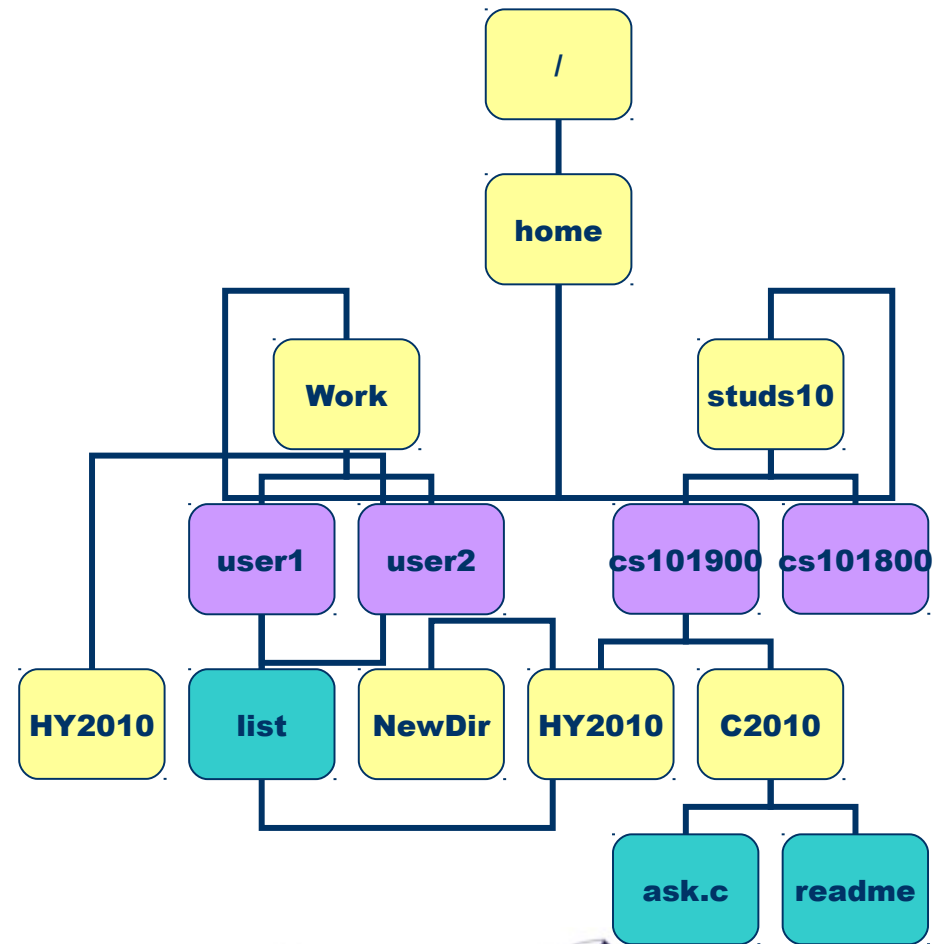
\$ **mkdir NewDir**

\$ **cd NewDir**

\$ **pwd**

/home/Work/user1/NewDir

\$ **mkdir Dir1**



Παράδειγμα mkdir

- Βρίσκομαι στο
/home/Work/user1

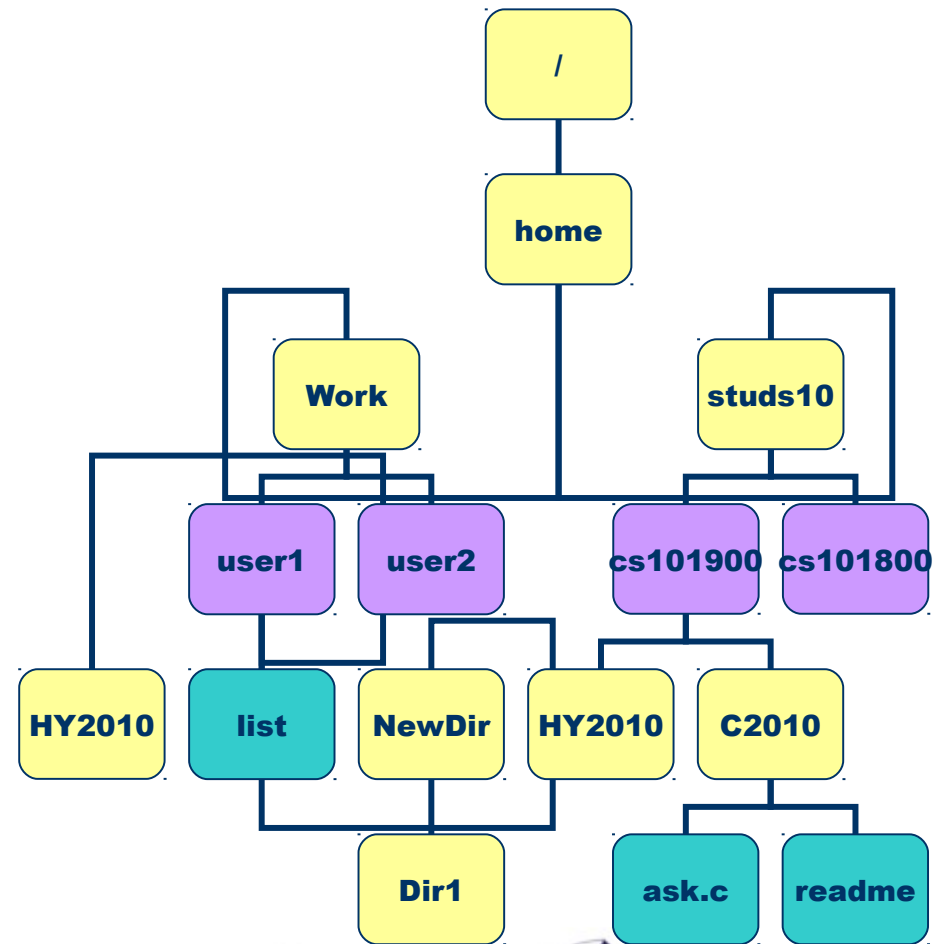
\$ **mkdir NewDir**

\$ **cd NewDir**

\$ **pwd**

/home/Work/user1/NewDir

\$ **mkdir Dir1**



Παράδειγμα mkdir

- Βρίσκομαι στο
/home/Work/user1

\$ **mkdir NewDir**

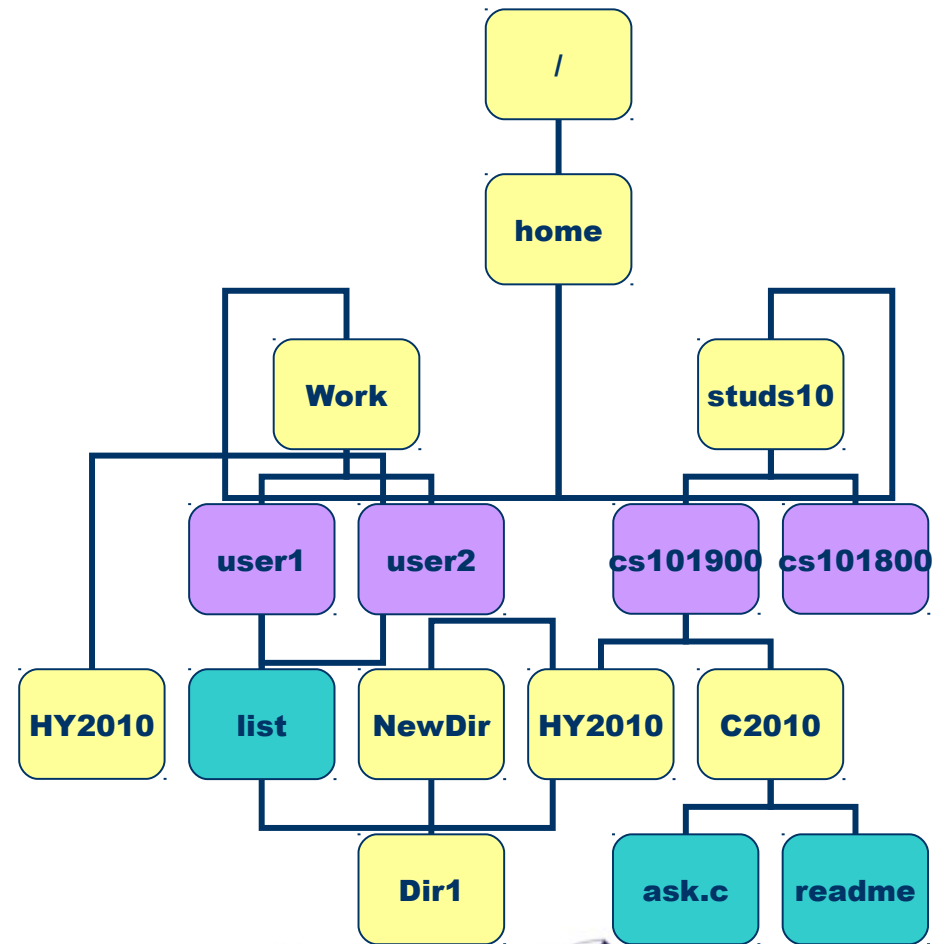
\$ **cd NewDir**

\$ **pwd**

/home/Work/user1/NewDir

\$ **mkdir Dir1**

\$ **mkdir ../../TestDir**



Παράδειγμα mkdir

- Βρίσκομαι στο
/home/Work/user1

\$ **mkdir NewDir**

\$ **cd NewDir**

\$ **pwd**

/home/Work/user1/NewDir

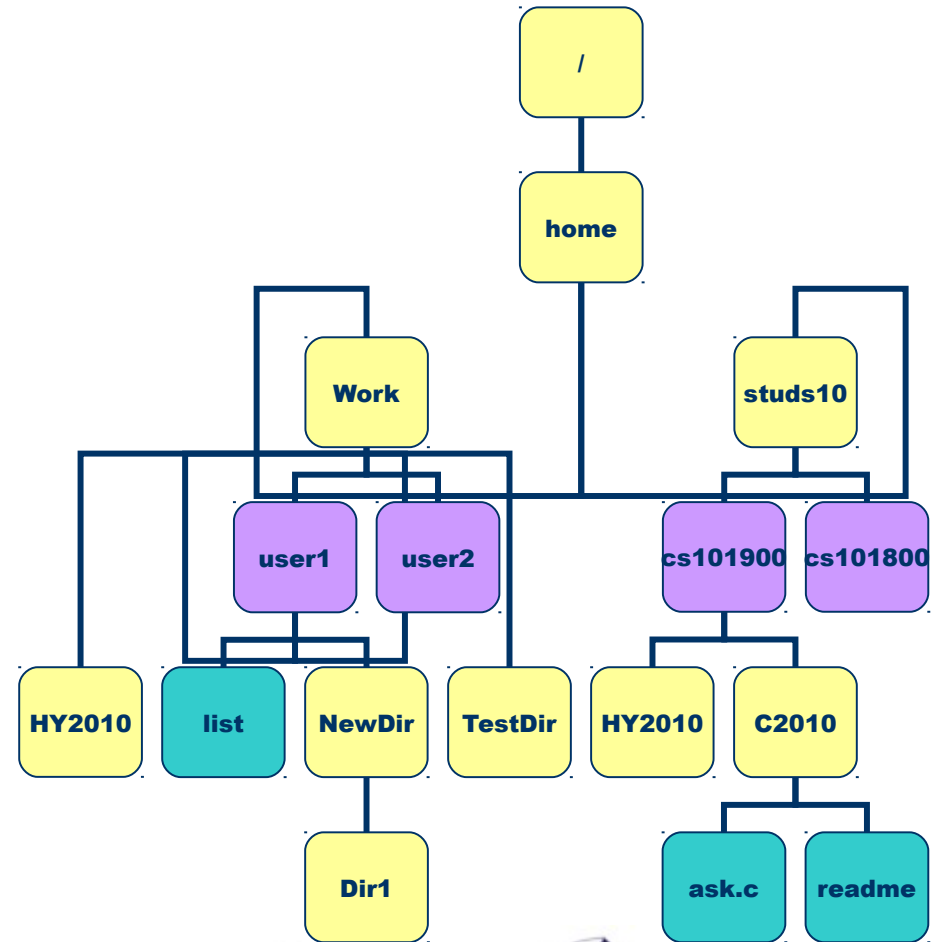
\$ **mkdir Dir1**

\$ **mkdir ../../TestDir**

\$ **pwd**

/home/Work/user1/NewDir

\$ **mkdir ~user2/NewDir**



Παράδειγμα mkdir

- Βρίσκομαι στο
/home/Work/user1

\$ mkdir NewDir

\$ cd NewDir

\$ pwd

/home/Work/user1/NewDir

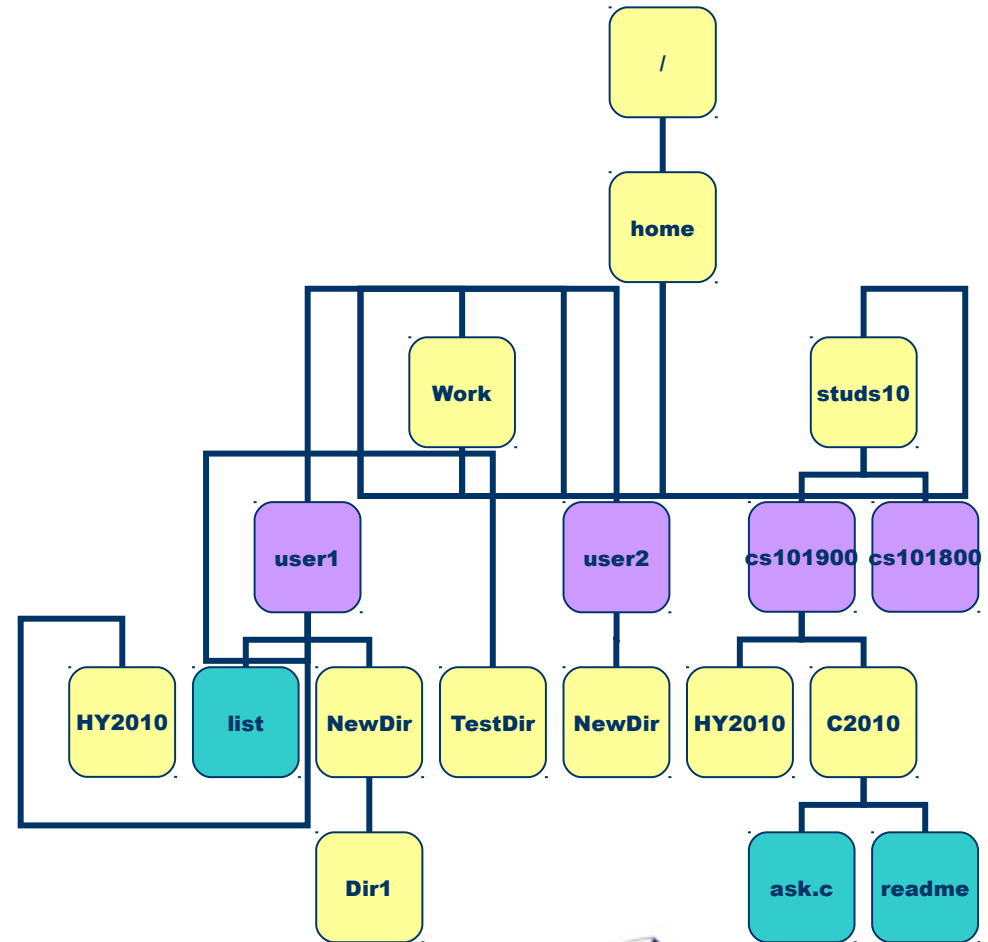
\$ mkdir Dir1

\$ mkdir ../../TestDir

\$ pwd

/home/Work/user1/NewDir

\$ mkdir ~user2/NewDir



Διαγραφή καταλόγων – Εντολή rmdir

- Σύνταξη
 - `rmdir [-p][-s] <dirname> . . .`
- Περιγραφή
 - Διαγραφή του καταλόγου <dirname> (εάν είναι κενός)
- Παράμετροι
 - **-p** Διέγραψε τον κατάλογο <dirname> και τον γονικό κατάλογο αν είναι πλέον κενός
 - **-s** Δεν εμφανίζει μήνυμα
- Παράδειγμα

`rmdir letter`



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Ταμείο
Περιφερειακής
Ανάπτυξης



ψηφιακή Ελλάδα
είναι είναι
Επικρατικό Πρόγραμμα
"Ψηφιακή Σύγκληση"



Διασυνδέοντας την Έρευνα και τ



ΕΣΠΑ
2007-2013
Πρόγραμμα για την ανάπτυξη
Ποιότητα ζωής για όλους



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΙΩΑΝΝΙΝΩΝ



Επιστημονικό &
Τεχνολογικό
Πάρκο
Ηπείρου



MEDICAL TECHNOLOGY
INTELLIGENT
INFORMATION SYSTEMS

Παράδειγμα rmdir

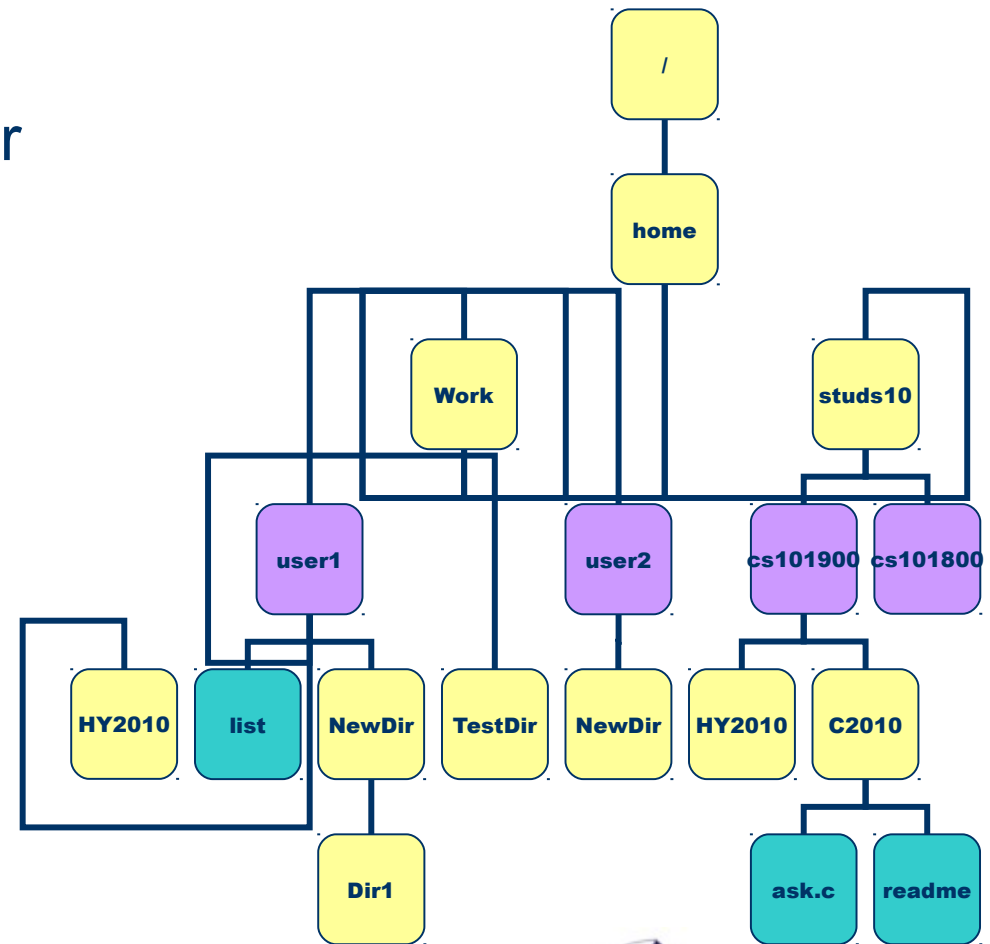
- Βρίσκομαι στο
/home/Work/user1/NewDir

\$ cd ..

\$ pwd

/home/Work/user1

\$ rmdir NewDir



Παράδειγμα rmdir

- Βρίσκομαι στο
/home/Work/user1/NewDir

\$ cd ..

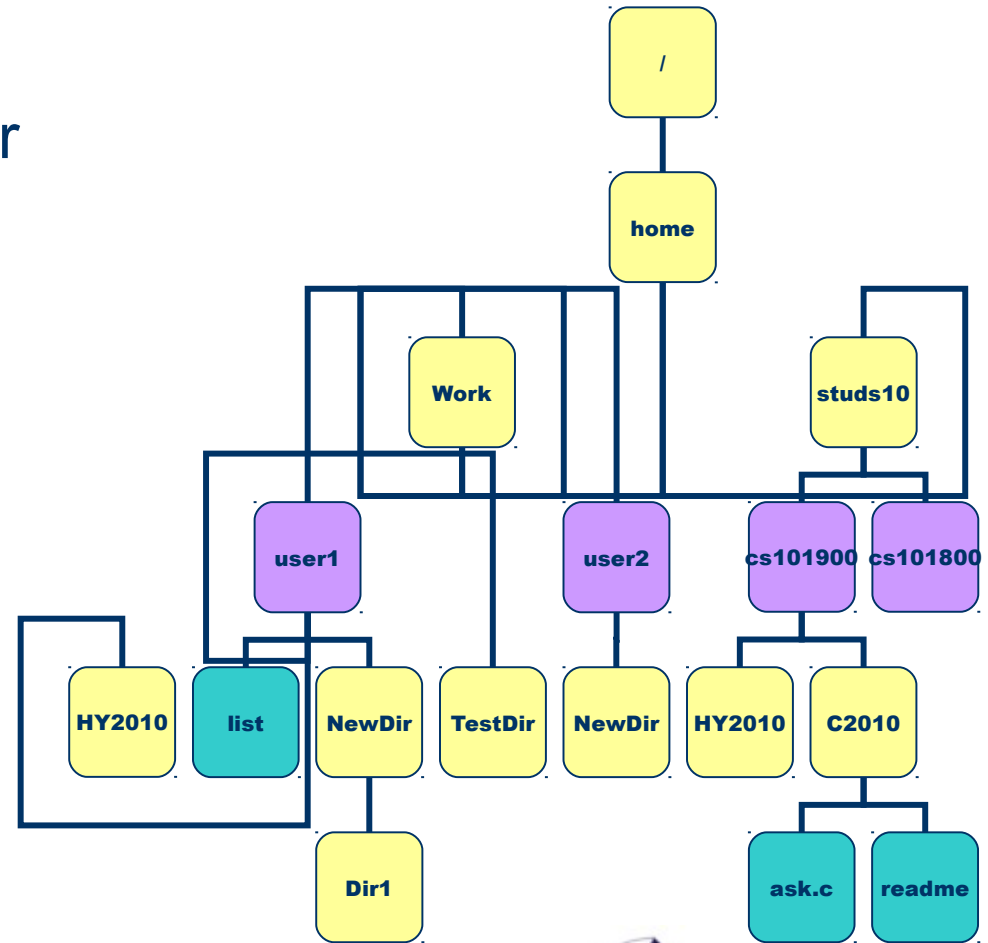
\$ pwd

/home/Work/user1

\$ **rmdir NewDir**

<Μήνυμα λάθους!>

\$ rmdir TestDir



Παράδειγμα rmdir

- Βρίσκομαι στο
/home/Work/user1/NewDir

\$ cd ..

\$ pwd

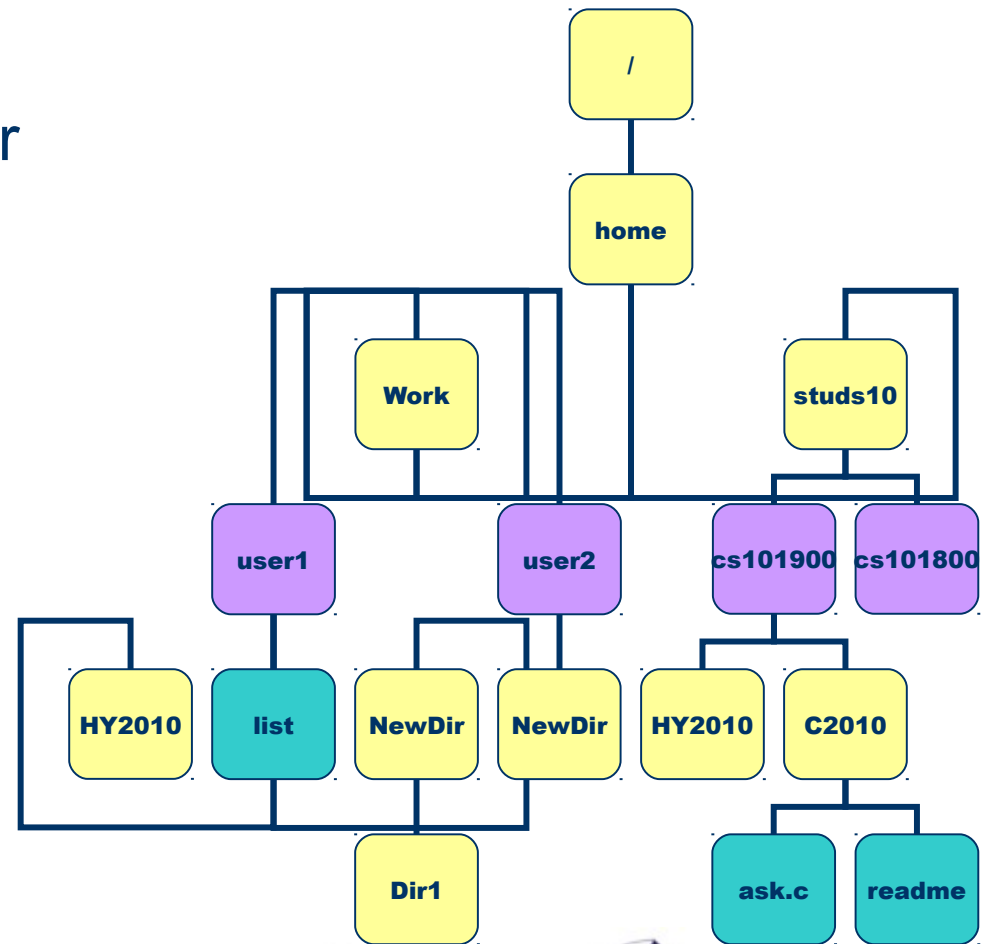
/home/Work/user1

\$ **rmdir NewDir**

<Μήνυμα λάθους!>

\$ **rmdir TestDir**

\$ **rmdir -p NewDir/Dir1**



Παράδειγμα rmdir

- Βρίσκομαι στο
/home/Work/user1/NewDir

\$ cd ..

\$ pwd

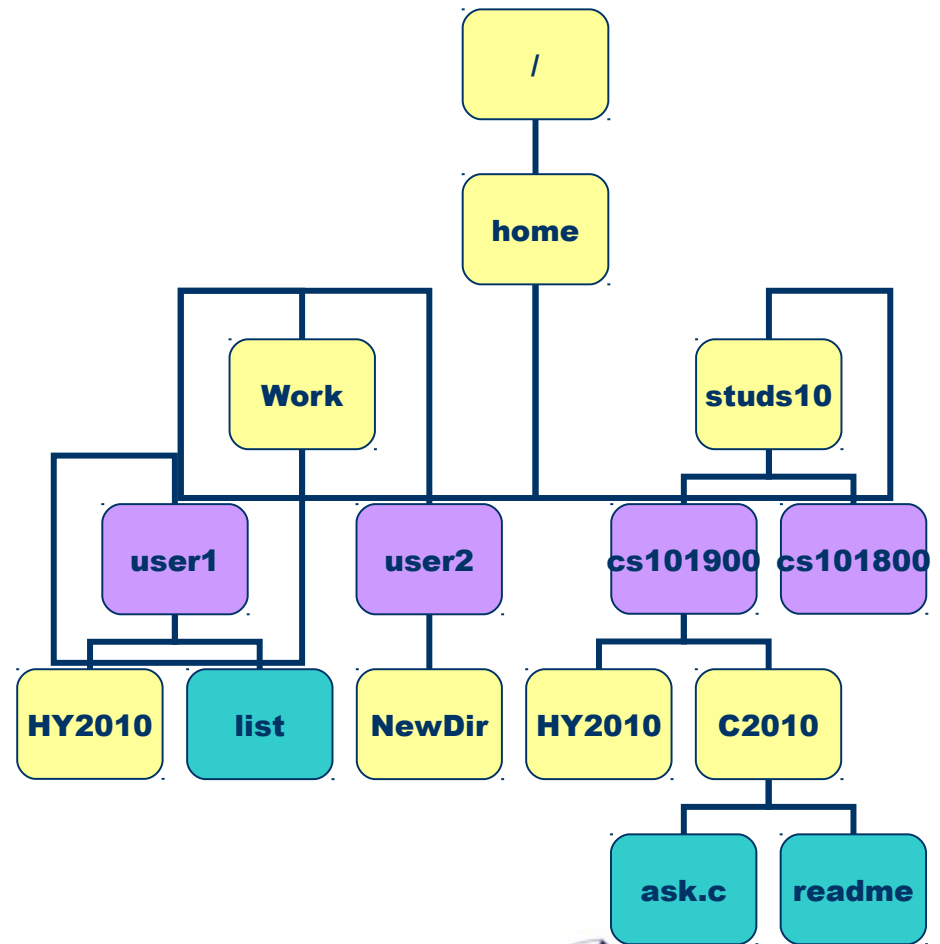
/home/Work/user1

\$ **rmdir NewDir**

<Μήνυμα λάθους!>

\$ **rmdir TestDir**

\$ **rmdir -p NewDir/Dir1**



Δημιουργία αρχείων

- Με κειμενογράφο
 - kwrite gedit, pico, vi
 - `$ kwrite <file>`
 - `$ gedit <file>`
 - `$ pico <file>`
 - `$ vi <file>`
 - `$ nedit <dirname>/<file>`
- Με εντολή
 - cat, touch

**Το αρχείο δημιουργείται
στον τρέχοντα κατάλογο**

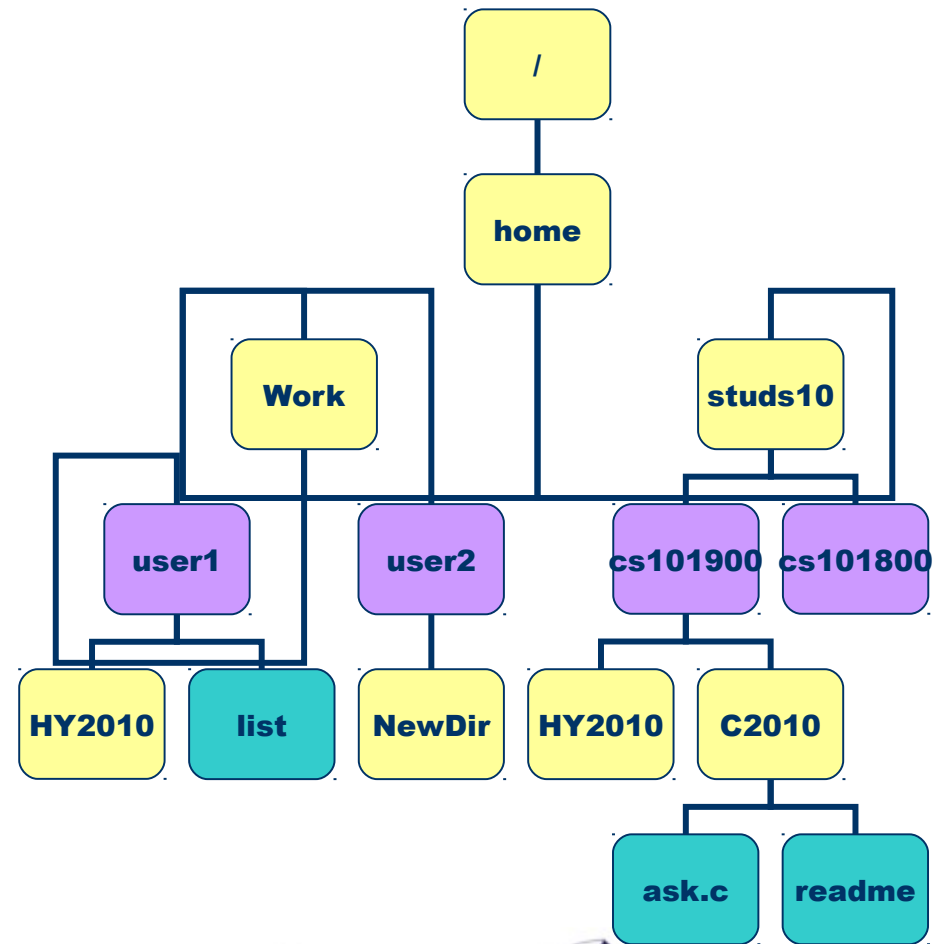
**Το αρχείο δημιουργείται
στον κατάλογο <dirname>**

Δημιουργία αρχείων – Η εντολή touch

- Σύνταξη
 - touch [-a] [-m] <file> . . .
- Περιγραφή
 - Η εντολή προορίζεται να αλλάζει το χρόνο (ημερομηνία/ώρα) δημιουργίας ή πρόσβασης ή τροποποίησης του αρχείου. Αν το αρχείο δεν υπάρχει το δημιουργεί. Ο χρόνος του αρχείου γίνεται ο τρέχων χρόνος του συστήματος
- Παράμετροι
 - -a Αλλάζει τον χρόνο πρόσβασης
 - -m Αλλάζει τον χρόνο τροποποίησης
- Παράδειγμα
 - touch file1
 - touch Dir1/file1

Παράδειγμα touch

- Βρίσκομαι στο
/home/Work/user1
\$ touch file1

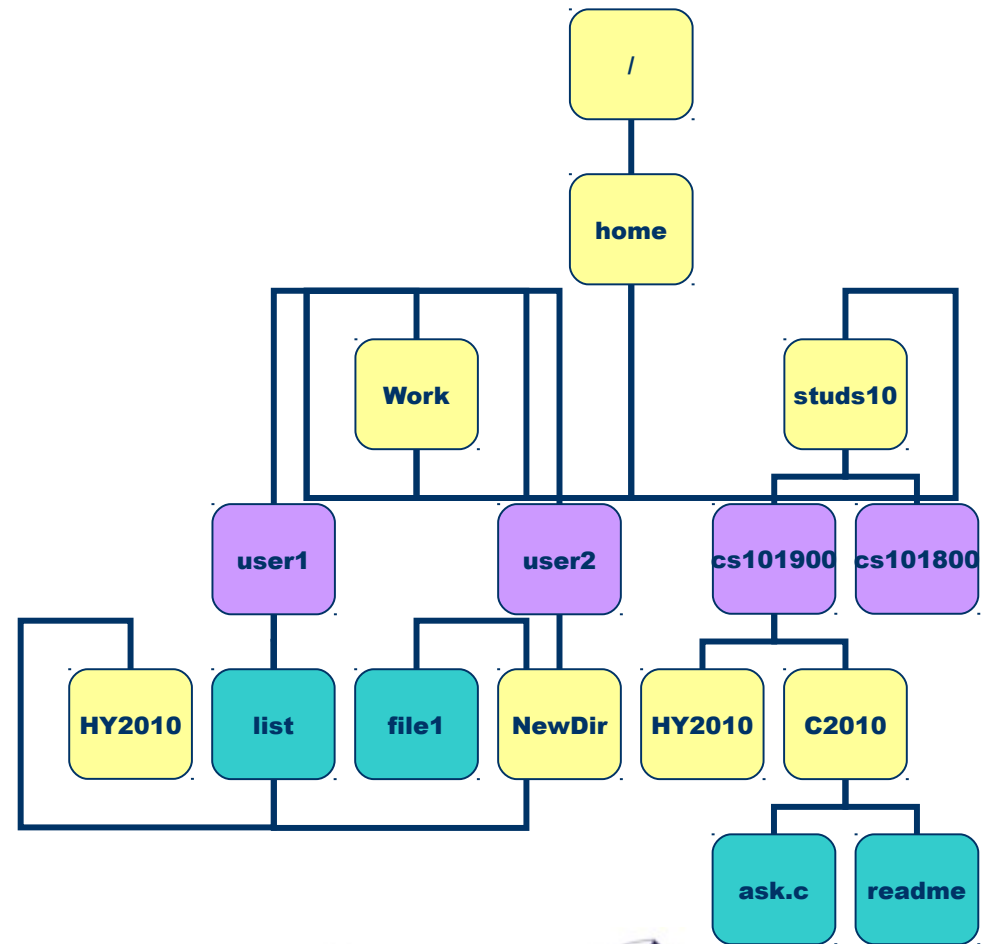


Παράδειγμα touch

- Βρίσκομαι στο
/home/Work/user1

\$ touch file1

\$ touch HY2010/file1

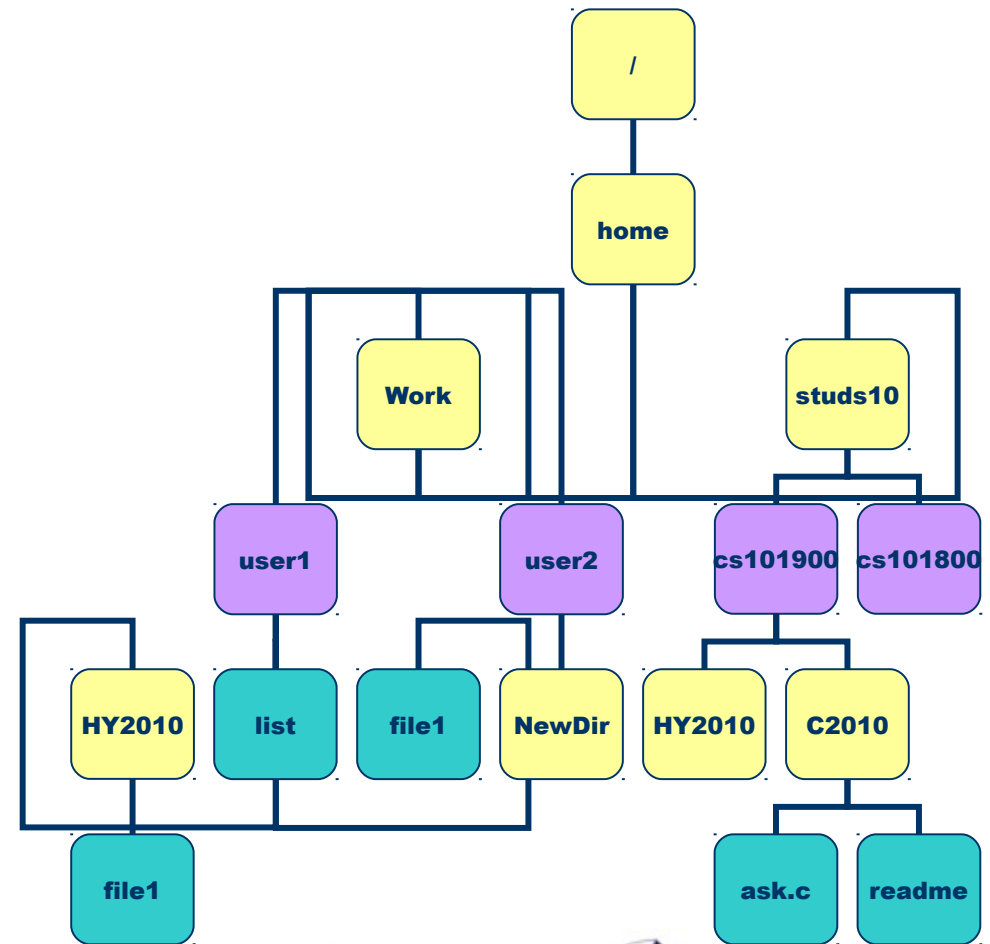


Παράδειγμα touch

- Βρίσκομαι στο
/home/Work/user1

\$ touch file1

\$ touch HY2010/file1



Δημιουργία αρχείων – Η εντολή cat

- Σύνταξη
 - `cat > <file>`
- Περιγραφή
 - Δημιουργία του αρχείου <file>. Ότι πληκτρολογήσει ο χρήστης μέχρι να πατήσει <CTRL + D> γράφεται στο αρχείο <file>
- Παράδειγμα

```
$ cat > file1
Molis dimiourgisa ena
arxeio keimenou ... <CTRL+D>
$ ls
file1
```

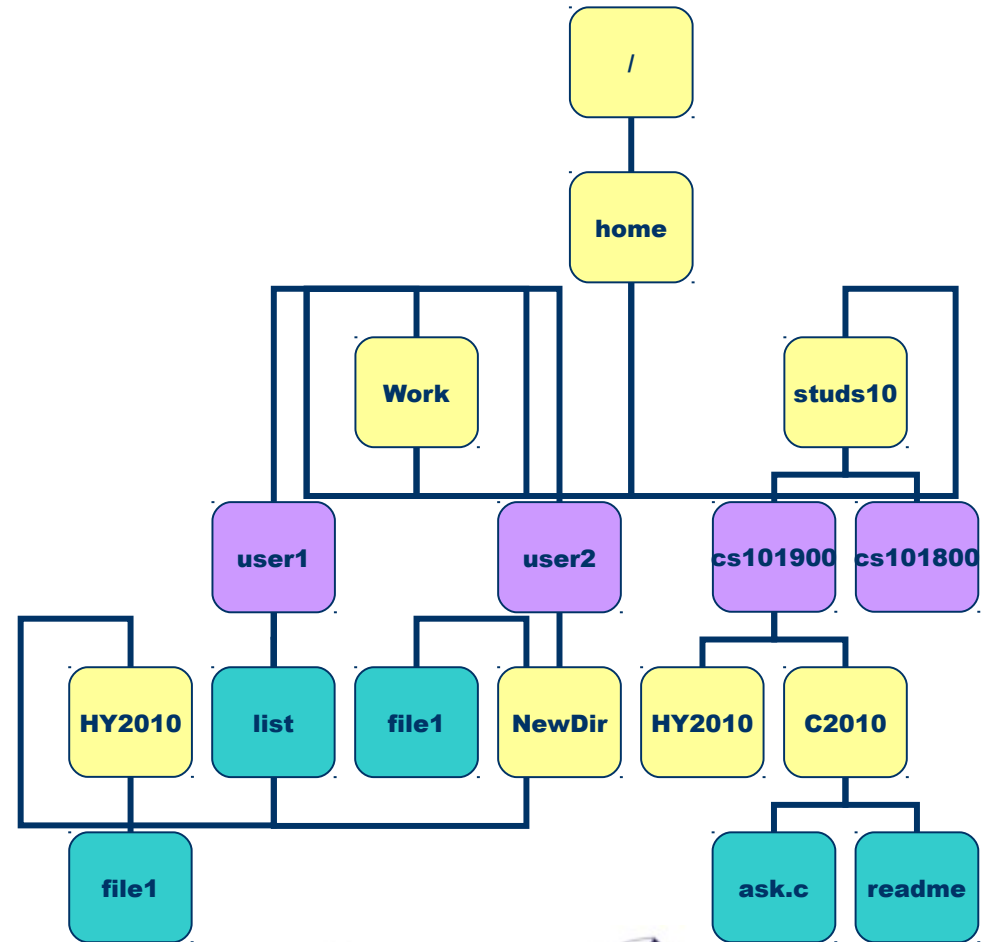
Διαγραφή αρχείων – Η εντολή rm

- Σύνταξη
 - `rm [-f] [-i] <file> ...`
 - `rm -r [-f] [-i] <dirname>... [<file>...]`
- Περιγραφή
 - Διαγράφει αρχεία και καταλόγους.
- Παράμετροι
 - **-f** «Εξαναγκασμένο» σβήσιμο αρχείων χωρίς να ενημερώνεται ο χρήστης
 - **-i** Σβήσιμο αρχείων που απαιτεί επιβεβαίωση από το χρήστη
 - **-r** Αναδρομικό σβήσιμο αρχείων και καταλόγων
- Παράδειγμα
 - `rm file1`
 - `rm -r Dir1/`

Παράδειγμα rm

- Βρίσκομαι στο
/home/Work/user1

\$ rm file1

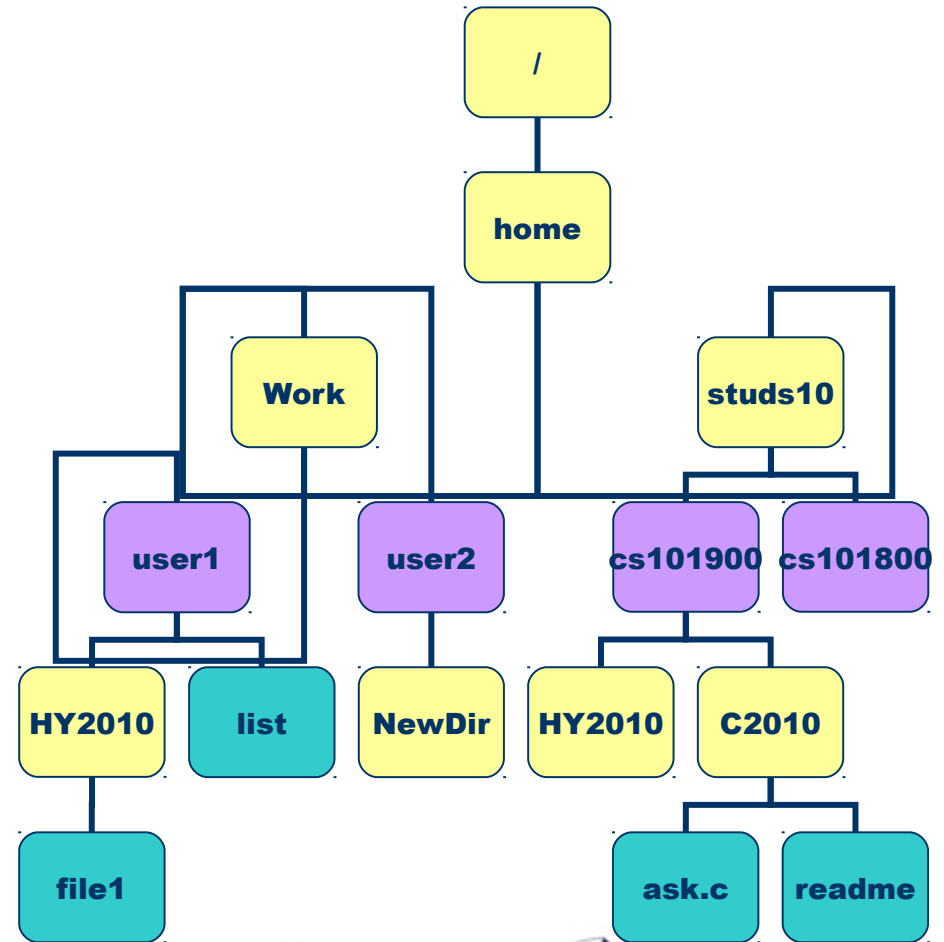


Παράδειγμα rm

- Βρίσκομαι στο
/home/Work/user1

\$ rm file1

\$ rm HY2010/file1

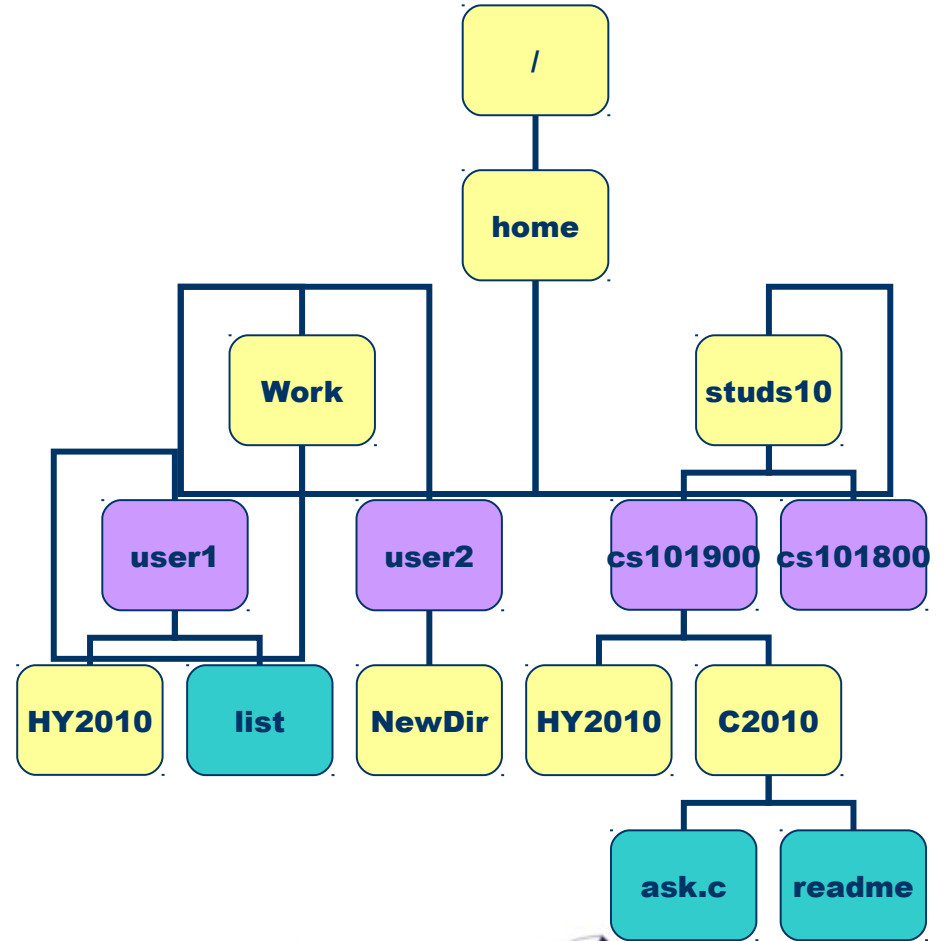


Παράδειγμα rm

- Βρίσκομαι στο `/home/Work/user1`

```
$ rm file1
```

\$ rm HY2010/file1

$$r_{\text{m-r}} \sim 10^{19} \text{ cm}$$


Παράδειγμα rm

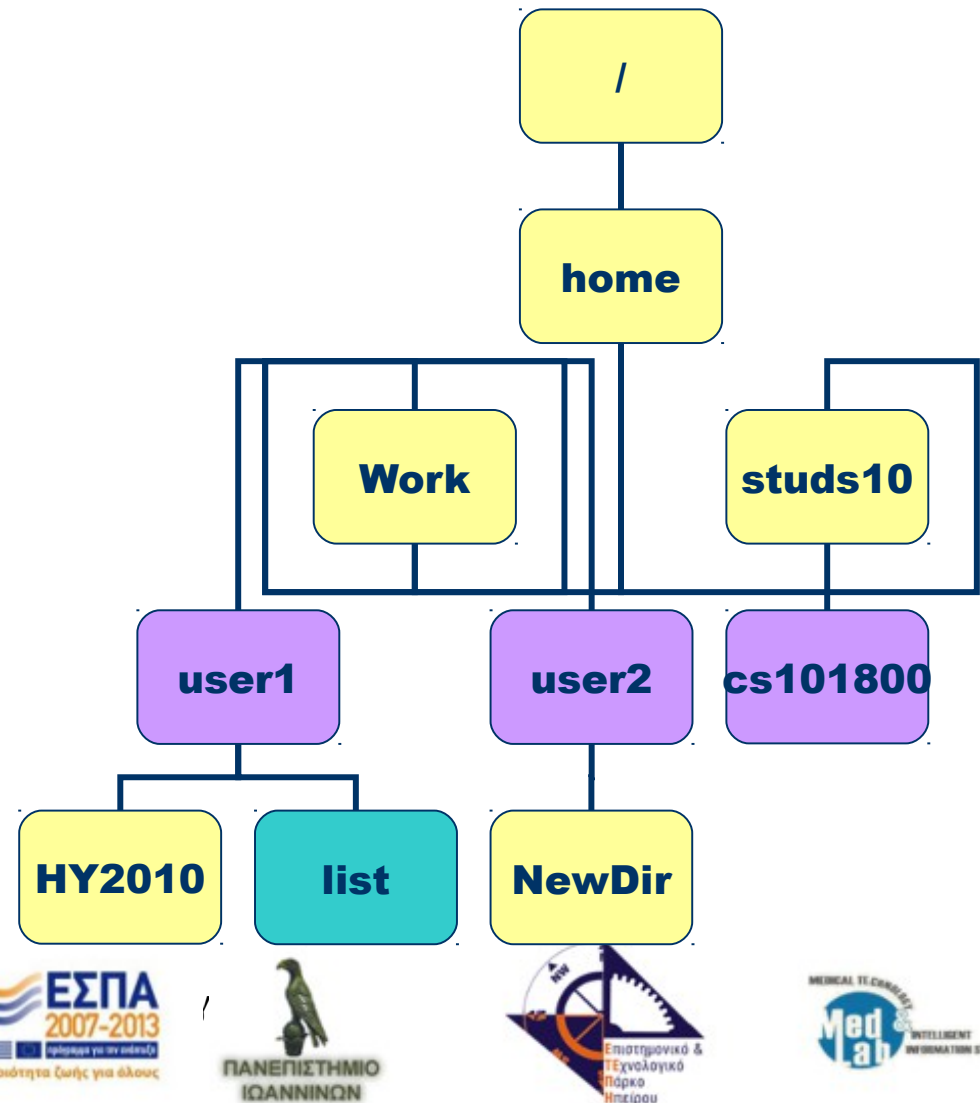
- Βρίσκομαι στο
/home/Work/user1

\$ rm file1

\$ rm HY2010/file1

\$ rm -r ~cs101900/

\$ rm file1



Παράδειγμα rm

- Βρίσκομαι στο
/home/Work/user1

\$ rm file1

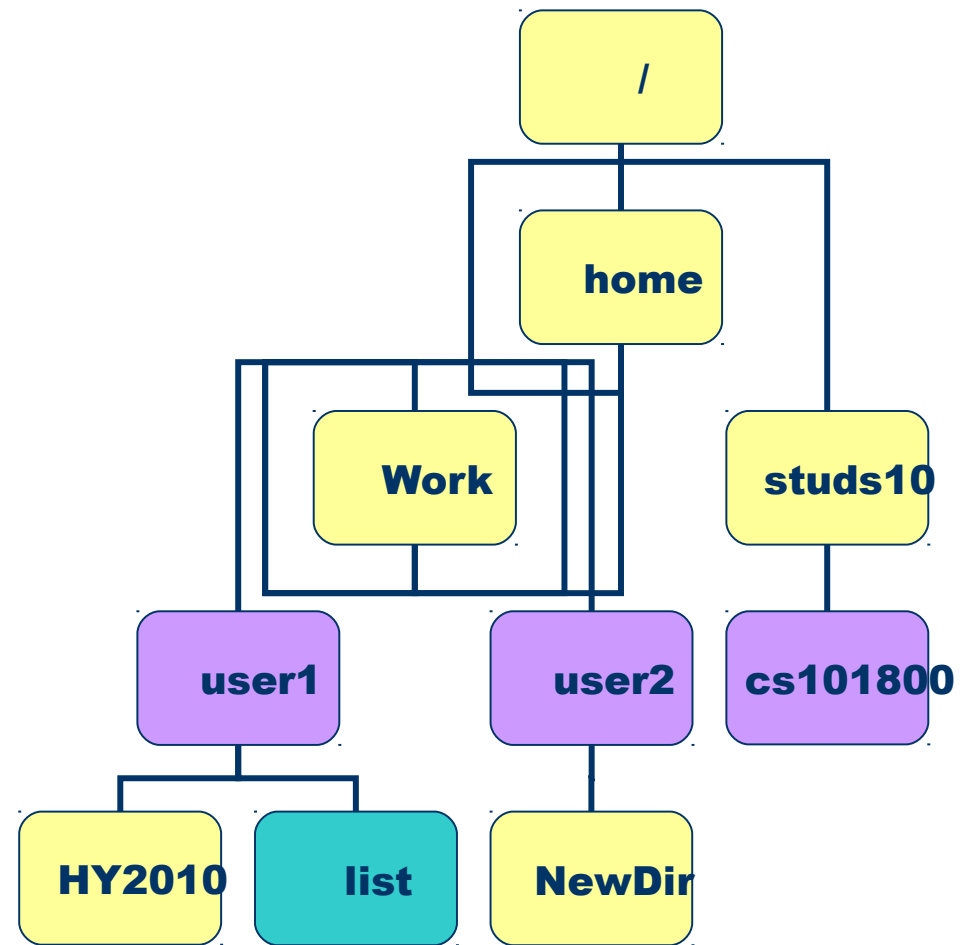
\$ rm HY2010/file1

\$ rm -r ~cs101900/

\$ rm file1

<Μήνυμα λάθους! Δεν
υπάρχει το file1>

\$



Αντιγραφή αρχείων – Η εντολή cp

- Σύνταξη
 - `cp [-r][-i] <file1>[<file2>] <target>`
- Περιγραφή
 - Το/τα αρχείο/α <file1> αντιγράφεται στην θέση <target>
 - Αν το <target> ορίζει κατάλογο το όνομα του αντιγράφου θα είναι <file1>
 - Αν το <target> περιέχει στο τέλος και όνομα αρχείου, γίνεται και μετονομασία
- Παράμετροι
 - **-r** Αναδρομική αντιγραφή καταλόγων
 - **-i** Το σύστημα ρωτά το χρήστη πριν την αντιγραφή
- Παράδειγμα
 - `cp file1 file2`
 - `cp file1 file2 Dir1/`

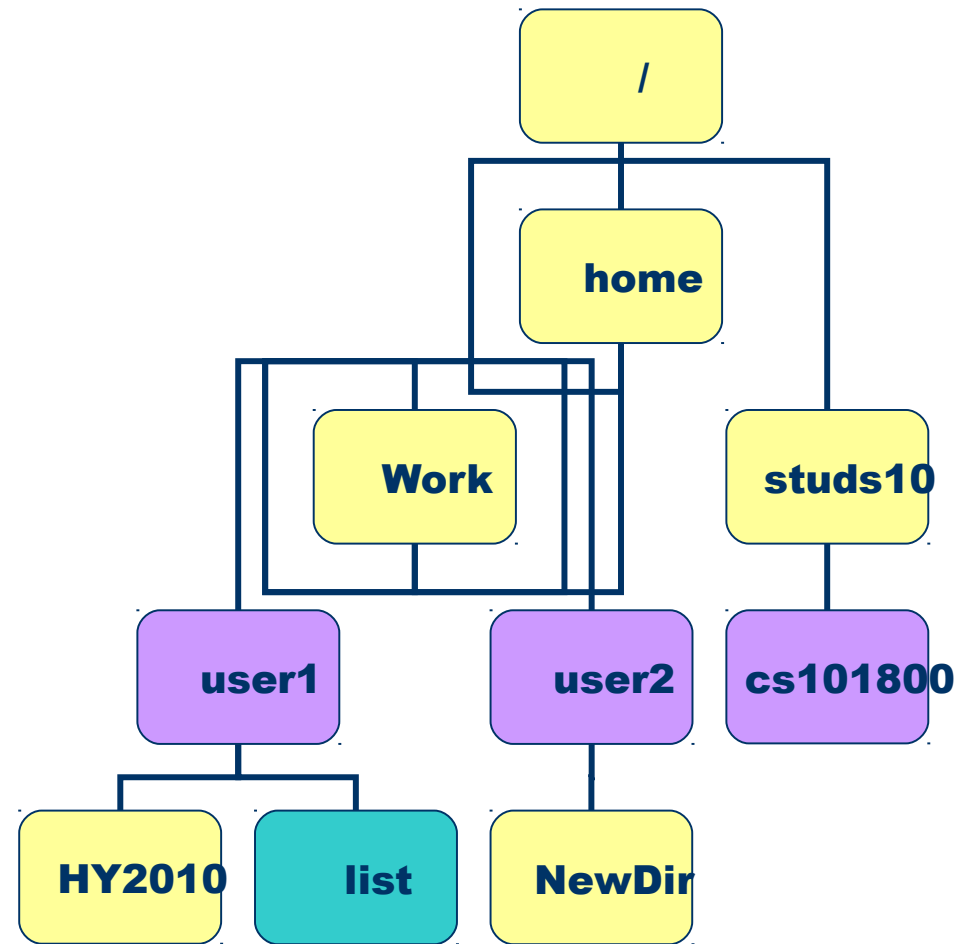
Αντιγραφή αρχείων – Η εντολή cp

- Με την εντολή cp μπορώ:
 - Να δημιουργήσω ένα αντίγραφο αρχείο στον ίδιο κατάλογο (cp file1 file2)
 - Να δημιουργήσω αντίγραφα αρχείων σε διαφορετικούς καταλόγους (cp Dir1/file1 Dir2/file2)
 - Να δημιουργήσω αντίγραφο με το ίδιο όνομα (cp file1 Dir2)
 - Να αντιγράψω αναδρομικά καταλόγους σε άλλες θέσεις στο σύστημα (cp -r Dir1 Dir2)
 - Αν το Dir2 υπάρχει, το Dir1 γίνεται υποκατάλογός του
 - Αν το Dir2 ΔΕΝ υπάρχει, τα Dir1 και Dir2 έχουν κοινό γονικό κατάλογο.

Παράδειγμα cp

- Βρίσκομαι στο
/home/Work/user1

\$ cp list list1

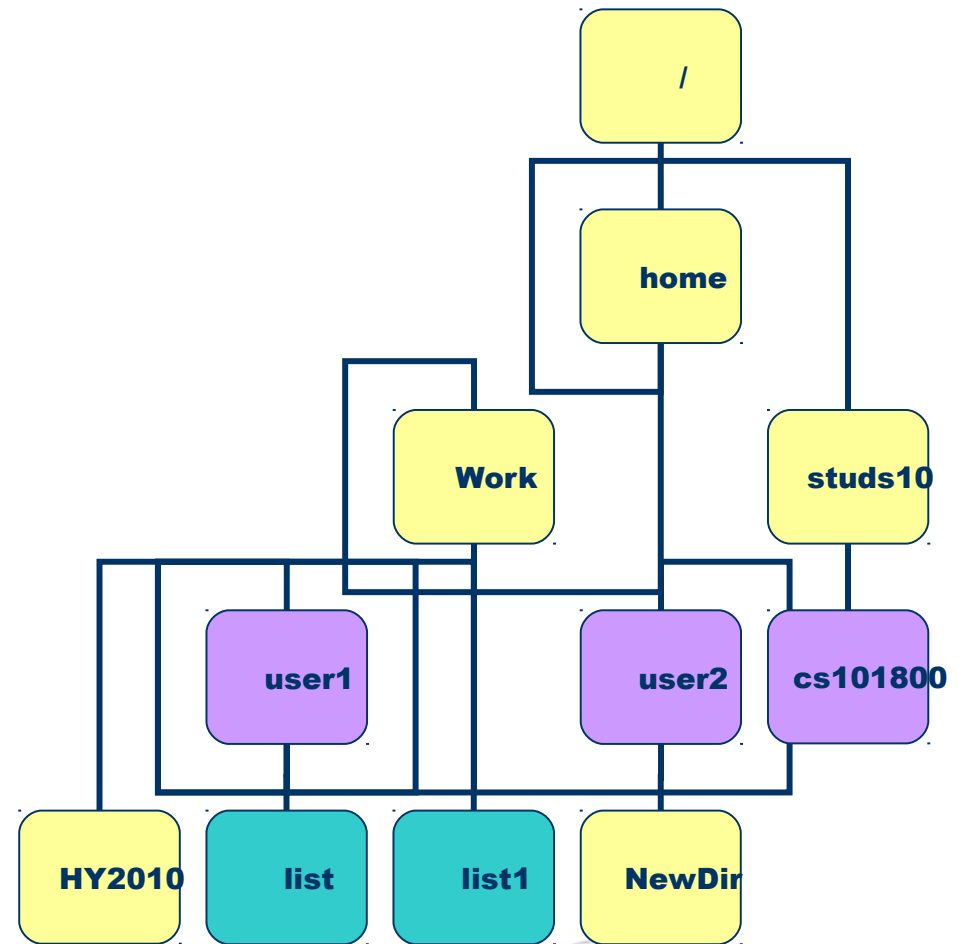


Παράδειγμα cp

- Βρίσκομαι στο
/home/Work/user1

\$ **cp list list1**

\$ **cp list HY2010**



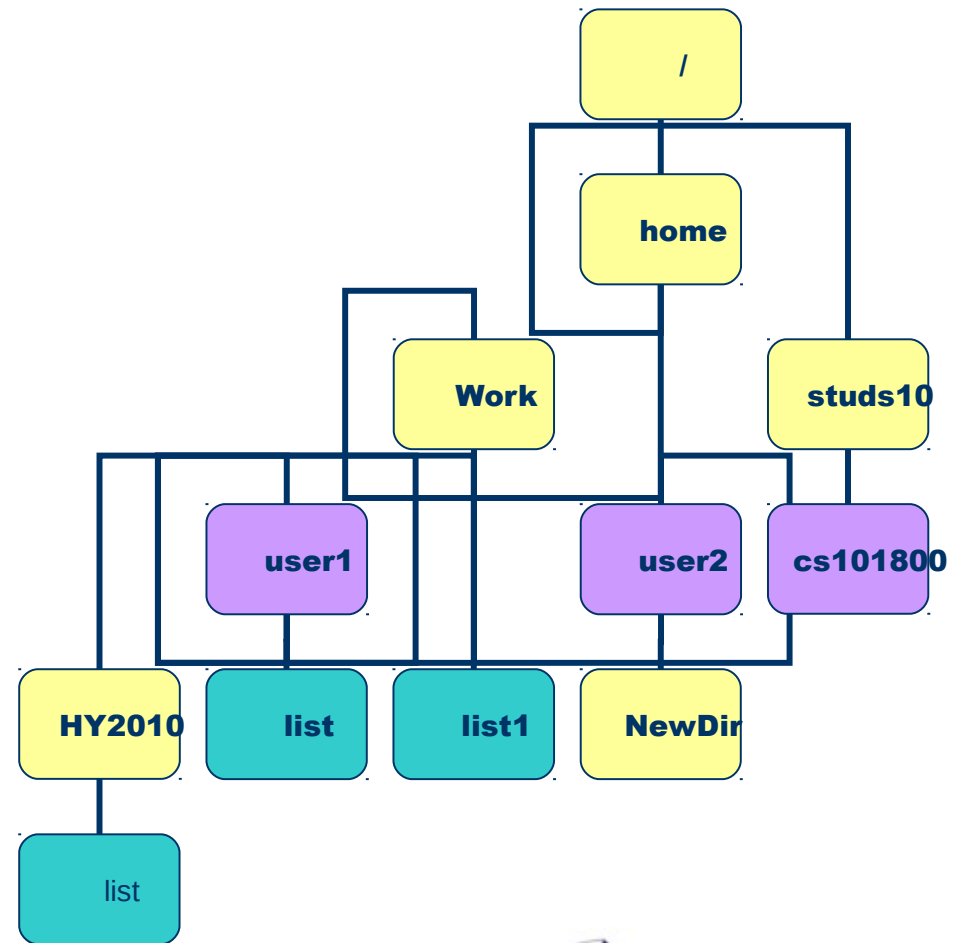
Παράδειγμα cp

- Βρίσκομαι στο
/home/Work/user1

\$ cp list list1

\$ cp list HY2010

\$ cp list HY2010/list1



Παράδειγμα cp

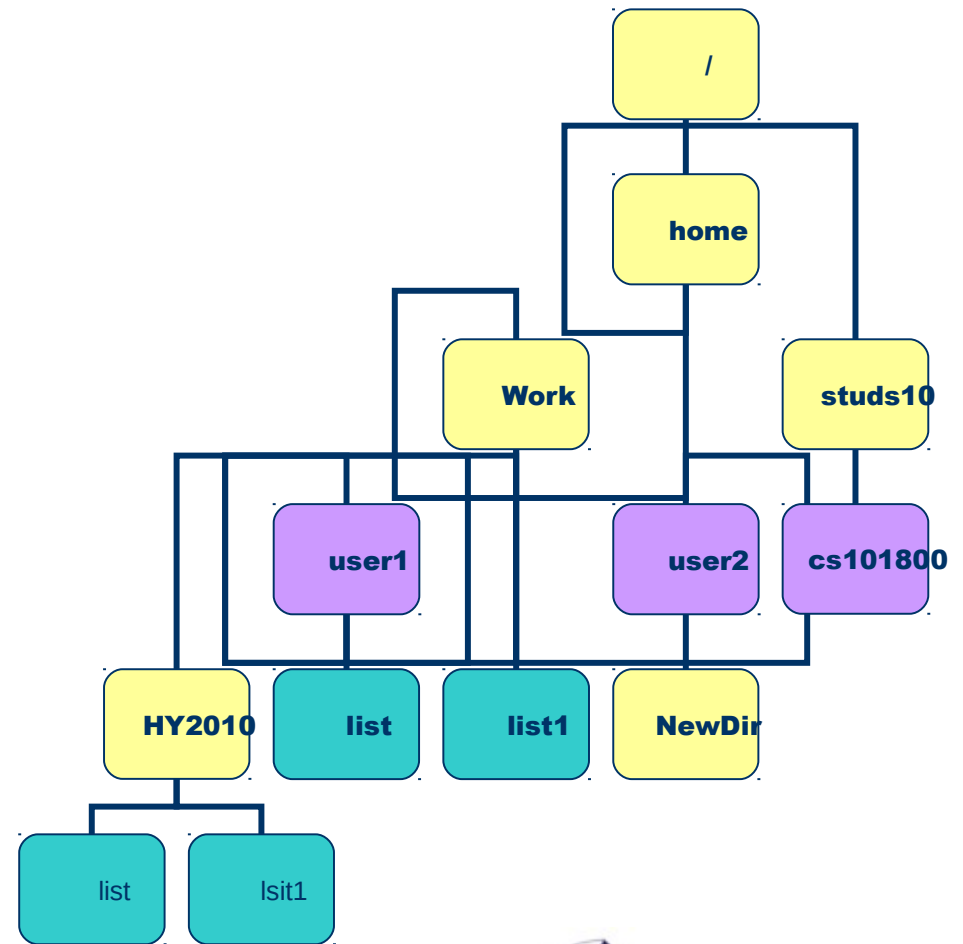
- Βρίσκομαι στο
/home/Work/user1

\$ **cp list list1**

\$ **cp list HY2010**

\$ **cp list HY2010/list1**

\$ **cp -r HY2010 ~user2/**



Παράδειγμα cp

- Βρίσκομαι στο
/home/Work/user1

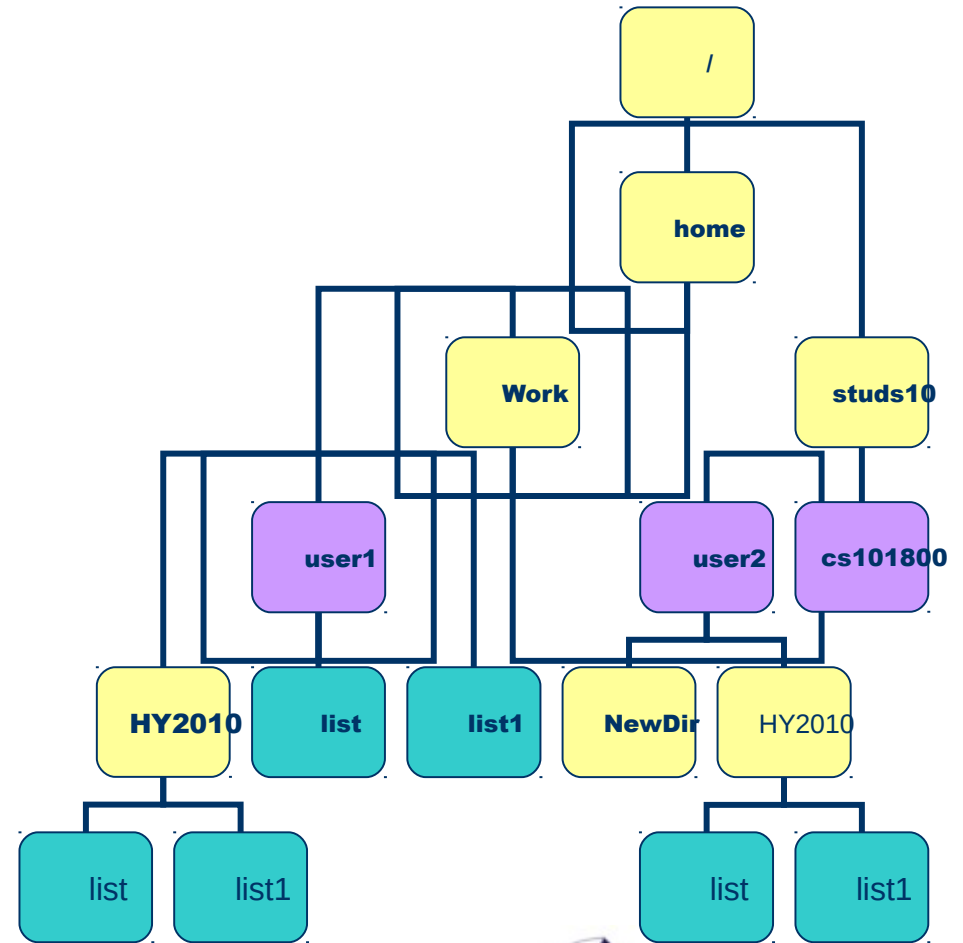
\$ **cp list list1**

\$ **cp list HY2010**

\$ **cp list HY2010/list1**

\$ **cp -r HY2010
/home/Work/user2**

\$ **cp -r HY2010 user2**



Παράδειγμα cp

- Βρίσκομαι στο
/home/Work/user1

\$ **cp list list1**

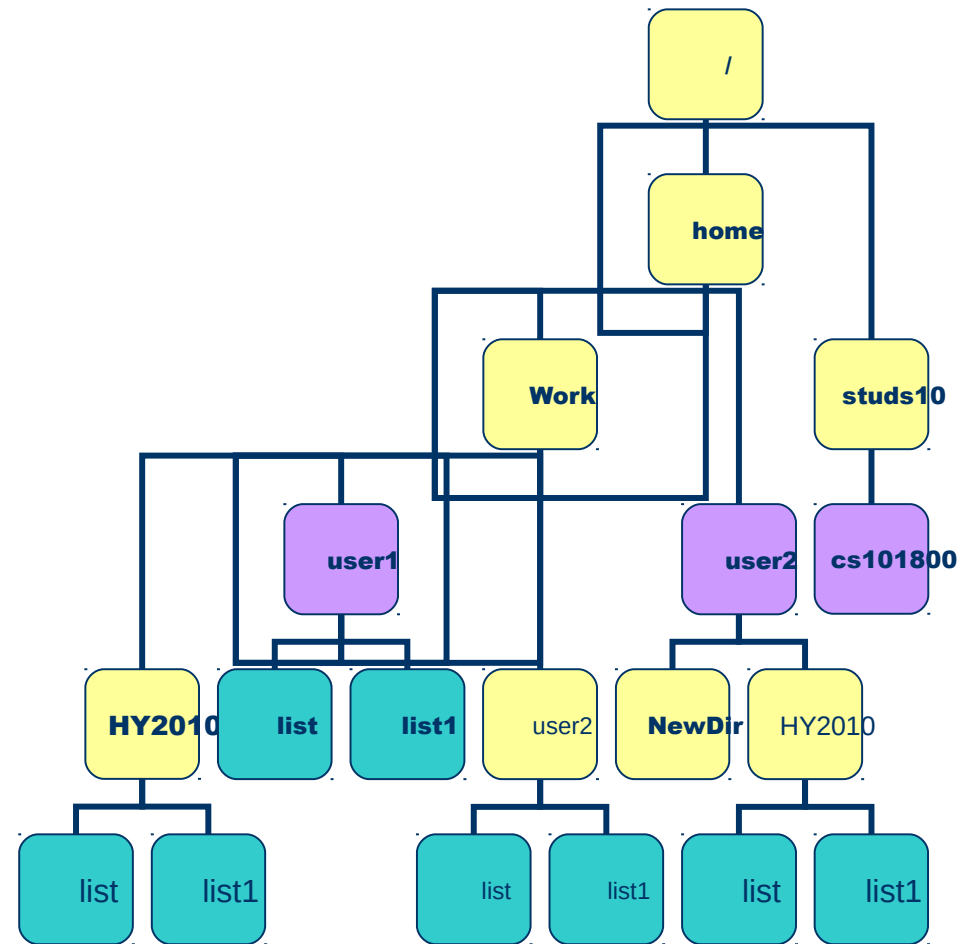
\$ **cp list HY2010**

\$ **cp list HY2010/list1**

\$ **cp -r HY2010**
/home/Work/user2

\$ **cp -r HY2010 user2**

\$ **cp HY2010/list**
../user2/NewDir



Παράδειγμα cp

- Βρίσκομαι στο
/home/Work/user1

\$ cp list list1

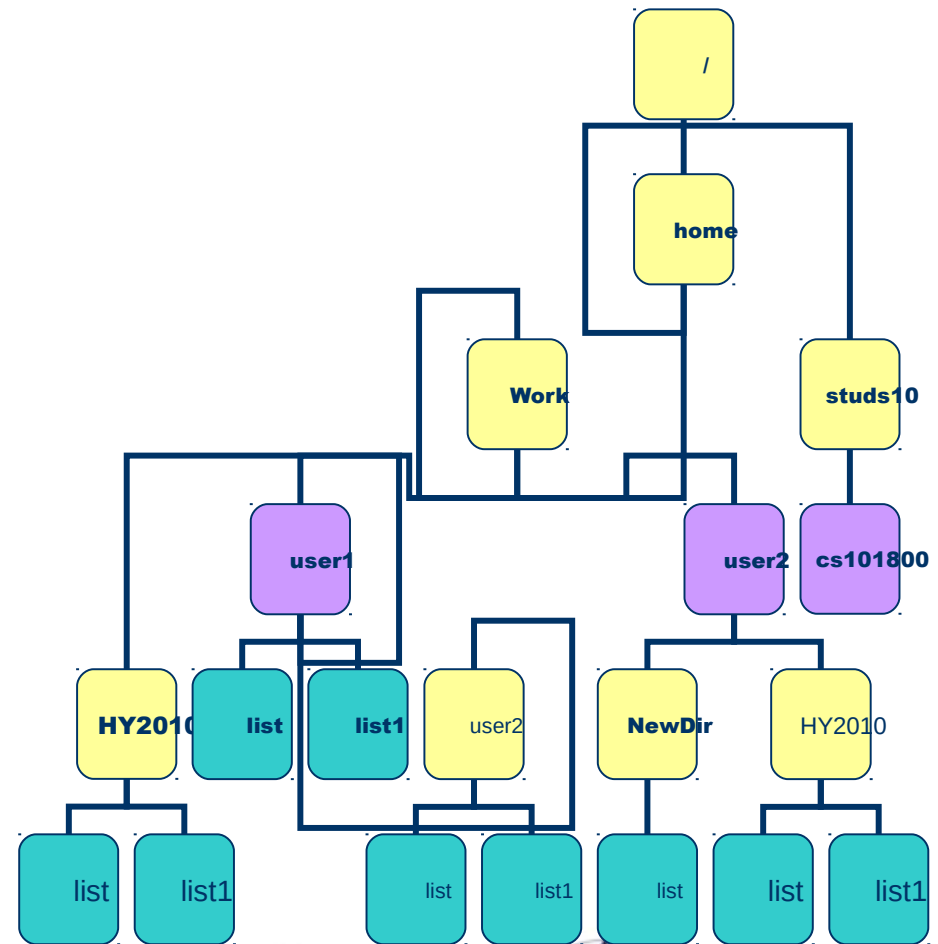
\$ cp list HY2010

\$ cp list HY2010/list1

**\$ cp -r HY2010
/home/Work/user2**

\$ cp -r HY2010 user2

**\$ cp HY2010/list
../user2/NewDir**



Αντιγραφή αρχείων – Η εντολή ln

- Σύνταξη
 - `ln [-s] <file1> <target>`
- Περιγραφή
 - Δημιουργία ειδικού αρχείου συνδέσμου στο αρχείο <file1> με το όνομα <target>
- Παράμετροι
 - **-s** Δημιουργία soft link
- Παράδειγμα
 - `ln file1 file2`
 - `ln -s /home/Work/user1/file1 /home/file1`
 - `ln -s /etc/init.d/httpd my_httpd`
- Οι σύνδεσμοι φαίνονται με την εντολή `ls -l`

Μετακίνηση αρχείων – Η εντολή mv

- Σύνταξη
 - `mv [-i] <file1>[<file2>] <target>`
- Περιγραφή
 - Το/τα αρχείο/α <file1> μετακινούνται στην θέση <target>
 - Αν το <target> δείχνει στον τρέχοντα κατάλογο η εντολή mv ισοδυναμεί με **μετονομασία**
- Παράμετροι
 - **-i** Το σύστημα ρωτά το χρήστη πριν την μετακίνηση
- Παράδειγμα
 - `mv file1 file2` #Υπάρχει μόνο το file2
 - `mv file1 file2 Dir1/`

Μετακίνηση αρχείων – Η εντολή mv

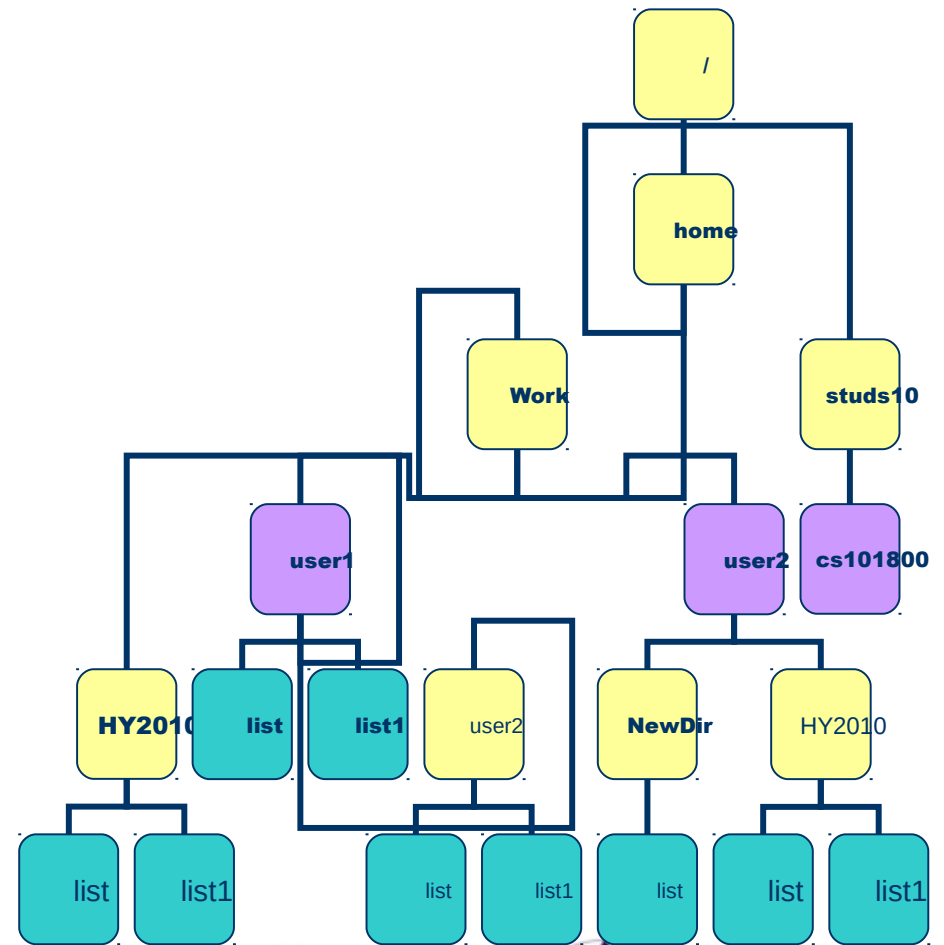
- Με την εντολή mv μπορώ:
 - Να μετακινήσω ένα αρχείο από ένα κατάλογο σε έναν άλλο με άλλο όνομα (`mv Dir1/file1 Dir2/file2`)
 - Να μετακινήσω ένα αρχείο από ένα κατάλογο σε έναν άλλο με το ίδιο όνομα (`mv Dir1/file1 Dir2`)
 - Να μετονομάσω ένα αρχείο ή κατάλογο (`mv file1 file2` ή `mv Dir1 Dir2`) που βρίσκονται στην ίδια θέση
- Δεν μπορώ να κάνω μαζική μετονομασία!!!!

Παράδειγμα mv

- Βρίσκομαι στο
/home/Work/user1

\$ mv list1 list2

\$



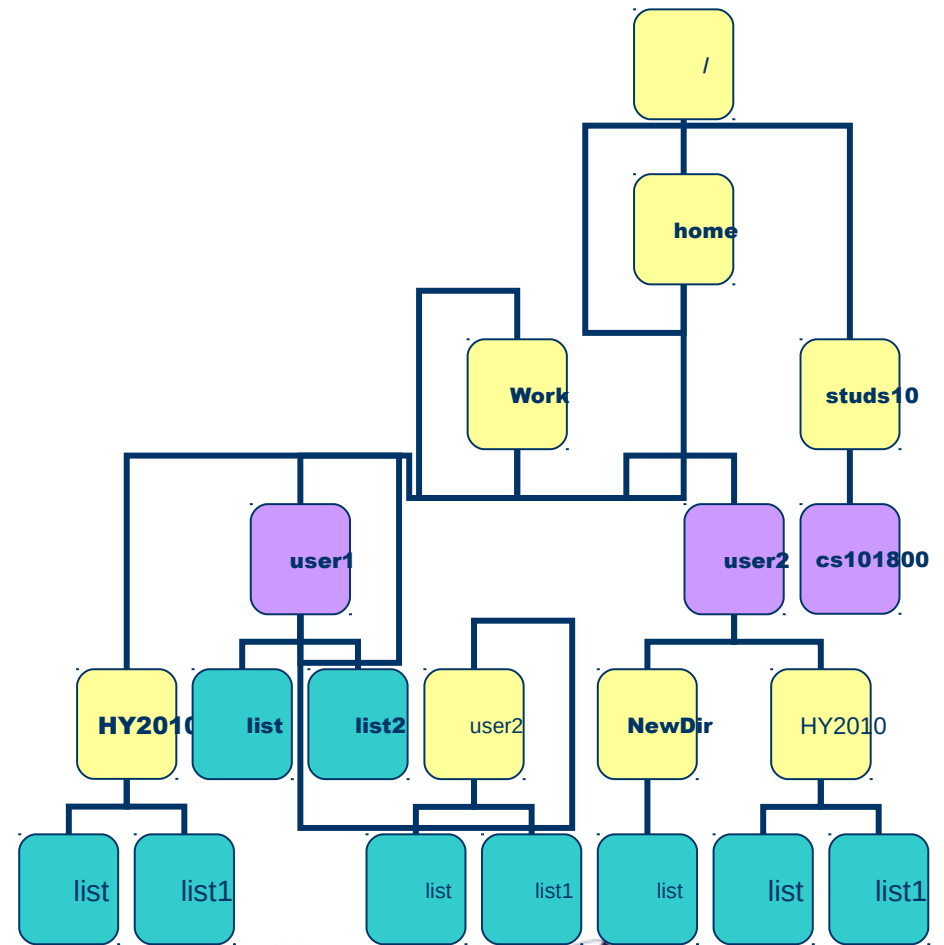
Παράδειγμα mv

- Βρίσκομαι στο
/home/Work/user1

\$ mv list1 list2

\$ mv HY2010 cs415

\$



Παράδειγμα mv

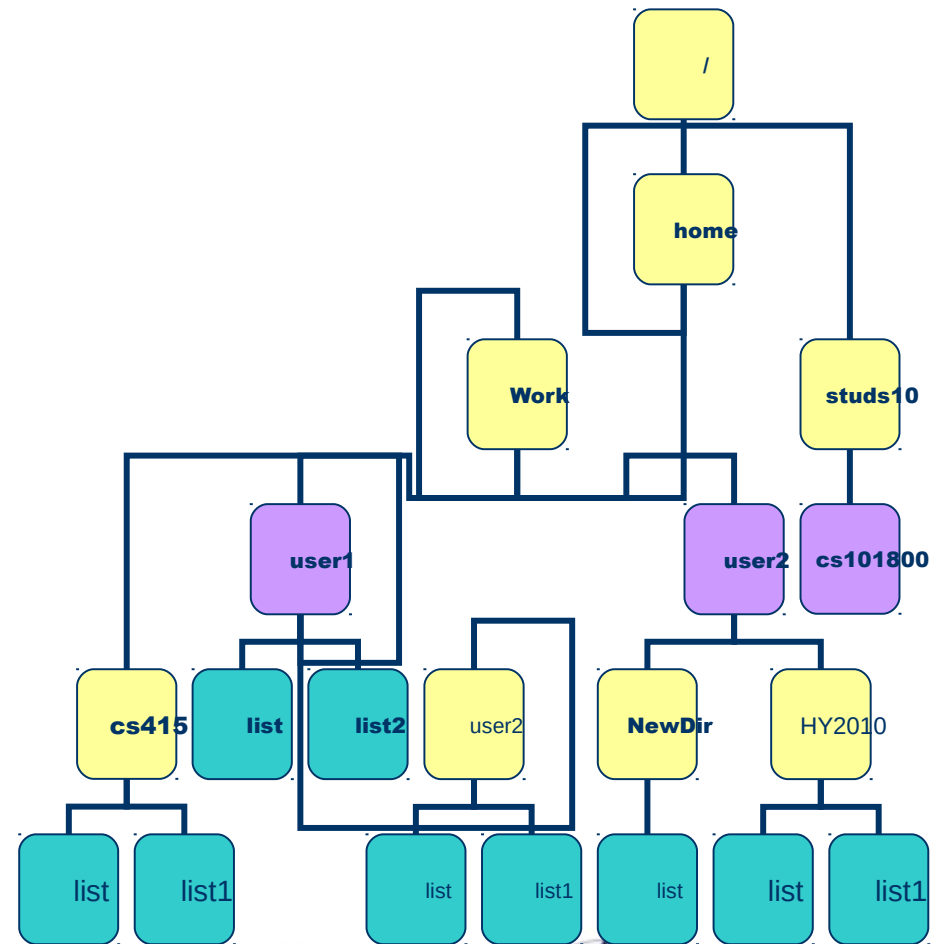
- Βρίσκομαι στο
/home/Work/user1

\$ mv list1 list2

\$ mv HY2010 cs415

\$ mv user2 cs415

\$



Παράδειγμα τῆν

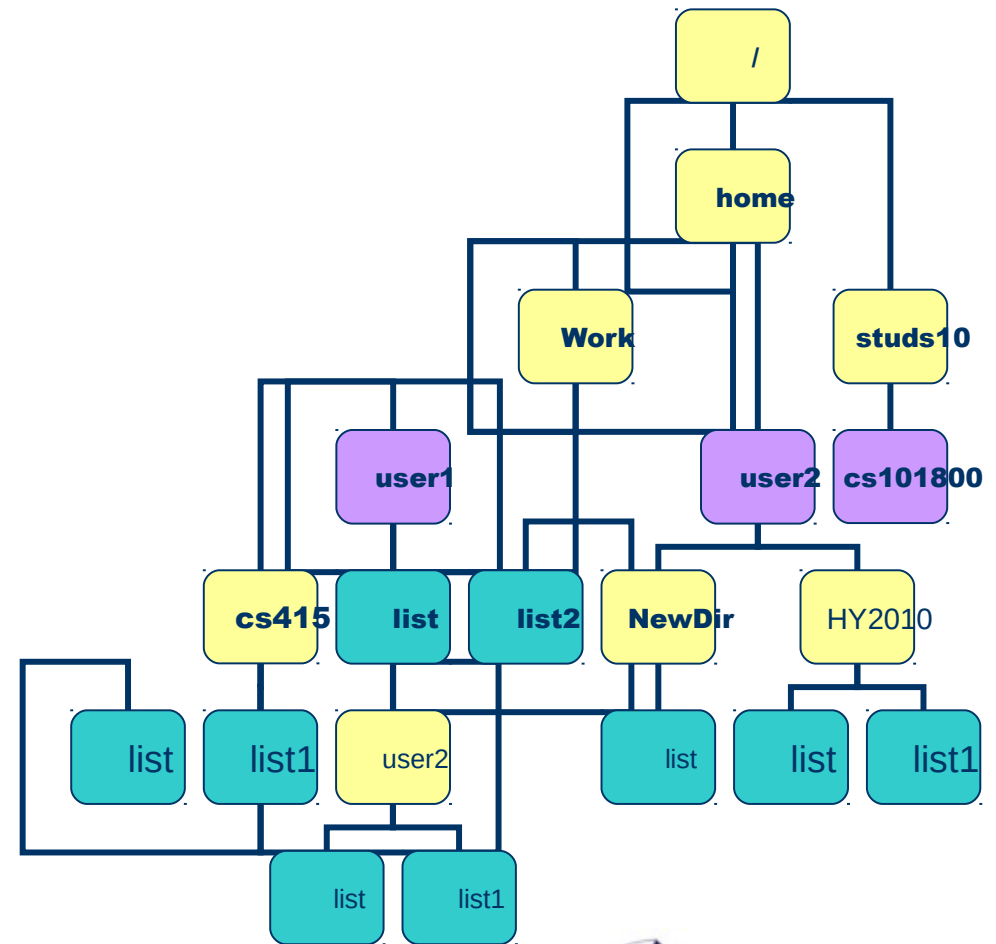
- Βρίσκομαι στο `/home/Work/user1`

\$ mv list1 list2

\$ mv HY2010 cs415

```
$ mv user2 cs415
```

```
$ mv user2/list list2
```



Παράδειγμα mv

- Βρίσκομαι στο
/home/Work/user1

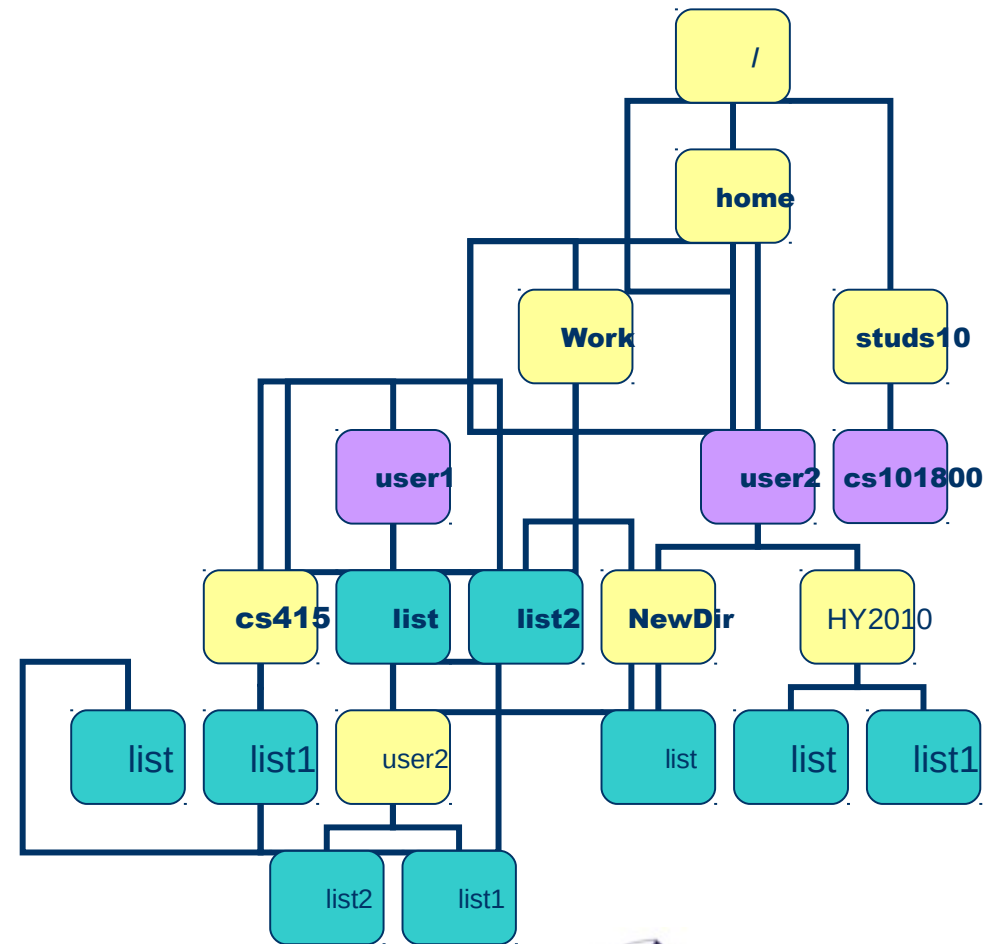
\$ mv list1 list2

\$ mv HY2010 cs415

\$ mv user2 cs415

\$ mv user2/list user2/list2

\$ mv user2/list aris



Παράδειγμα mv

- Βρίσκομαι στο
/home/Work/user1

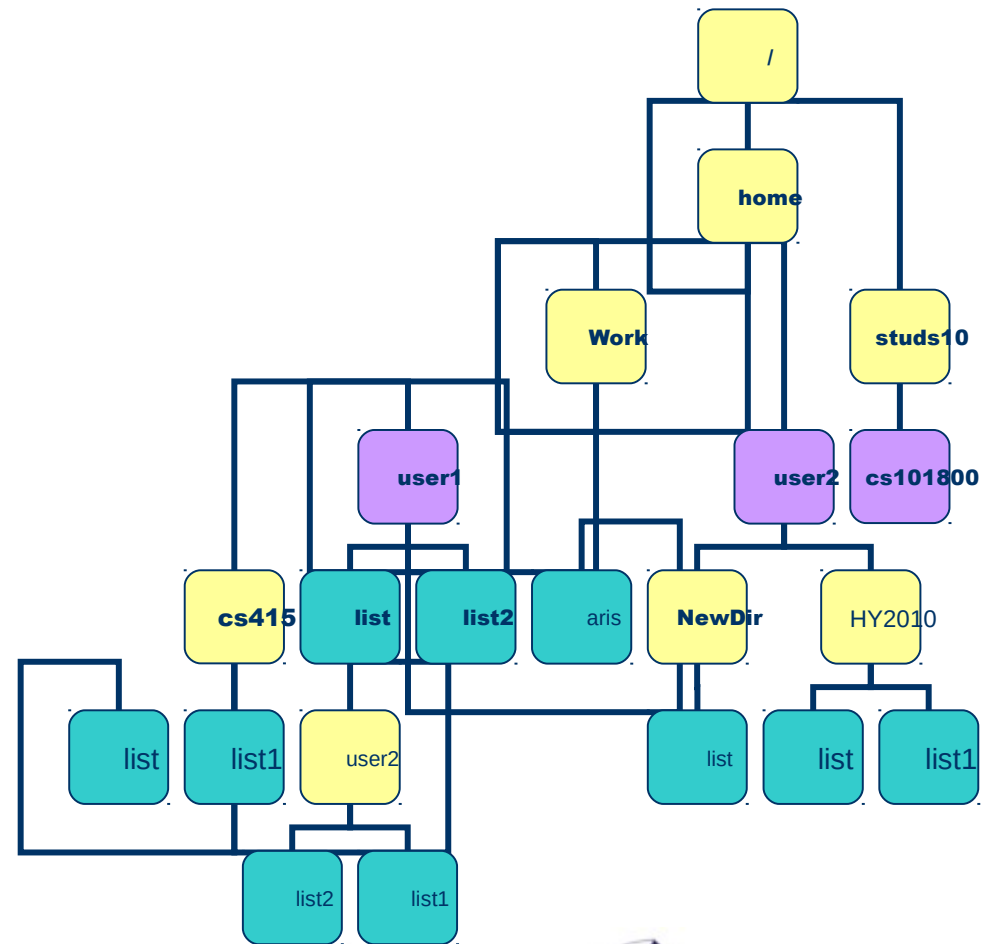
\$ mv list1 list2

\$ mv HY2010 cs415

\$ mv user2 cs415

\$ mv user2/list user2/list2

\$ mv user2/list aris



Μεταχαρακτήρες

- Ειδικοί χαρακτήρες με ειδικό νόημα

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

- Χρησιμοποιούνται για να συντομεύουν τις εντολές και να διευκολύνουν το χρήστη

** Ταιριάζει μηδέν ή περισσότερους χαρακτήρες*

```
$ ls a*
```

Εκτύπωσε όλα τα αρχεία που ξεκινούν από «a»

```
$ ls a*z
```

Εκτύπωσε όλα τα αρχεία που ξεκινούν από «a» και τελειώνουν σε «z»

```
$ cp a* Dir1
```

Αντέγραψε όλα τα αρχεία που ξεκινούν από «a» στον κατάλογο Dir1

```
$ rm junk*
```

Διέγραψε όλα τα αρχεία που ξεκινούν από «junk»

Μεταχαρακτήρας «?»

- Ταιριάζει έναν οποιοδήποτε χαρακτήρα

```
$ ls a?
```

Εκτύπωσε όλα τα αρχεία με όνομα δυο χαρακτήρων που ξεκινούν από «a» (πχ. a1, a2, ab, aa...)

```
$ ls a?z
```

Εκτύπωσε όλα τα αρχεία με όνομα 3 χαρακτήρων ξεκινούν από «a» και τελειώνουν σε «z» (πχ. a1z, a2z, abz...)

```
$ cp a? Dir1
```

Αντέγραψε όλα τα αρχεία με όνομα 2 χαρακτήρων που ξεκινούν από «a» στον κατάλογο Dir1 (πχ. a1, a2, ab, aa)

```
$ rm junk*
```

Διέγραψε όλα τα αρχεία που ξεκινούν από «junk»



Ευρωπαϊκό Ταμείο
Περιφερειακής
Ανάπτυξης



Όλα είναι δυνατά
Επικρατικό Πρόγραμμα
"Ψηφιακή Σύγκλιση"



Διασυνδέοντας την Έρευνα και τ



ΕΣΠΑ
2007-2013
Πρόγραμμα για την ανάπτυξη
Ποιότητα ζωής για όλους



Μεταχαρακτήρες «[clist]»

- Ταιριάζει έναν οποιοδήποτε χαρακτήρα από τη λίστα clist. Το clist:
 - $[c_1, c_2, \dots, c_n]$: Ένας από τους χαρακτήρες c_i
 - $[c_1 c_2 \dots c_n]$: Ένας από τους χαρακτήρες c_i
 - $[c_1 - c_2]$: Ένας από τους χαρακτήρες ανάμεσα στο c_1 και στο c_2

```
$ ls a[abc]
```

Εμφάνισε τα αρχεία aa, ab, ac

```
$ ls a[abc,0-9]
```

Εμφάνισε τα αρχεία aa0, ab1, ac4 κλπ

```
$ cp a[a-z][0-9]  
Dir1
```

Αντέγραψε τα αρχεία aa0, az4, ac1, ab6



Ευρωπαϊκή
Περιφέρεια
Ανάπτυξης

rnet

της Έρευνας και Τ



Ποιότητα ζωής για όλους



Μεταχαρακτήρες «[!]»

- Ταιριάζει έναν οποιοδήποτε χαρακτήρα ΕΚΤΟΣ από αυτούς που βρίσκονται τη λίστα clist.

```
$ ls a[!abc]
```

Εμφάνισε τα αρχεία ae, az, ak

```
$ ls a[!0-9]
```

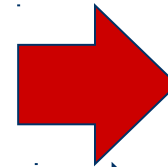
Εμφάνισε τα αρχεία aa, ab, ar κλπ

```
$ cp a[!0-9][!0-9]  
Dir1
```

Αντέγραψε τα αρχεία που δεν έχουν αριθμούς στον δεύτερο και τρίτο χαρακτήρα

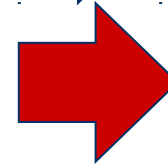
Συνδυασμός μεταχαρακτήρων

```
$ ls a?[123]
```



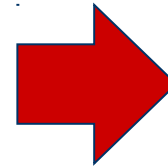
aa1, au3, az2

```
$ ls data[89][12]
```



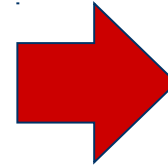
data81, data91, data82

```
$ ls mail?[xy]*
```



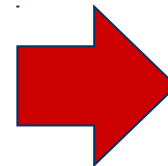
mailxx1234, mail8y

```
$ ls [a-z,A-Z]*.c
```



arxeio.c, ask.c, Ask.c

```
$ ls [a-z,A-Z]*.???
```



**arxeio.txt,
user1.exe, mano.doc**

Περιεχόμενα διάλεξης

- Προβολή περιεχομένων αρχείων
 - Εντολή cat
 - Εντολή more
 - Εντολή less
 - Εντολή head
 - Εντολή tail
- Πληροφορίες για αρχεία
 - Εντολή file
 - Εντολή wc
- Αναζήτηση σε αρχεία
 - Εντολή grep
 - Εντολή find

Εντολή cat

- Σύνταξη
 - `cat [-n] <file> . . .`
- Περιγραφή
 - Ενώνει αρχεία και τυπώνει τα περιεχόμενά τους στην οθόνη
- Παράμετροι
 - **-n** Τυπώνει και τον αριθμό γραμμής
- Παραδείγματα
 - `cat file1`
 - `cat file1 file2 file3`
 - `cat -n file1`

Παραδείγματα cat

```
bash-2.05$ cat a.c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
main()
{
    char line[30];
    FILE *fp;

    fp = fopen("all_words.txt", "r");

    while (fgets(line, 30, fp) != NULL)
    {
        if (strlen(line) <= 7)
        {
            printf("%s", line);
        }
    }
}
bash-2.05$
```

```
bash-2.05$ cat -n a.c
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  main()
5  {
6      char line[30];
7      FILE *fp;
8
9
10     fp = fopen("all_words.txt", "r");
11
12     while (fgets(line, 30, fp) != NULL)
13     {
14         if (strlen(line) <= 7)
15         {
16             printf("%s", line);
17         }
18     }
19
20 }
bash-2.05$
```

Παραδείγματα cat (2)

```
bash-2.05$ cat a.c file.txt
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
main()
{
    char line[30];
    FILE *fp;

    fp = fopen("all_words.txt", "r");

    while (fgets(line, 30, fp) != NULL)
    {
        if (strlen(line) <= 7)
        {
            printf("%s", line);
        }
    }
}
Ayto einai ena arxeio
keimenou. To mono pou periexei
einai keimeno!
bash-2.05$
```

Αρχείο a.c

Αρχείο file.txt



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Ταμείο
Περιφερειακής
Ανάπτυξης



Ψηφιακή Αθήνα
Όλα είναι δυνατά
Εθνικό Πρόγραμμα
"Ψηφιακή Σύγκληση"



Διασυνδέοντας την Έρευνα και τ



ΕΣΠΑ
2007-2013
Πρόγραμμα για την ανάπτυξη
Ποιότητα ζωής για όλους



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΙΩΑΝΝΙΝΩΝ



Επιστημονικό &
Τεχνολογικό
Πάρκο
Ηπείρου



MEDICAL TECHNOLOGY
INTELLIGENT
INFORMATION SYSTEMS

Εντολή more

- Σύνταξη
 - `more [-<αριθμός>] <file> . . .`
- Περιγραφή
 - Προβάλλει τα περιεχόμενα του αρχείου στην οθόνη ανά σελίδα
- Παράμετροι
 - **-<αριθμός>** Κάθε σελίδα αποτελείται από <αριθμός> γραμμές
- Χειρισμός
 - Με το πλήκτρο SPACE αλλάζει σελίδα-σελίδα
 - Με το πλήκτρο ENTER αλλάζει γραμμή-γραμμή
 - Με το πλήκτρο «q» επιστροφή στο κέλυφος
 - ΔΕΝ πηγαίνω σε προηγούμενη σελίδα
- Παραδείγματα
 - `more file1`
 - `more -10 file1`

Παράδειγμα more

```
bash-2.05$ more ask1.c
#include "ask1.h"

FILE *fp;

void SaveCanvas()
{
    rectangle rect;
    point dest_corner={0,0};

    ScreenCanvas=SRGP_inquireActiveCanvas();
    TempCanvas=SRGP_createCanvas(WIN_X_SIZE,WIN_Y_SIZE);
    SRGP_useCanvas(TempCanvas);

    rect.bottom_left.x=0;
    rect.bottom_left.y=0;
    rect.top_right.x=WIN_X_SIZE;
    rect.top_right.y=WIN_Y_SIZE;

    SRGP_copyPixel(ScreenCanvas,rect,dest_corner);
    SRGP_useCanvas(ScreenCanvas);
}
--More-- (3%)
```

Μια σελίδα



Ε
Π
Α



Εντολή less

- Σύνταξη
 - `less <file> . . .`
- Περιγραφή
 - Προβάλλει τα περιεχόμενα του αρχείου στην οθόνη ανά σελίδα, αλλά επιτρέπει και κίνηση στις προς τα πίσω σελίδες/γραμμές.
- Χειρισμός
 - Με το πλήκτρο SPACE αλλάζει σελίδα-σελίδα προς τα εμπρός
 - Με το πλήκτρο «b» αλλάζει σελίδα-σελίδα προς τα πίσω
 - Με το πλήκτρο ENTER αλλάζει γραμμή-γραμμή
 - Με το πλήκτρο «y» αλλάζει γραμμή-γραμμή προς τα πίσω
 - Με το πλήκτρο «q» επιστροφή στο κέλυφος
- Παραδείγματα
 - `less file1`
 - `less file1 file2`

Εντολή head

- Σύνταξη
 - `head [-<αριθμός>] [-n <N>] [-c <C>] <file> . . .`
- Περιγραφή
 - Τυπώνει τις 10 πρώτες γραμμές ενός αρχείου
- Παράμετροι
 - `-<αριθμός>` Τυπώνει τις <αριθμός> πρώτες γραμμές
 - `-n <N>` Τυπώνει τις <N> πρώτες γραμμές
 - `-c <C>` Τυπώνει τα <C> πρώτα bytes
- Παραδείγματα
 - `head file1`
 - `head -20 file1`
 - `head -n 20 file1`
 - `head -c 100 file1`

Παράδειγμα head

```
user1@host:~$ head astar.c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define N 5

int board[N][N] = {
{1, 1, 1, 1, 2},
{1, 2, 2, 2, 2},
{3, 3, 4, 4, 3},
user1@host:~$
```

```
user1@host:~$ head -c 100 astar.c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define N 5

int board[N][N] = {
{1, 1, 1,user1@host:~$
```

```
user1@host:~$ head -n 20 astar.c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define N 5

int board[N][N] = {
{1, 1, 1, 1, 2},
{1, 2, 2, 2, 2},
{3, 3, 4, 4, 3},
{1, 1, 4, 3, 3},
{4, 1, 1, 1, 1}
};

typedef struct prob_list
{
    struct prob_list *parent;
    struct prob_list *next;
    int x, y;
user1@host:~$
```


Εντολή tail

- Σύνταξη
 - `tail [-<αριθμός>] [-n <N>] [-c <C>] <file>`
- Περιγραφή
 - Τυπώνει τις 10 τελευταίες γραμμές ενός αρχείου
- Παράμετροι
 - `-<αριθμός>` Τυπώνει τις <αριθμός> τελευταίες γραμμές
 - `-n <N>` Τυπώνει τις <N> τελευταίες γραμμές
 - `-c <C>` Τυπώνει τα <C> τελευταία bytes
- Παραδείγματα
 - `tail file1`
 - `tail -n 20 file1`
 - `tail -c 100 file1`

Παράδειγμα tail

```
user1@host:~$ tail astar.c
user1@host:~$ tail astar.c
    tmp->F = tmp->h + tmp->g;
    metopo = tmp;
    super_metopo = tmp;

    cc = astar();

    printf("Synolika dimiourgi8ikan
%i paidia \n", cc);
}
user1@host:~$
```

```
user1@host:~$ tail -c 50 astar.c
f("Synolika dimiourgi8ikan %i paidia \n",
cc);
}
user1@host:~$
```

```
user1@host:~$ tail -n 20 astar.c
    final_state->parent = NULL;

    tmp = (LIST *) malloc(sizeof(LIST));
    tmp->x = 1-1;
    tmp->y = 4-1;
    tmp->next = NULL;
    tmp->parent = NULL;
    tmp->g = 0.0;
    tmp->h = state_evaluation(tmp);

    tmp->F = tmp->h + tmp->g;
    metopo = tmp;
    super_metopo = tmp;

    cc = astar();

    printf("Synolika dimiourgi8ikan %i paidia
\n", cc);
}
user1@host:~$
user1@host:~$
```

Εντολή file

- Σύνταξη
 - `file [-f filename] <file>`
- Περιγραφή
 - Τυπώνει τον τύπο του αρχείου.
- Παράμετροι
 - **-f filename** Εξετάζει ένα προς ένα τα αρχεία που βρίσκονται στο αρχείο filename
- Παραδείγματα
 - `file file.txt`
`file.txt: ascii text`
 - `file fuzzy.zip`
`fuzzy.zip: ZIP archive`
 - `file Java`
`Java: directory`

Εντολή wc

- Σύνταξη
 - `wc [-l] [-c] [-w] <file> . . .`
- Περιγραφή
 - Μετράει χαρακτήρες, λέξεις και γραμμές των αρχείων
- Παράμετροι
 - **-c** Αριθμός bytes=χαρακτήρες
 - **-w** Αριθμός λέξεων
 - **-l** Αριθμός γραμμών
- Παραδείγματα
 - `wc -l file1`
 - `wc -cw file1`
 - `wc file1`

Παράδειγμα wc

```
user1@host:~$ wc -l astar.c
357 astar.c
user1@host:~$
```

```
user1@host:~$ wc astar.c
 357   786 6384 astar.c
user1@host:~$
```

```
user1@host:~$ wc -c astar.c
6384 astar.c
user1@host:~$
```

Εντολή grep

- Σύνταξη
 - `grep <pattern> <file> . . .`
- Περιγραφή
 - Αναζητά στο αρχείο <file> γραμμές που να περιέχουν το <pattern>
- Χρήση
 - Το <pattern> μπορεί να είναι μια λέξη
 - Το <pattern> μπορεί να είναι μια πρόταση σε “...”
 - Το <pattern> μπορεί να είναι μια κανονική έκφραση
- Παραδείγματα
 - `grep while ask1.c`
 - `grep "a = 1" file1`
 - `grep "[a,b]" file1`

Κανονικές εκφράσεις στην grep

- `"."` Ταιριάζει έναν οποιοδήποτε χαρακτήρα
- `"[c1, c2, ...cn]"` Ταιριάζει έναν από τους χαρακτήρες c₁, c₂, ..., c_n
- `"[^c1, c2, ...cn]"` Ταιριάζει έναν από τους χαρακτήρες ΕΚΤΟΣ από c₁, c₂, ..., c_n
- `"[c1-c2]"` Ταιριάζει έναν από τους χαρακτήρες c₁ έως c₂
- `"\"` Χαρακτήρας διαφυγής, ότι έπεται είναι κυριολεκτικό
- `"^"` Ταιριάζει την αρχή της γραμμής
- `"expr1 \| expr2"` Ταιριάζει είτε τη μια, είτε την άλλη έκφραση

Παραδείγματα grep

- Αρχείο εισόδου ένας μεγάλος κώδικας σε C (1017 γραμμές)

```
$ grep if grafika.c
```

```
user1@host:~$ grep if grafika.c
if (!XAllocNamedColor (display, colorMap, "red", &color[0], &color[0]))
if (!XAllocNamedColor (display, colorMap, "green", &color[1], &color[1]))
if (!XAllocNamedColor (display, colorMap, "blue", &color[2], &color[2]))
if (!XAllocNamedColor (display, colorMap, "yellow", &color[3], &color[3]))
if (!XAllocNamedColor (display, colorMap, "orange", &color[4], &color[4]))
if (!XAllocNamedColor (display, colorMap, "black", &color[5], &color[5]))
...
if ((getButtonClick (colorBtn[i], xM, yM)) == 1)
    if (colorPalletIsVisible == 0)
    if (report.xbutton.button == Button3)
case MotionNotify:
    if (brushMode == 1)
        // Elegxoume me th Button1Mask, an to MotionNotifyEvent exei proklh8ei
    if ((report.xbutton.state & Button1Mask) != 0){
user1@host:~$
```


Παραδείγματα grep

`$ grep "^ if" grafika.c`

Στην αρχή της γραμμής δύο κενά και μετά if

```
user1@host:~$ grep "^ if" grafika.c
if (!XAllocNamedColor (display, colorMap, "red", &color[0], &color[0]))
if (!XAllocNamedColor (display, colorMap, "green", &color[1], &color[1]))
if (!XAllocNamedColor (display, colorMap, "blue", &color[2], &color[2]))
if (!XAllocNamedColor (display, colorMap, "yellow", &color[3], &color[3]))
if (!XAllocNamedColor (display, colorMap, "orange", &color[4], &color[4]))
if (!XAllocNamedColor (display, colorMap, "black", &color[5], &color[5]))
if (!XAllocNamedColor (display, colorMap, "salmon", &color[6], &color[6]))
...
if (bx - colorMoveX >= btn.x && bx - colorMoveX <= btn.x + btn.width)
if (!(size_hints = XAllocSizeHints ())) ||
if ((display = XOpenDisplay (display_name)) == NULL)
if (XStringListToTextProperty (&window_name, 1, &windowName) == 0)
if (XStringListToTextProperty (&icon_name1, 1, &iconName1) == 0)
if (XStringListToTextProperty (&icon_name2, 1, &iconName2) == 0)
if (XStringListToTextProperty (&icon_name3, 1, &iconName3) == 0)
if ((font_info = XLoadQueryFont (display, "9x15")) == NULL)
user1@host:~$
```

Παραδείγματα grep

`$ grep "while\|for" grafika.c`

Γραμμές που περιέχουν είτε while, είτε for

```
user1@host:~$ grep "while\|for" grafika.c
Colormap colorMap;          /* color map to use for allocating colors. */
XColor color[16];           /* used for allocation of the given color map
for (i = 0; i < 16; i++)
for (i = 0; i < 16; i++)
/* Allocate memory for our structures */
/* Set hints for window manager before mapping window */
fprintf(stderr, "%s: structure allocation for windowName failed.\n",
fprintf(stderr, "%s: structure allocation for iconName failed.\n",
...
        while (1)
            while (1)
        while (1)
        while (1)
        for (i = 0; i < 16; i++)
            // diaforo tou mhdenos tote mporei na zografisei me th bourtsa
user1@host:~$
```

Παραδείγματα grep

\$ grep "[0-9].[0-9]" grafika.c

Βρες αριθμός χαρακτήρας αριθμός

```
user1@host:~$ grep "[0-9].[0-9]" grafika.c
    GRAFIKA 2008
char *icon_name1 = "Giorgos Baggelhs : AM 865";
char *icon_name2 = " 1h Askhsh Grafika 2008 ";
    XMoveWindow (display, colorPallet, width - 100, height);
    strcpy (equBtn.caption, "y=500/x");
x = y = 100;
/* Load a font called "9x15" */
if ((font_info = XLoadQueryFont (display, "9x15")) == NULL)
    fprintf (stderr, "%s: cannot open 9x15 font.\n", appname);
static char *message = "Giorgos Baggelhs : AM 865";
static char *message1 = " 1h Askhsh Grafika 2008 ";
    for (timer = 0; timer < 200000000; timer++)
        printf ("LINE CLICKGO1111!!\n");
        printf ("ARC CLICKGO1111!!\n");
        labs (yM2 - yM), 0, 360 * 64);

user1@host:~$
```



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Ταμείο
Περιφερειακής
Ανάπτυξης



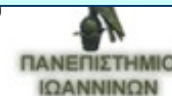
Ψηφιακή
Όλη είναι δύναμη
Επικρατικό Πρόγραμμα
"Ψηφιακή Σύγκληση"



Διασυνδέοντας την Έρευνα και τ



2007-2013
Πρόγραμμα για την υγεία
Ποιότητα ζωής για όλους



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΙΩΑΝΝΙΝΩΝ



Επιστημονικό &
Τεχνολογικό
Πάρκο
Ηπείρου



MEDICAL TECHNOLOGY
INTELLIGENT
INFORMATION SYSTEMS

Παραδείγματα grep

```
$ grep "[0-9]\.[0-9]" grafika.c
```

```
user1@host:~$ grep "[0-9]\.[0-9]" grafika.c
```

```
user1@host:~$
```

Παραδείγματα grep

`grep "([a,b])*" grafika.c`

```
user1@host:~$ grep "([a,b])*" grafika.c
if (b.btnState == PRESSED)
if (b.btnType == EMPTY)
    if (b.caption[0] != ' ')
        strlen (b.caption));
    metablhto (analoga) me tis diastaseis ths
    to xroma pou 8a exoun (an exoun)
strcpy (arcBtn.caption, "ARC");
strcpy (brushBtn.caption, "BRUSH");
draw_button (arcBtn, menuBar, gc);
draw_button (brushBtn, menuBar, gc);
if ((btn.isColorBtn >= 0) && (colorPalletIsVisible == 1))
if (bx - colorMoveX >= btn.x && bx - colorMoveX <= btn.x + btn.width)
    if (by >= btn.y && by <= btn.y + btn.height)
        else if ((getButtonClick (arcBtn, xM, yM)) == 1)
        else if ((getButtonClick (brushBtn, xM, yM)) == 1)
    if (brushMode == 1)
user1@host:~$
```

Παραδείγματα grep

`grep "([^(]*)" grafika.c`

```
user1@host:~$ grep "([^(]*)" grafika.c
void init_color ();
void init_Pallets (GC gc);
void init_buttons ();
void reset_button_state(GC gc);
void draw_button (Button btn, Window win, GC gc);
void set_button_paddings ();
void draw_all_buttons (GC gc);
int getButtonClick (Button btn, int bx, int by);
init_color ()
    if (!XAllocNamedColor (display, colorMap, "red", &color[0], &color[0]))
        exit (1);
    if (!XAllocNamedColor (display, colorMap, "green", &color[1], &color[1]))
        exit (1);
...
    Button1Mask);
                                printf(" TEST : %d\n", report.xbutton.state & Button1Mask);
                                // apo ton Aristero klik. Duadiko KAI (AND) kai an to
apotelesma einai
    if ((report.xbutton.state & Button1Mask) != 0){
        XSetForeground (display, gc, currentColor.pixel);
        XFillRectangle (display, win, gc, xM, yM, 3, 3);
```

Παραδείγματα grep

`grep "if ([^()]*)" grafika.c`

```
user1@host:~$ grep "if ([^()]*)" grafika.c
    if (colorPalletIsVisible == 1)
    if (b.btnState == PRESSED)
    if (b.btnType == EMPTY)
        if (b.caption[0] != ' ')
    if (bx - colorMoveX >= btn.x && bx - colorMoveX <= btn.x + btn.width)
        if (by >= btn.y && by <= btn.y + btn.height)
            if (report.xexpose.count != 0)
            if (splashScreen)
            if (keysym == 'x' || keysym == 'X')
                if (splashScreen)
                    if (report.type == ButtonPress)
                    if (report.type == ButtonPress)
                    if (report.type == ButtonPress)
                    if (report.type == ButtonPress)
                    if (report.type == ButtonPress)
            if (colorPalletIsVisible == 1)
            if (colorPalletIsVisible == 0)
        if (report.xbutton.button == Button3)
        if (brushMode == 1)
user1@host:~$
```

Η εντολή find

Με την εντολή **find** εντοπίζει κανείς αρχεία ακόμη και εάν δεν γνωρίζει το πλήρες όνομά τους. Συνήθης σύνταξη :

```
$ find <startdir> -name "*.f" -print
```

- **startdir** Κατάλογος αφετηρίας της έρευνας
- **-name "*.f"**
- Η έρευνα γίνεται για όλα τα αρχεία που το όνομά τους τελειώνει σε **.f**
- **-print** Εντέλει την εκτύπωση του πλήρους ονόματος κάθε εντοπιζόμενου αρχείου



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Ταμείο
Περιφερειακής
Ανάπτυξης



Όλα είναι δυνατά
Επικρατικό Πρόγραμμα
"Ψηφιακή Σύγκλιση"



Διασυνδέοντας την Έρευνα και τ



ΕΣΠΑ
2007-2013
Πρόγραμμα για την ανάπτυξη
Ποιότητα ζωής για όλους



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΙΩΑΝΝΙΝΩΝ



Επιστημονικό &
Τεχνολογικό
Πάρκο
Ηπείρου



MEDICAL TECHNOLOGY
INTELLIGENT
INFORMATION SYSTEMS

Παράδειγμα

```
$ find /usr -name "v*.h" -print
```

Η παραπάνω εντολή δημιουργεί μιά λίστα με όλα τα αρχεία που το όνομά τους αρχίζει από **v** και τελειώνει σε **.h** και ευρίσκονται είτε στο **/usr** είτε σε περιεχόμενο υποκατάλογο. Μια πιθανή έξοδος μπορεί να έχει την μορφή:

```
/usr/spool/src/vlimit.h  
/usr/include/sys/var.h  
/usr/home/user9801/van.h
```

find . . .

- Η εντολή **find** έχει πολλές παραμέτρους με τις οποίες μπορεί να συνταχθεί.
- Η πλήρης περιγραφή υπάρχει στο εγχειρίδιο του Linux (Linux **manual**).
- Επίσης η εντολή: **man find** δίνει χρήσιμες πληροφορίες.

Παραδείγματα find

- `find . -name "ask1.c" -print`
 - Βρες ξεκινώντας από τον τρέχοντα κατάλογο ένα αρχείο με το όνομα "ask1.c" και τύπωσε το μονοπάτι προς αυτό
- `find . -name "*.c" -exec cat '{}' \;`
 - Βρες ξεκινώντας από τον τρέχοντα κατάλογο όλα τα αρχεία που τελειώνουν σε ".c" και τύπωσε τα περιεχόμενά τους στο τερματικό
- `find . -name "*.c" -exec grep "if" '{}' \; -print`
 - Βρες ξεκινώντας από τον τρέχοντα κατάλογο όλα τα αρχεία που τελειώνουν σε ".c" και εκτέλεσε σε κάθε ένα από αυτά την εντολή `grep "if"`. Επίσης για κάθε αποτέλεσμα τύπωσε και το μονοπάτι