

Εισαγωγή στο περιβάλλον προγραμματισμού

Λειτουργικά Συστήματα - 6ο εξάμηνο

ακαδημαϊκό έτος 2023 - 2024

Εργαστήριο Υπολογιστικών Συστημάτων



Εργαστήριο

- 1+3 εργαστηριακές ασκήσεις
 - Εκφώνηση
 - Παρουσίαση: περιέχει ό,τι χρειάζεται για τη λύση
 - Εξέταση (Ασκήσεις 1-3): Ανά ομάδα. Εκπρόθεσμη εξέταση: επιβάρυνση 10%
 - Αναφορά: κώδικας, έξοδος προγράμματος, συνοπτικές απαντήσεις στις ερωτήσεις
- Ώρες εργαστηρίου:
 - Τρίτη: 10:45 - 12:15
 - Τρίτη: 12:45 - 14:15
- [Site μαθήματος @ helios](#)
- Mailing list @ site

Βιβλιογραφία

- The C Programming Language (K&R) - Brian Kernighan and Dennis Ritchie
- Advanced Programming in the UNIX® Environment - W. Richard Stevens
- The Linux Programming Interface - Michael Kerrisk
- Linux System Programming: Talking Directly to the Kernel and C Library - Robert Love

Σύνδεσμοι

- [Linux man pages](#)
- [GNU C Library](#)

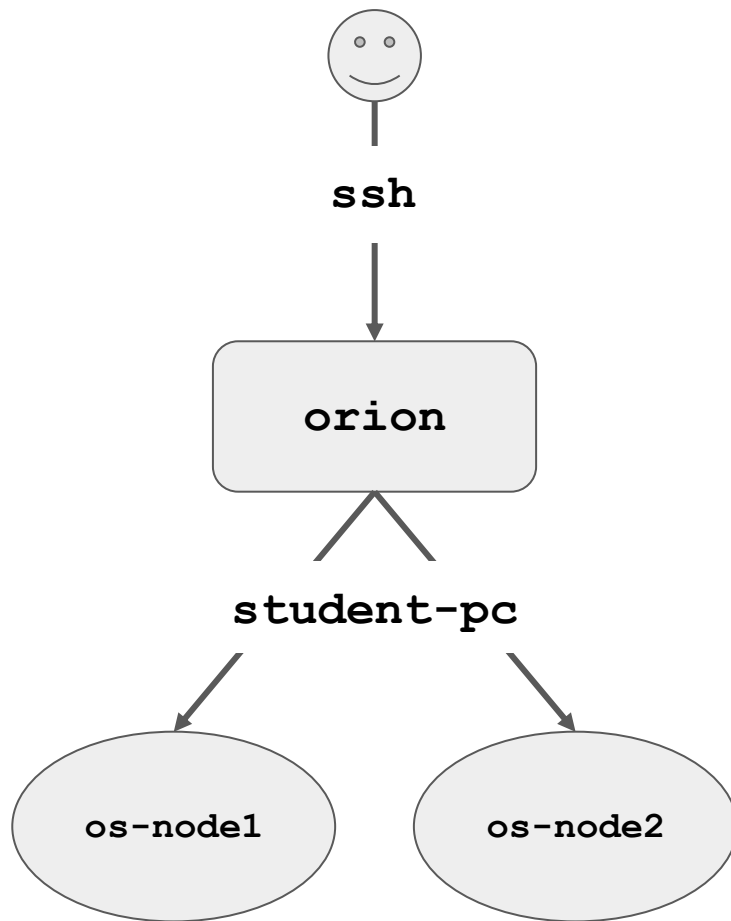
Διαδικαστικά

Λογαριασμοί:

- Username: **oslabXX**
- Αλλαγή password: **yppasswd**

Μηχανήματα:

- **orion.cslab.ece.ntua.gr**
- **os-node1, os-node2**
- Σύνδεση: **ssh** ή **putty**



Σύνδεση στα μηχανήματα

1. Συνδεθείτε στο `orion.cslab.ece.ntua.gr` με `ssh` ή μέσω `putty`
2. Εκτελέστε την εντολή `student-pc` για να συνδεθείτε σε ένα από τα δύο μηχανήματα (`os-node1, 2`) στα οποία θα δουλεύετε (η εντολή επιλέγει τυχαία το ένα από τα δύο).

ΠΡΟΣΟΧΗ:

Το `orion` είναι αποκλειστικά *gateway* μηχάνημα.

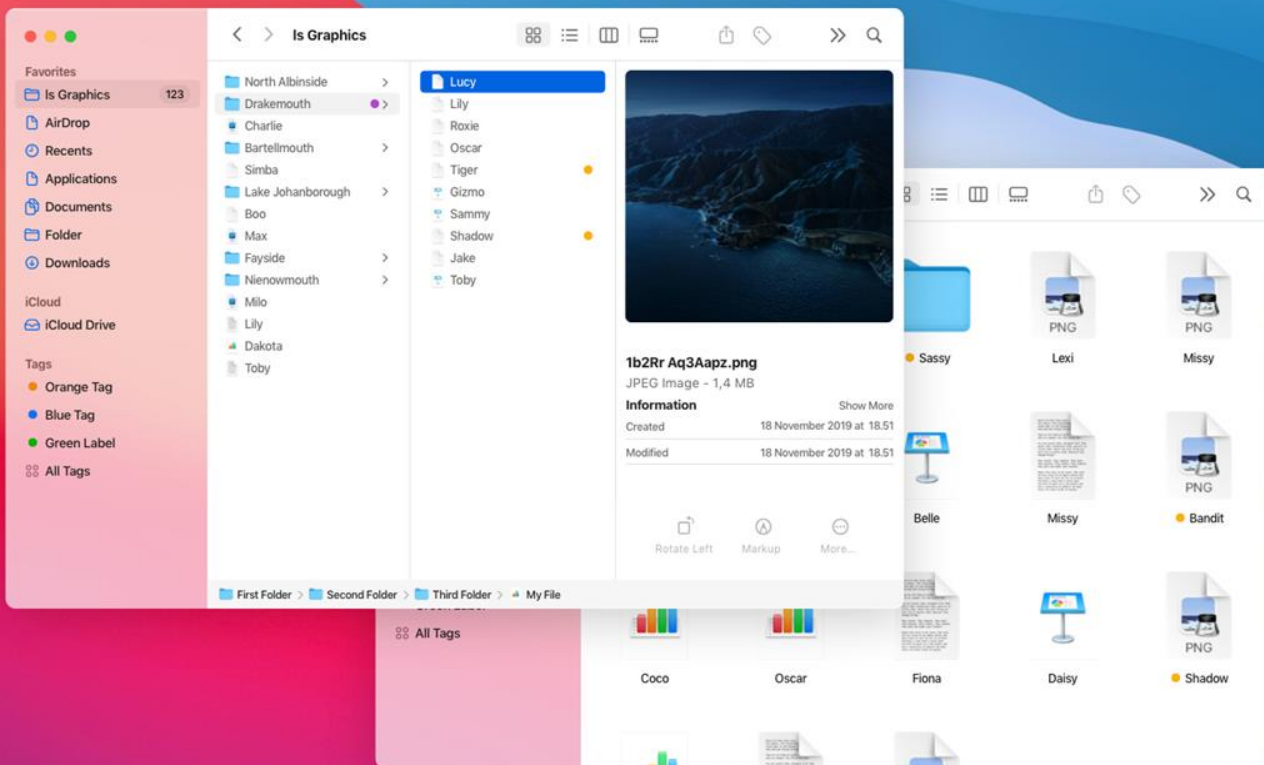
Το χρησιμοποιείτε μόνο για να εκτελέσετε `student-pc`.

ΔΕΝ το χρησιμοποιείτε για να εκτελέσετε τα προγράμματα σας γιατί οι αναγκαίες βιβλιοθήκες είναι εγκατεστημένες μόνο στα `os-nodes`.

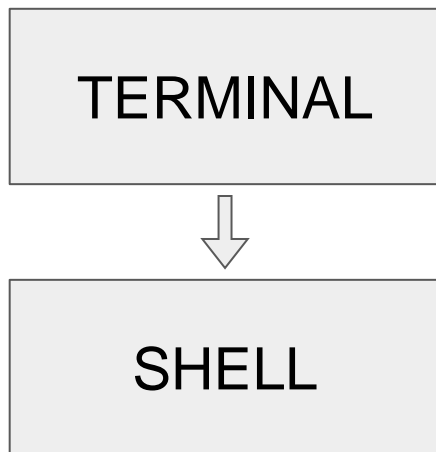
Διεπαφές γραφικών

Τα περισσότερα συστήματα έχουν διεπαφές γραφικών (graphical UIs)

- Δύσκολα προσαρμόσιμες
- Τι κάνουμε όταν δεν είναι διαθέσιμες;



Shell



Για να χρησιμοποιήσουμε ένα SHELL, ανοίγουμε ένα terminal

Μία old-school διεπαφή
βασισμένη σε κείμενο


```
oslabXX@orion:~$
```



Όνομα χρήστη που
χρησιμοποιεί το
terminal

```
oslabXX@orion:~$
```



Όνομα του υπολογιστή στον
οποίο είναι ανοιχτό το terminal

Μπορεί να αλλάξει ενώ έχουμε ανοιχτό ένα terminal;

- Ναι, αν συνδεθούμε με ssh σε άλλο μηχάνημα

```
oslabXX@orion:~$
```

Το μονοπατι του τρέχοντος καταλόγου (current/working directory).

~ : Συντομογραφία για τον προσωπικό κατάλογο (home directory)

home directory: default working directory όταν ανοίγουμε ένα terminal (συνήθως)

Παράδειγμα: Αν home directory: /home/users/oslabXX

και το current directory είναι: /home/users/oslabXX/dir1

Θα βλέπαμε:

```
oslabXX@orion:~/dir1 $
```

oslabXX@orion:~\$



Prompt

Συνήθως:

- \$: απλός χρήστης
- # : διαχειριστής (χρήστης root)

Χρήση του shell

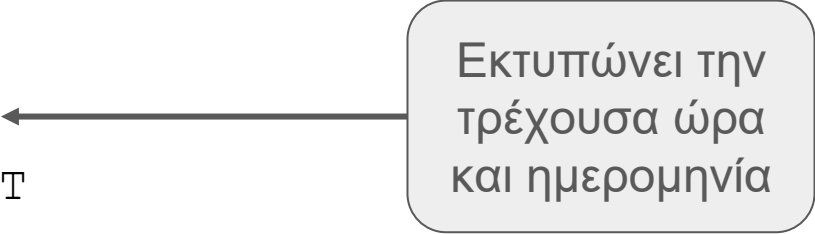
```
oslabXX@orion:~$
```

Χρήση του shell

```
oslabXX@orion:~$ date
```

```
Wed 7 Feb 2024 11:49:31 AM EST
```

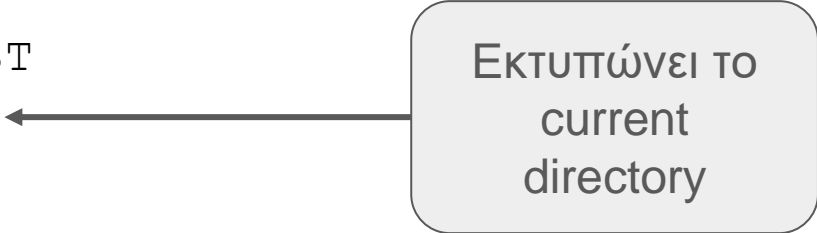
```
oslabXX@orion:~$
```



Εκτυπώνει την
τρέχουσα ώρα
και ημερομηνία

Χρήση του shell

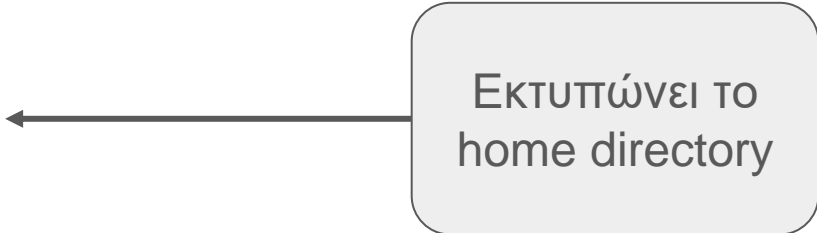
```
oslabXX@orion:~$ date  
Wed 7 Feb 2024 11:49:31 AM EST  
oslabXX@orion:~$ pwd  
/home/users/oslabXX  
oslabXX@orion:~$
```



ΕΚΤΥΠΩΝΕΙ ΤΟ
current
directory

Χρήση του shell

```
oslabXX@orion:~$ date  
Wed 7 Feb 2024 11:49:31 AM EST  
oslabXX@orion:~$ pwd  
/home/users/oslabXX  
oslabXX@orion:~$ echo $HOME  
/home/users/oslabXX  
oslabXX@orion:~$
```



ΕΚΤΥΠΩΝΕΙ ΤΟ
home directory

Χρήση του shell

```
oslabXX@orion:~$ date
```

```
Wed 7 Feb 2024 11:49:31 AM EST
```

```
oslabXX@orion:~$ pwd
```

```
/home/users/oslabXX
```

```
oslabXX@orion:~$ echo $HOME
```

```
/home/users/oslabXX
```

```
oslabXX@orion:~$ export A="hello-world"
```

```
oslabXX@orion:~$
```



Θέτει τη
μεταβλητή A ίση
με "hello-world"

Χρήση του shell

```
oslabXX@orion:~$ date
```

```
Wed 7 Feb 2024 11:49:31 AM EST
```

```
oslabXX@orion:~$ pwd
```

```
/home/users/oslabXX
```

```
oslabXX@orion:~$ echo $HOME
```

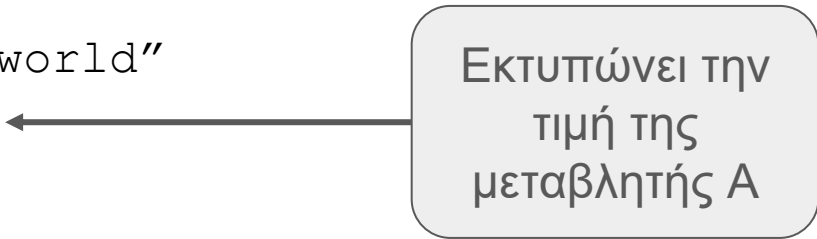
```
/home/users/oslabXX
```

```
oslabXX@orion:~$ export A="hello-world"
```

```
oslabXX@orion:~$ echo $A
```

```
hello-world
```

```
oslabXX@orion:~$
```



Εκτυπώνει την
τιμή της
μεταβλητής A

Μεταβλητές περιβάλλοντος

- Ορίζονται με `export`
- Τυπώνονται με `echo`
- Προκαθορισμένες μεταβλητές:
 - `$HOME`: home directory του χρήστη
 - `$PATH`: Λίστα directories που περιέχουν εκτελέσιμα αρχεία.
Για προγράμματα εκτός του `$PATH`, απαιτείται μονοπάτι:
`./program`
- [man env](#)

Σύστημα αρχείων του Linux

Συστήματα αρχείων: ιεραρχικές δομές

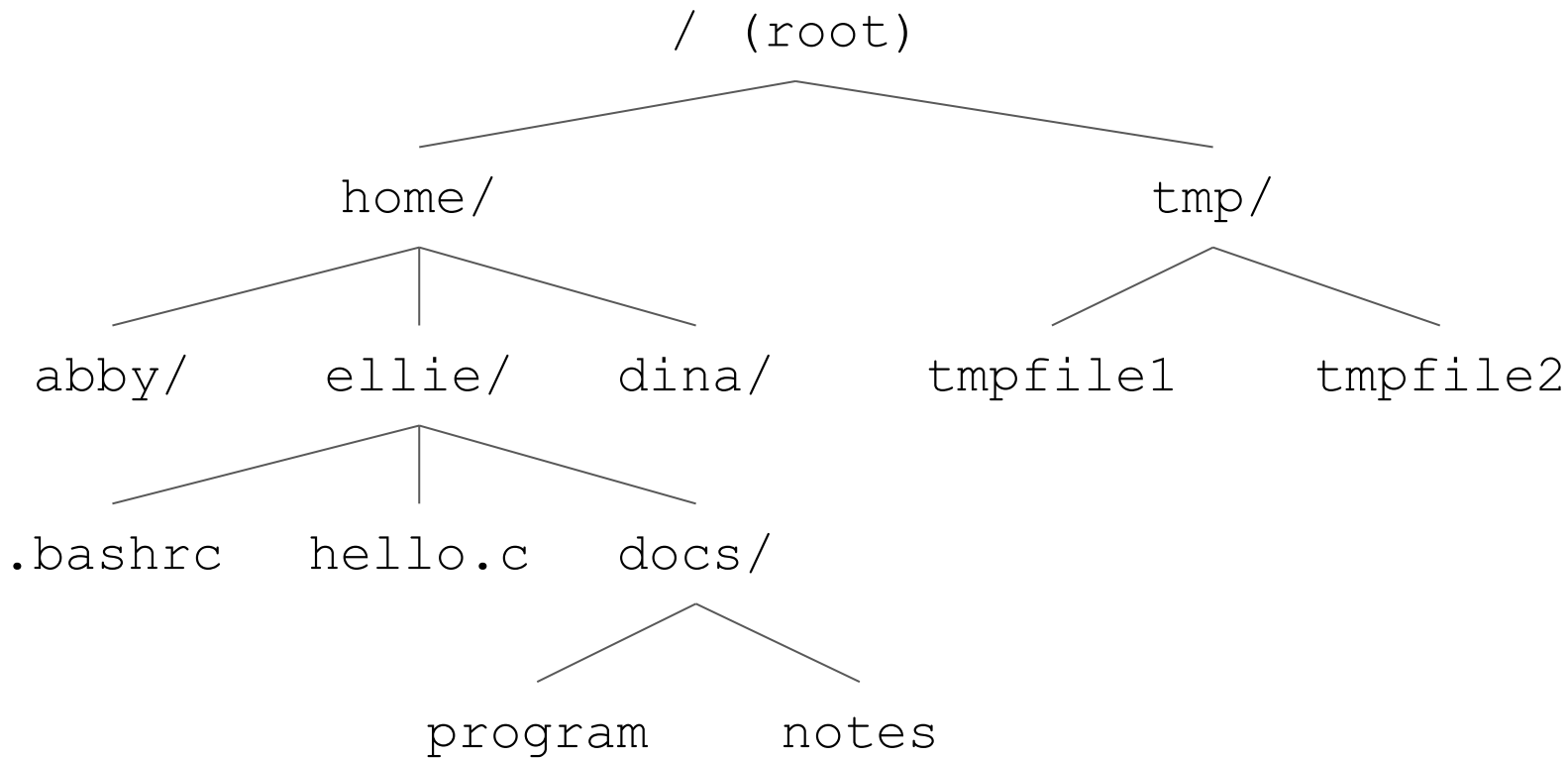
- **Windows:** `C:\Users\user\Documents\folder\file.pdf`
- **Linux:** `/home/Users/user/Documents/folder/file.pdf`

Μονοπάτι (path): Συμβολοσειρά από αναγνωριστικά χωρισμένα με τον χαρακτήρα /

Ένα μονοπάτι είναι:

- **απόλυτο:** ξεκινάει με /, αφετηρία είναι η αρχή της ιεραρχίας
- **σχετικό:** δεν ξεκινάει με /, αφετηρία το current directory

Σύστημα αρχείων του Linux



Πλοήγηση στο σύστημα αρχείων

Αναγνωριστικά:

- `.` : current directory
- `..` : parent directory

Εντολές:

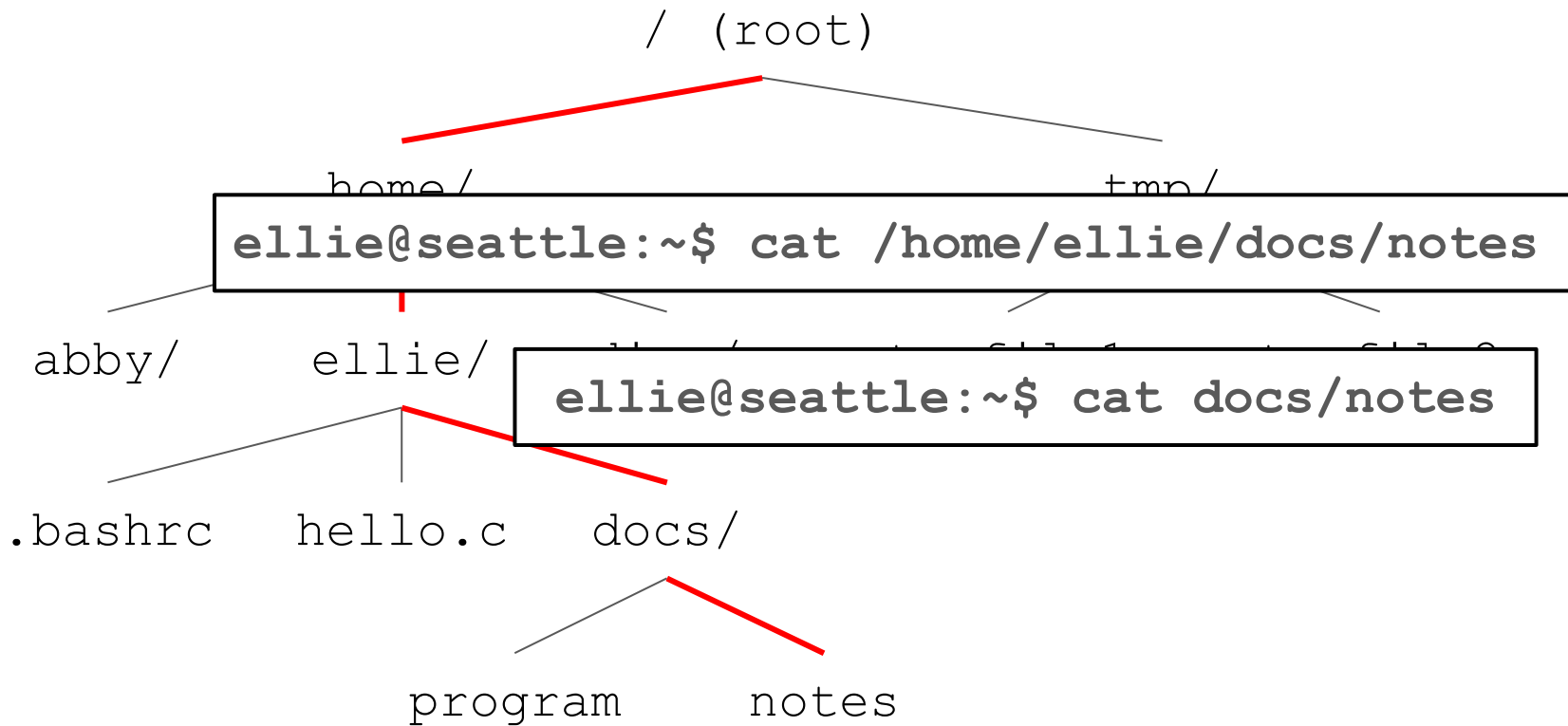
- **ls**: εμφάνιση περιεχομένων του directory

```
ellie@seattle:~$ ls  
hello.c docs/
```

- **cd**: αλλαγή directory

```
ellie@seattle:~$ cd docs  
ellie@seattle:~/docs$
```

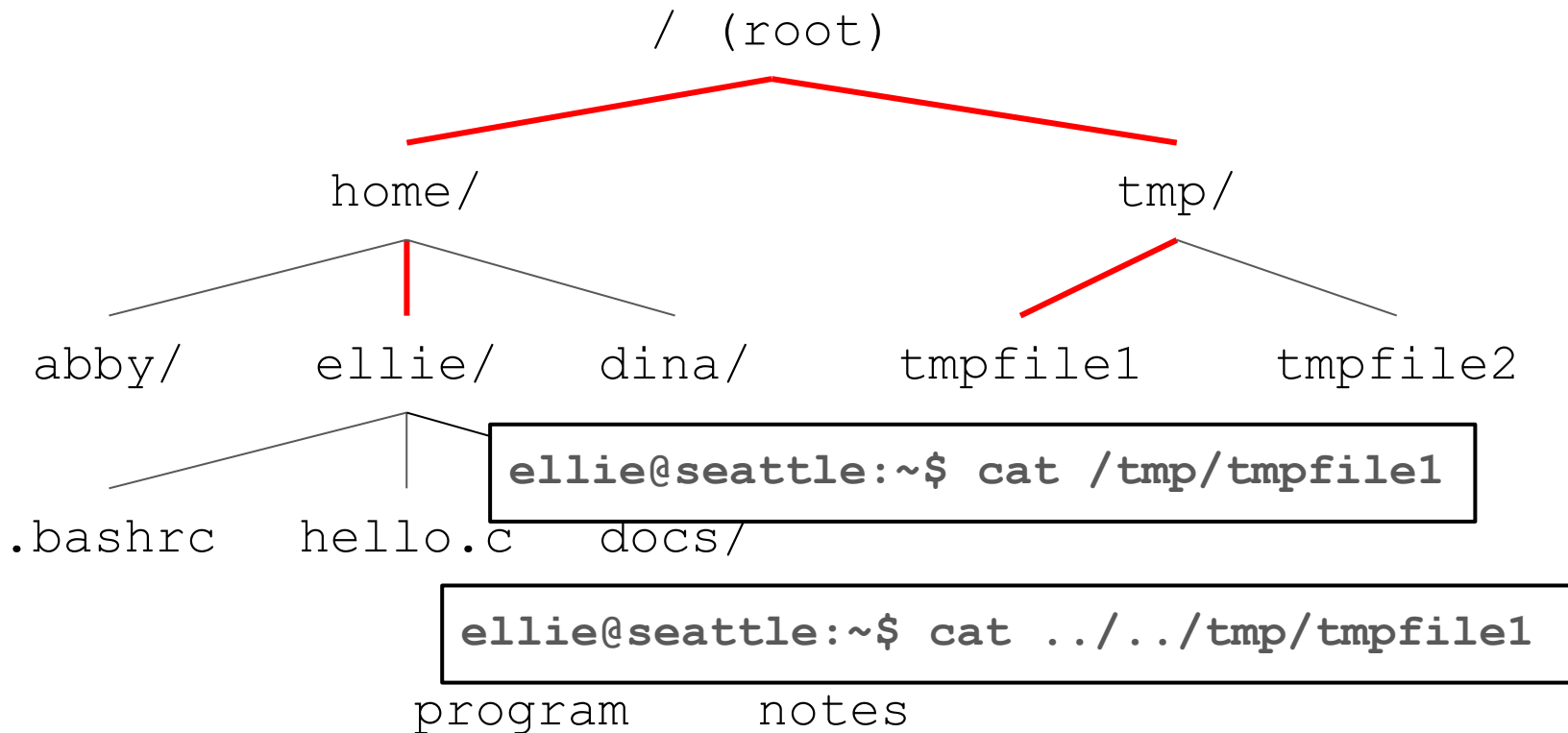
Πλοήγηση προς το αρχείο notes



Εκτέλεση του program



Πλοήγηση προς το tmpfile1



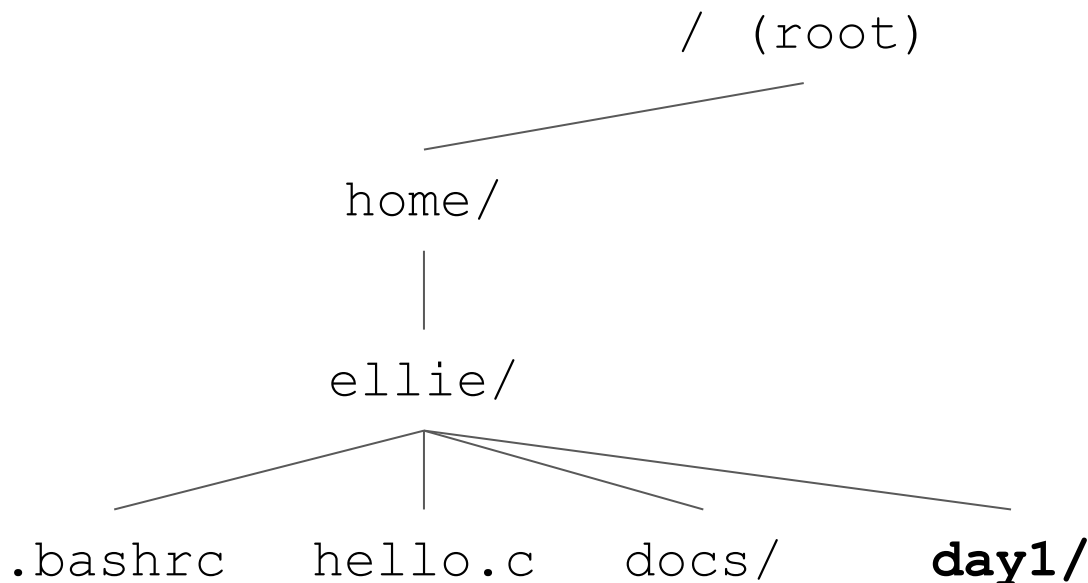
Χρήσιμες εντολές για αρχεία και καταλόγους

```
ellie@seattle:~$ mkdir day1
```

```
ellie@seattle:~$ ls
```

```
hello.c docs/ day1/
```

```
ellie@seattle:~$
```



Χρήσιμες εντολές για αρχεία και καταλόγους

```
ellie@seattle:~$ mkdir day1
```

```
ellie@seattle:~$ ls
```

```
hello.c docs/ day1/
```

```
ellie@seattle:~$ rmdir day1
```

```
ellie@seattle:~$
```

Χρήσιμες εντολές για αρχεία και καταλόγους

```
ellie@seattle:~$ mkdir day1
```

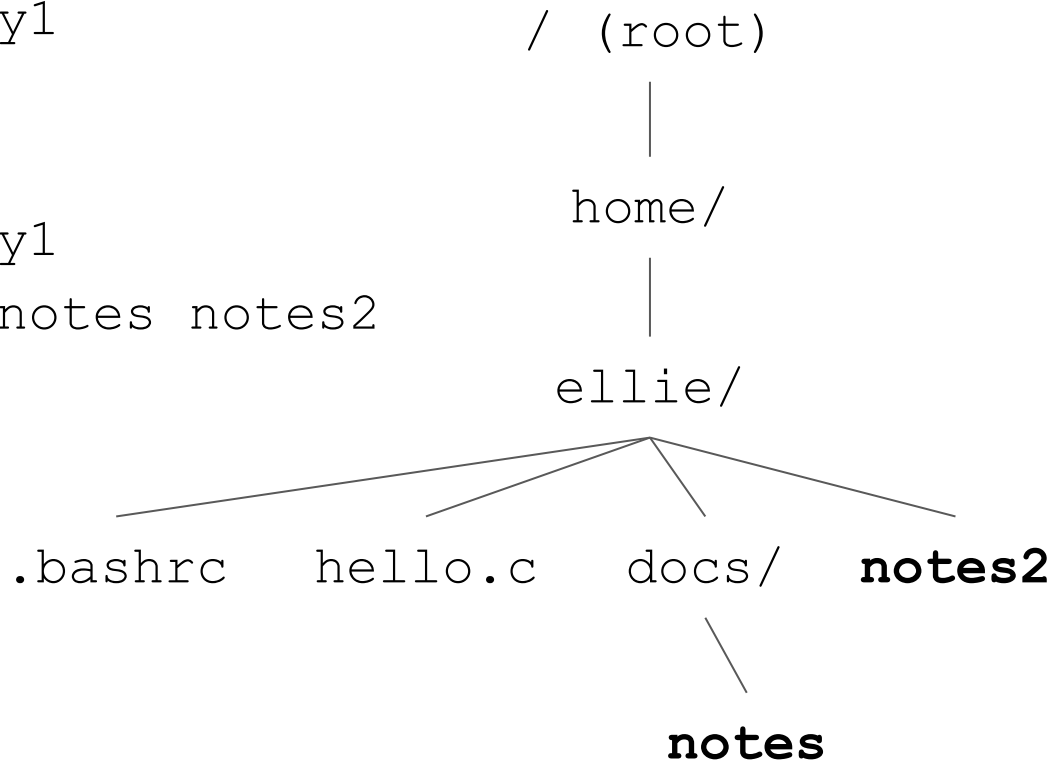
```
ellie@seattle:~$ ls
```

```
hello.c docs/ day1
```

```
ellie@seattle:~$ rmdir day1
```

```
ellie@seattle:~$ cp docs/notes notes2
```

```
ellie@seattle:~$
```



Χρήσιμες εντολές για αρχεία και καταλόγους

```
ellie@seattle:~$ mkdir day1
```

```
ellie@seattle:~$ ls
```

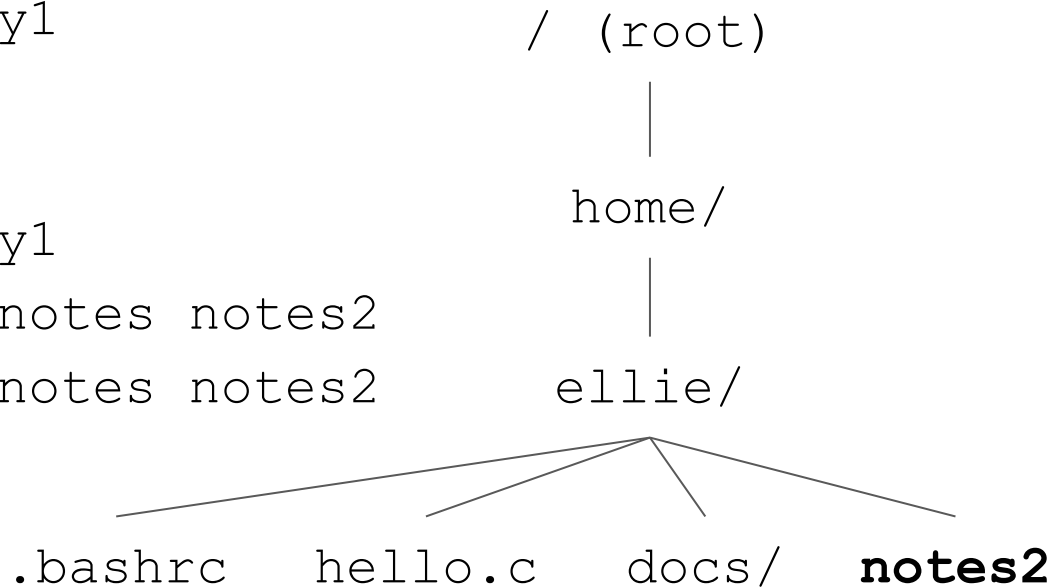
```
hello.c docs/ day1
```

```
ellie@seattle:~$ rmdir day1
```

```
ellie@seattle:~$ cp docs/notes notes2
```

```
ellie@seattle:~$ mv docs/notes notes2
```

```
ellie@seattle:~$
```



Χρήσιμες εντολές για αρχεία και καταλόγους

```
ellie@seattle:~$ mkdir day1
```

```
ellie@seattle:~$ ls
```

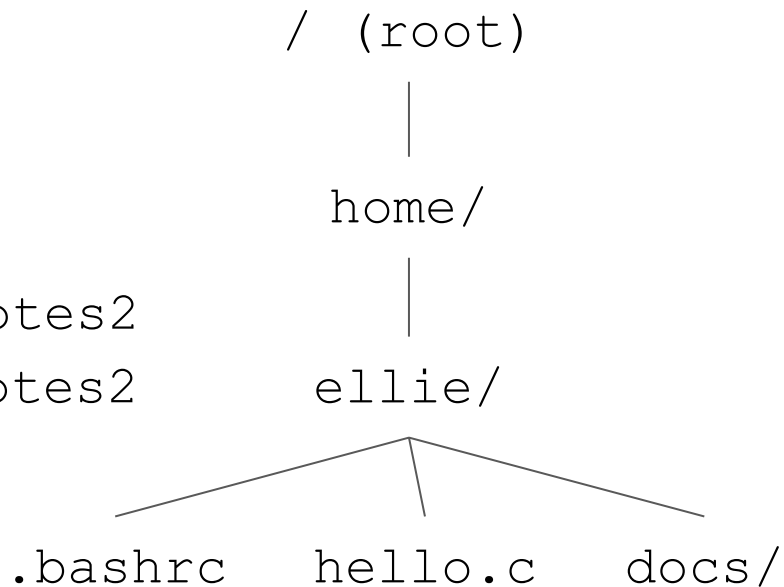
```
hello.c docs/ day1
```

```
ellie@seattle:~$ rmdir day1
```

```
ellie@seattle:~$ cp docs/notes notes2
```

```
ellie@seattle:~$ mv docs/notes notes2
```

```
ellie@seattle:~$ rm notes2
```



Ιστορικό του shell

Με τα πλήκτρα ↑ και ↓ πλοηγούμαστε στις προηγούμενες εντολές που χρησιμοποιήσαμε.

Με το TAB κάνουμε autocomplete:

```
ellie@seattle:~$ cd d[TAB]ocs
```

```
ellie@seattle:~/docs$
```

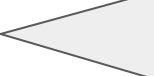
Τεκμηρίωση

Η εντολή `man` δείχνει το `man page` μιας εντολής:

```
ellie@seattle:~$ man ls
```

```
ellie@seattle:~$ man 2 read
```

```
ellie@seattle:~$ man -k printf
```



```
LS(1)                                User Commands                                LS(1)
NAME
  ls - list directory contents
SYNOPSIS
  ls [OPTION]... [FILE]...
DESCRIPTION
  List information about the FILES (the current directory by default).
  Sort entries alphabetically if none of -cftuvSUX nor --sort is speci-
  fied.
  Mandatory arguments to long options are mandatory for short options
  too.
  -a, --all
      do not ignore entries starting with .
  -A, --almost-all
      do not list implied . and ..
  --author
```


Σύνδεση σε απομακρυσμένο μηχάνημα

Εντολή: `ssh`

```
student@pc:~$ ssh oslabXX@orion.cslab.ece.ntua.gr
```

```
password:
```

```
oslabXX@orion:~$ student-pc
```

```
...
```

```
oslabXX@os-node1:~$
```

Editors

- Δημοφιλείς editors: VS Code, Atom, Geany, ...
- Το γραφικό περιβάλλον δεν είναι πάντα διαθέσιμο
- Λύση: Μαθαίνω VIM
- vim: modal editor:
 - normal
 - insert
 - replace
 - visual
 - command-line

VIM - Κίνηση μεταξύ modes

- Insert: **i**
- Replace: **R** (**r** για αντικατάσταση ενός χαρακτήρα)
- Visual:
 - **v** : κανονική επιλογή
 - **V** : επιλογή ολόκληρων γραμμών
 - **Ctrl+v** : box-like επιλογή
- Command Line: **:**

Tutorials: [vim Tutorial for beginners](#)
[Learn vim the simple way](#)

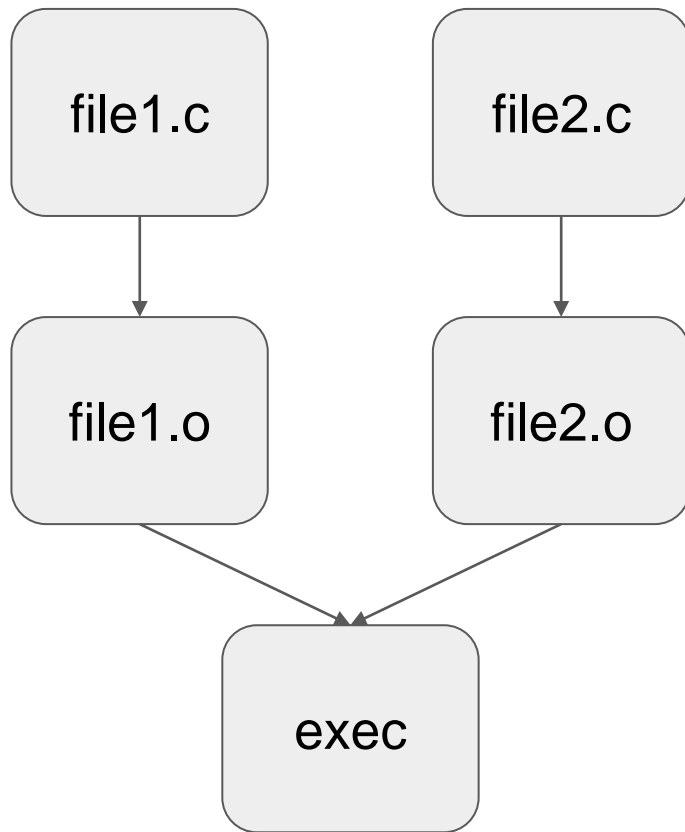
Παραγωγή εκτελέσιμου

Compile (μεταγλώττιση):

$\text{file1.c} \rightarrow \text{file1.s} \rightarrow \text{file1.o}$

Link:

$\text{file1.o} + \text{file2.o} \rightarrow \text{exec}$



Παράδειγμα: Hello world

hello.c

```
#include <stdio.h>
int main(int argc, char **argv)
{
    printf("Hello World!\n");
    return 0;
}
```

Γραμμή εντολών

```
$ gcc -Wall hello.c -o hello
$ ./hello
Hello World
```

Παράδειγμα: Hello world

Γραμμή εντολών

```
$ gcc -Wall -c hello.c  
$ gcc hello.o -o hello  
$ ./hello  
Hello World!
```

Πολλαπλά αρχεία

hello.c

```
#include <stdio.h>
void hello(const char *name)
{
    printf("Hello %s!\n", name);
}
```

main.c

```
void hello(const char *);

int main(int argc, char **argv)
{
    hello("World");
    return 0;
}
```

Γραμμή εντολών

```
$ gcc -Wall -c main.c
$ gcc -Wall -c hello.c
$ gcc main.o hello.o -o hello
$ ./hello
Hello World!
```

Επικεφαλίδες

- Διεπαφή προς άλλα κομμάτια κώδικα (API)
- Περιέχουν πρότυπα και δηλώσεις
 - Συναρτήσεις
 - Καθολικές (global) μεταβλητές
- .h αρχεία
- preprocessor:
`#include "header.h"`

Επικεφαλίδες: Παράδειγμα

```
void hello(const char *);
```

hello.h

```
#include <stdio.h>
void hello(const char *name)
{
    printf("Hello %s!\n", name);
}
```

hello.c

```
#include "hello.h"
int main(int argc, char **argv)
{
    hello("World");
    return 0;
}
```

main.c

Makefiles

- Εφαρμογή: make
- Χρήση για την αυτόματη δημιουργία προγραμμάτων από αρχεία κώδικα
- Διαδικασία:
 - Συγγραφή αρχείου Makefile που περιέχει *κανόνες*
 - Κλήση εντολής make για παραγωγή του προγράμματος
- Κανόνες:
 - Αρχείο-στόχος (target)
 - Αρχεία-απαιτήσεις (prerequisites)
 - Εντολή παραγωγής στόχου από τα απαιτούμενα αρχεία

Makefiles

Παράδειγμα:

```
target : prerequisites  
    command
```

Tutorials: [Makefile Tutorial](#)

[Unix Makefile Tutorial](#)

Makefiles: Παράδειγμα

Makefile

```
hello: hello.o main.o
gcc -o hello hello.o main.o
hello.o: hello.c
gcc -Wall -c hello.c
main.o: main.c
gcc -Wall -c main.c
```

Makefiles: Παράδειγμα

Γραμμή εντολών

```
$ make
gcc -Wall -c hello.c
gcc -Wall -c main.c
gcc -o hello hello.o main.o
$ make
make: `hello' is up to date.
$ edit main.c
$ make
gcc -Wall -c main.c
gcc -o hello hello.o main.o
```

Ορίσματα προγράμματος

```
int main(int argc, char **argv)
```

`argc` Αριθμός ορισμάτων

`argv` Πίνακας με τα ορίσματα

`argv[0]` Το όνομα του προγράμματος

Βοηθητική βιβλιοθήκη: `getopt(3)`

Ορίσματα προγράμματος

args.c

```
#include <stdio.h>
int main(int argc, char **argv)
{
    int i;
    for (i=0; i<argc; i++)
        printf("%d %s\n", i, argv[i]);
    return 0;
}
```

Γραμμή εντολών

```
$ ./args arg1 arg2
0 ./args
1 arg1
2 arg2
```