

Εργαστήριο

Εισαγωγή στον Προγραμματισμό

2^ο Εργαστήριο

Βασικές Έννοιες

Βασιλόπουλος Διονύσης

Ε.ΔΙ.Π. Τμήματος Πληροφορικής & Τηλεπικοινωνιών

Αντικείμενο 2^{ου} Εργαστηρίου

Hello Linux

- linux directory
- C stuff
- linux commands
- git

Linux

- Directory structure (/home, .., .) (cd /home (τι δείχνει?))

1. ls

ls (-a, -l, *, ?)

Βρείτε τα αρχεία που έχουν τουλάχιστον 5 γράμματα, το 3^ο είναι l, και έχουν κατάληξη .c

ls *i?* (πόσα γράμματα τουλάχιστον έχει το file?)

ls public_html (δείχνει τα αρχεία του καταλόγου public_html)

ls -d */ (εμφανίζει μόνο τους καταλόγους)

2. pwd

Δείχνει το πλήρες Path στο οποίο βρίσκεστε

3. mkdir (make directory)

mkdir myWork

Linux

4. cd (change directory)

cd myWork

cd public_html/thesis

cd (επιστροφή στο αρχικό μας directory)

cd \$HOME

5. cp (copy)

cp file1 file2 (cp hello.c hallo.c)

cp file directory_structure

cp -l file1 file (cp -i hello.c hallo.c : θα γίνει ερώτηση για αντικατάσταση)

cp -R dir1 dir2 (αντιγράφει τον κατάλογο dir1 στον dir2 (files+dirs.)

Αντιγράψτε στον κατάλογο myWork το αρχείο ~iphw/samples/mysin.c

C

- Comments
 - `/*` Comment
one or
more lines `*/`
 - `//` Comment a single line
- Include statement
 - `#include <filename>` (συμπεριλαμβάνει κάποιο file στον κώδικα)
Συνήθως (όπως στο `stdio.h` κάποιο header file)

<https://www.geeksforgeeks.org/c-c-include-directive-with-examples/>
<https://www.geeksforgeeks.org/header-files-in-c-cpp-and-its-uses/>

6.gcc (GNU compiler collection, or GNU C compiler)

```
gcc -o executable_file c_file.c
```

```
gcc -o executable_file c_file.c -lm (όταν θέλουμε και μαθηματικές συναρτήσεις, το  
-lm στο τέλος της εντολής)
```

Αν δεν είστε στον κατάλογο myWork να πάτε σε αυτόν με την εντολή cd

Δημιουργήστε το εκτελέσιμο αρχείο mysin

7.Τρέξτε το εκτελέσιμο mysin

C

6.gcc (GNU compiler collection, or GNU C compiler)

```
gcc -o mysin mysin.c -lm
```

7.

./mysin (ίσως να δουλεύει και απλά το mysin)

8.

```
cp ~iphw/samples/mymain.c mymain.c
```

9.

```
cp ~iphw/samples/myfunct.o myfunct.o
```

10.

`gcc -c mymain.c` (παράγεται το `mymain.o`)

11.

`gcc -o myprog mymain.o myfunct.o`

```
//FILE: mymain.c  
void myfunct();
```

```
int main()  
{  
    myfunct();  
}
```

12.

`./myprog`

Linux

13.

`rm file`

`rm -r directory`

`rm -i file` για επιβεβαίωση διαγραφής (ορισμένες φορές είναι η default συμπεριφορά)

`rmdir directory` (ο κατάλογος σβήνεται μόνο εάν είναι κενός)

14.

`cp /usr/include/stdio.h $HOME`

15.

`mv file1 file2 | mv file1 directory/file2`

`mv stdio.h Mystdio.h`

16.

`cat file` (εμφανίζει τα περιεχόμενα ενός αρχείου)

`cat -n file` (εμφανίζει τα περιεχόμενα ενός αρχείου με αρίθμηση γραμμών)

`cat Mystdio.h -n`

17.

`touch file` (δημιουργούμε ένα κενό αρχείο)

`touch my_file`

`touch .my_file` (η τελεία **.** δημιουργεί ένα **κρυφό** αρχείο)

18.

`ls my_file`

or

`ls .my_file -la`

or

`ls *my_file -la` (δεν εμφανίζει και τα δύο, το ***** δεν αφορά τα κρυφά αρχεία)

19.chmod (change mode)

- chmod (r=read=4, w=write=2, x=execute=1, ugo/a)
- u(**u**ser), g(user **g**roup), o(**o**thers), a (**a**ll)
- chmod a+r file
- chmod 666 file
- chmod 777 file

chmod 754 .my_file

Ποια η διαφορά μεταξύ chmod g+w (a) και chmod g=w (b) ?

20.

ls .my_file

21.

ls -l

19.

- Ποια η διαφορά μεταξύ chmod g+w (a) και chmod g=w (b) ?
 - Στο a **προστίθεται** δικαίωμα w στο group του χρήστη ενώ στο b το δικαίωμα της ομάδας είναι **μόνο** το w
- Για να δώσουμε δικαιώματα σε συγκεκριμένο user:
setfacl -m u:username:rwx userfolder/userfile (συνήθως το αποφεύγουμε) ή
ως admin βάζουμε σε ένα group μόνο του τον user και δίνουμε δικαιώματα στο group
<https://www.geeksforgeeks.org/linux-setfacl-command-with-example/>

22.

ls -la

23. man command name

man ls (πατάτε ENTER για επόμενη γραμμή, SPACE για επόμενη σελίδα)

ls --help

24. grep search_text file(s)_to_search

grep size /usr/include/stdlib.h

grep size /usr/include/stdlib.h **-n** (εμφανίζει και τις γραμμές)

25. command >export_file (ανακατεύθυνση εξόδου)

grep size /usr/include/stdlib.h **>grepout.txt**

cat grepout.txt (to view file content)

26. Σωλήνωση (pipe/pipeline): Μεταφορά των αποτελεσμάτων μιας εντολής όχι σε κάποια έξοδο (π.χ. οθόνη, αρχείο), αλλά σε είσοδο άλλης εντολής.

```
command | command
```

```
touch pipeline.txt
```

```
chmod 600 pipeline.txt
```

```
ls -l | grep rw----- ή ls -l | grep "rw-----" ή ls -l | grep 'rw-----'
```

```
ls -l | grep -- -rw-----
```

```
ls -l | grep "\-rw-----"
```

```
ls -l | grep "^-rw-----"
```

(Δεν δείχνουν τους καταλόγους)

27. Ανακατεύθυνση εισόδου εξόδου

```
command <file_input >file_output
```

```
cp ~/iphw/samples/capitalize.c capitalize.c
```

```
gcc -o capitalize capitalize.c
```

```
capitalize <capitalize.c >CAPITALIZE.c
```

28. Γιατί ορισμένα keywords είναι κεφαλαία

(<https://www.scaler.com/topics/is-c-language-case-sensitive/>)

+το όνομα των αρχείων στο linux είναι case sensitive (δεν βρίσκει π.χ. το STDIO.Η αφού υπάρχει μόνο το stdio.h)

Περίληψη

- Χρήση linux εντολών
- Git
- Διαβάζετε διαφάνειες 1^{ου} και 2^{ου} μαθήματος θεωρίας.