

Introduction to Bioinformatics

Alexandros C. Dimopoulos
alexdem@di.uoa.gr

Master of Science
“Data Science and Information Technologies”
Department of Informatics and Telecommunications
National and Kapodistrian University of Athens

2024-25



about me

Alexandros C. Dimopoulos, Ph.D.

- BSc Electrical and Computer Engineer, NTUA (2004)
- Ph.D. in Computer Science, NTUA (2009)
- Adjunct Lecturer, Harokopio University (2010-2020)
- Post-Doc Researcher, BSRC Al. Fleming (2012 -)
- M.Sc. “Data Science and Information Technologies” DIT, UoA (2017 -)
- Lecturer, Hellenic Naval Academy (2020 -)



Course Overview

- 1 Thursday, 17 October 2024: Introduction to GNU/Linux and to basic commands
- 2 Thursday, 24 October 2024: Introduction to the R programming language and to RStudio utilization
- 3 Thursday, 31 October 2024: More advanced programming in R and introduction to Bioconductor
- 4 Thursday, 14 November 2024: Usage of CLI tools such as bedtools, vcftools, samtools etc.
- 5 Thursday, 28 November 2024: SNP calling Pipelines



GNU/Linux

GNU/Linux

Linux is a Unix-like computer operating system assembled under the model of free and open-source software development and distribution. The defining component of Linux is the Linux kernel, an operating system kernel first released on September 17, 1991 by Linus Torvalds. The Free Software Foundation uses the name GNU/Linux to describe the operating system, which has led to some controversy.



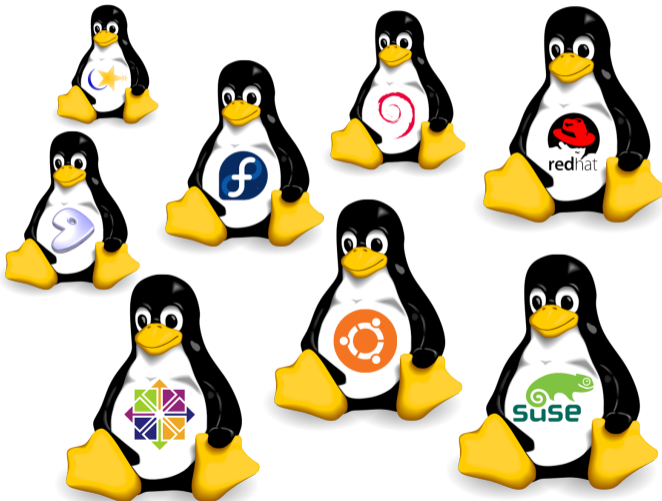
Linus Torvalds & Richard Stallman



Linus Torvalds and Richard Stallman



GNU/Linux distributions (distros)



Shell

- The oldest way of communicating with the computer
- Not always very (user) friendly

```
paste <(cat out_23Genes.txt | cut -f16-18 | awk '{
    print "chr"$1"\t"$2-1"\t"$2 }') <(cat out_23Genes.txt
) >out_23Genes.new.bed
```

- Very useful for combining existing commands/tools and redirection (pipes)
- Various different shells: Bash, Tcsh/Csh, Ksh, Zsh, Fish, ...

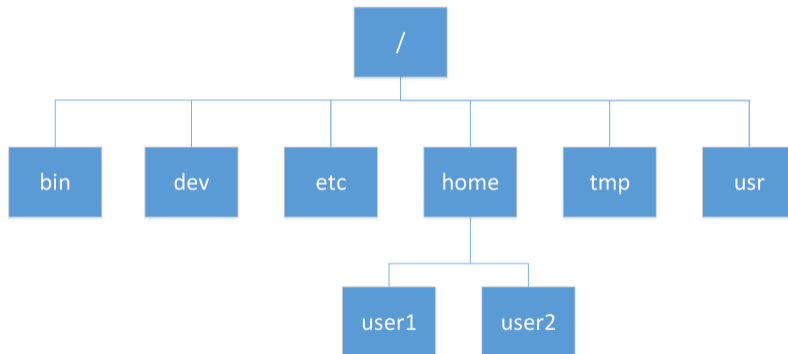


Bash (Bourne-again shell)

- `$` <user privileges>
- `#` <root privileges>
- auto-completion while typing by pressing the Tab key
- program execution in the foreground
- program execution in the background (&)



File structure I



File structure II

```
ls -l /
total 120
drwxr-xr-x  4 root root 12288 Jul 24 11:30 bin
drwxr-xr-x  3 root root  4096 Aug  7 08:14 boot
drwxr-xr-x 20 root root  3500 Sep 13 11:41 dev
drwxr-xr-x 220 root root 16384 Sep 12 19:43 etc
lrwxrwxrwx  1 root root    15 Apr  4 14:42 home
.....
```

- EVERYTHING is a file (files, directories, hard-drives, modems, keyboards, printers)
- home folder (~)
 - unlimited access/rights from the user
 - cd ~ or just cd



Change directory (folder)

```
$ cd /tmp
$ pwd
/tmp
$ cd ~
$ pwd
/home/alexdem
$ cd /tmp
$ cd
$ pwd
/home/alexdem
```



List directory contents

```
$ ls /etc/dhcp/
debug  dhclient.conf  dhclient-enter-hooks.d  dhclient-exit-hooks.d
$ ls /etc/dhcp/ -l # number 1
debug
dhclient.conf
dhclient-enter-hooks.d
dhclient-exit-hooks.d
$ ls /etc/dhcp/ -l # smallcase L
total 16
-rw-r--r-- 1 root root 1426 Nov 26  2016 debug
-rw-r--r-- 1 root root 1735 Nov 26  2016 dhclient.conf
drwxr-xr-x 2 root root 4096 Jul 18  11:33 dhclient-enter-hooks.d
drwxr-xr-x 2 root root 4096 Jul 24  11:30 dhclient-exit-hooks.d
```



Help

```
$ man ls
```

```
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  
specified.
```

```
Mandatory arguments to long options are mandatory for short options  
too.
```

```
-a, --all
```

```
do not ignore entries starting with .
```

- [google](#)



Total and relevant Paths

- . → current directory
- .. → one level back

```
$ pwd
/tmp/directory1/directory2
$ cd .
$ pwd
/tmp/directory1/directory2
$ cd ..
$ pwd
/tmp/directory1/
$ cd /tmp/directory3
$ pwd
/tmp/directory3
$ cd ../directory2
$ pwd
/tmp/directory2
```



Creating and modifying directories

- create directory

```
mkdir dirName
```

- rename directory

```
mv oldDirName newDirName
```

- move directory

```
mv oldDirName /tmp/newDirName
```

- remove (delete) directory

```
rmdir dirName (if it is empty)
```

```
rm -r dirName (even if not empty; any files are first deleted and then removed)
```



File permissions I

```
drwxr-xr-x 220 root root 16384 Sep 12 19:43 etc
```

3 access categories:

- 1 user : refers only to the user that owns the file
- 2 group : refers to all the users that belong to the specific group
- 3 other : refers to all the system users

Permission	Meaning for directory	Meaning for file
r	List the directory	Read contents
w	Create or remove files	Write contents
x	Access files and subdirectories	Execute



File permissions II

Value	Meaning
0	---
1	--x
2	-w-
3	-wx
4	r--
5	r-x
6	rw-
7	rwX

```
chmod 740 fname
```



Program execution

- `./a.out` (if the executable is in the current directory)
- `/<PATH_TO_FILE>/a.out`



Useful commands & programs I

cp - copy

```
cp source destination
```

mv - move

```
mv source destination
```

cat - concatenate files and print on the standard output

```
cat text_file
```

echo - display a line of text

```
$echo "hello world"  
hello world
```



Useful commands & programs II

head - output the first part of files

```
head text_file
```

```
head -n 30 text_file (first 30 lines)
```

tail - output the last part of files

```
tail text_file
```

```
tail -n 30 text_file (last 30 lines)
```



Useful commands & programs III

```
$cat n.txt
```

```
101  
102  
103  
104  
105  
106  
...  
199  
200
```

```
$head -n 5 n.txt
```

```
101  
102  
103  
104  
105
```

```
$tail -n 3 n.txt  
198  
199  
200
```



Useful commands & programs IV

more - file perusal filter for crt viewing

```
more text_file  
cat text_file | more
```

less - opposite of more

```
less text_file  
cat text_file | less
```



Redirection

- `>` : redirecting output (stdout) into a file - create/overwrite a new/existing file
e.g. `ls > /tmp/out.txt`
- `>>` : redirecting output (stdout) into a file - append to an existing file
e.g. `ls >> /tmp/out.txt`
- `2>` : redirecting standard error (stderr) into a file
e.g. `ls 2> /tmp/out_error.txt`
- `&>` : redirecting both stdout and stderr into a file
e.g. `ls &> /tmp/out_stdout_error.txt`
- `|` : redirecting (stdout) to be used as input by another command
e.g. `cat a.txt | less`



Additional useful command & programs I

grep - print lines matching a pattern

```
grep pattern text_file  
cat text_file | grep pattern
```

```
$cat n.txt
```

```
1
```

```
3
```

```
5
```

```
7
```

```
9
```

```
11
```

```
13
```

```
$grep 1 n.txt
```

```
1
```

```
11
```

```
13
```



Additional useful command & programs II

grep options:

- -i: ignore case
- -v: invert match
- -n: line number
- -A NUM: print NUM lines after-context
- -B NUM: print NUM lines before-context
- ...



Additional useful command & programs III

cut - remove sections from each line of files

```
cut text_file -f 10 -d ","  
cat text_file | cut -f 10 -d ","
```

```
$cat n.txt  
a,b,c,d,e  
f,g,h,i,j
```

```
$cut -f 2,3 -d "," n.txt  
b,c  
g,h
```



Additional useful command & programs IV

tr - translate or delete characters

```
cat text_file | tr SET1 SET2
```

```
$cat n.txt  
1,2,3,4,5  
6,7,8,9,10
```

```
$cat n.txt | tr "," "_"  
1_2_3_4_5  
6_7_8_9_10
```



Additional useful command & programs V

gzip - compress files

```
gzip text_file
```

gunzip - expand files

```
gunzip file.gz
```

zcat - cat for compressed files

```
zcat file.gz
```

zless - less for compressed files

```
zless file.gz
```



Additional useful command & programs VI

tree - list contents of directories in a tree-like format

```
tree /usr/  
tree /usr/ -d (List directories only)
```

find - search for files in a directory hierarchy

```
find . -name filename.txt  
find . |grep filename.txt  
find . -iname filename.txt (ignore case)  
find . -type f -iname filename.txt (find files only)  
find . -type f -iname -perm 0777 filename.txt (find files only with 777  
permissions)
```



Additional useful command & programs VII

seq - print a sequence of numbers

```
seq 3
```

```
1
```

```
2
```

```
3
```

```
$seq 3
```

```
1
```

```
2
```

```
3
```

```
$seq 11 13
```

```
11
```

```
12
```

```
13
```

```
$seq -f alex%g 4 6
```

```
alex4
```

```
alex5
```

```
alex6
```



Additional useful command & programs VIII

sort - sort lines of text files

```
sort file
```

```
cat file|sort
```

```
$cat n.txt
```

```
10
```

```
5
```

```
4
```

```
3
```

```
2
```

```
1
```

```
$cat n.txt|sort -n
```

```
1
```

```
2
```

```
3
```

```
4
```

```
5
```

```
10
```

```
$cat n.txt|sort
```

```
1
```

```
10
```

```
2
```

```
3
```

```
4
```

```
5
```



Additional useful command & programs IX

`wc` - print newline, word, and byte counts for each file

```
$cat n.txt
```

```
1
```

```
2
```

```
3
```

```
4
```

```
5
```

```
$wc n.txt
```

```
5 5 10 n.txt
```



Variables

```
user@pc$ STR="Hello World!"  
user@pc$ echo $STR  
Hello World!
```



Variables

```
user@pc$ STR="Hello World!"  
user@pc$ echo $STR  
Hello World!
```

```
user@pc$ a=1  
user@pc$ b=$((a + 1 ))  
user@pc$ echo $a "+ 1 = " $b  
1 + 1 = 2
```



Conditional

```
user@pc$ a=1
user@pc$ if [ $a = 1 ]; then
user@pc$ echo true
user@pc$ fi
true
```



Conditional

```
user@pc$ a=1
user@pc$ if [ $a = 1 ]; then
user@pc$ echo true
user@pc$ fi
true
```

```
user@pc$ a=2
user@pc$ if [ $a = 1 ]; then
user@pc$ echo true
user@pc$ else
user@pc$ echo false
user@pc$ fi
false
```



For loop I

```
user@pc$ for i in a b c d
user@pc$ do
user@pc$ echo $i
user@pc$ done
a
b
c
d
```



For loop I

```
user@pc$ for i in a b c d
user@pc$ do
user@pc$ echo $i
user@pc$ done
a
b
c
d
```

```
user@pc$ for i in `seq 1 5`
user@pc$ do
user@pc$ echo $i
user@pc$ done
1
2
3
4
5
```



For loop II

```
user@pc$ for i in `ls /usr/sbin`  
user@pc$ do  
user@pc$ echo $i  
user@pc$ done  
a2disconf  
a2dismod  
a2dissite  
a2enconf  
a2enmod  
a2ensite  
a2query  
...
```



Executable scripts

```
user@pc$ cat runme.sh
#!/bin/bash
echo "Hello World"
```



Executable scripts

```
user@pc$ cat runme.sh  
#!/bin/bash  
echo "Hello World"
```

```
user@pc$ ./runme.sh  
bash: ./runme.sh: Permission denied
```



Executable scripts

```
user@pc$ cat runme.sh
#!/bin/bash
echo "Hello World"
```

```
user@pc$ ./runme.sh
bash: ./runme.sh: Permission denied
```

```
user@pc$ chmod +x ./runme.sh
user@pc$ ./runme.sh
Hello World
```



ssh (Secure Shell)

- Secure
- Encrypted
- Remote connection
 - And many MORE potentials

Two ways of authentication:

- By password
- By key usage



Connection example

```
ssh mypc.uoa.gr
```

```
The authenticity of host 'mypc.uoa.gr (10.100.52.11)' can't be established.  
RSA key fingerprint is c8:03:20:79:18:0d:ea:1d:e3:1c:29:0d:0b:ce:a9:f4.  
Are you sure you want to continue connecting (yes/no)?
```



Creating a pair of private & public keys

- Creating a pair of keys and storing them in `~/.ssh/id_rsa` (**private**) & `~/.ssh/id_rsa.pub` (**public**)

```
ssh-keygen -t rsa
```



Creating a pair of private & public keys

- Creating a pair of keys and storing them in `~/.ssh/id_rsa` (**private**) & `~/.ssh/id_rsa.pub` (**public**)

```
ssh-keygen -t rsa
```

```
Generating public/private rsa key pair.
```

```
Enter passphrase (empty for no passphrase):
```

```
Enter same passphrase again:
```

```
Your identification has been saved in ~/.ssh/id_rsa.
```

```
Your public key has been saved in ~/.ssh/id_rsa.pub.
```

```
The key fingerprint is:
```

```
ca:0f:15:49:09:2e:e9:d8:59:16:8b:8c:30:d2:b9:77 root@snf
```

```
The key's randomart image is:
```

```
+--[ RSA 2048 ]-----+
|  ..o.      +++++  |
|              |
+-----+

```

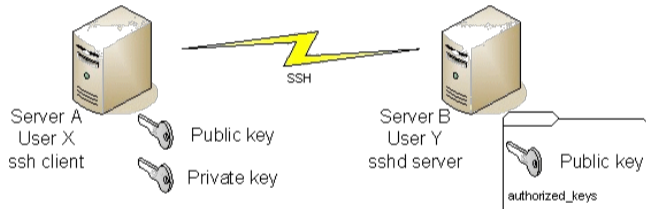


Storing a “foreign” public key

```
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys2  
chmod 644 ~/.ssh/authorized_keys2
```



How it works?



VPS on Genomics Lab I



"ALEXANDER FLEMING"
Biomedical Sciences Research Center

- Virtual private server (VPS)
 - 16 CPUs
 - 32 GB RAM
 - 100 + 2TB HDDs (with quotas)
 - IPv4
 - running Ubuntu 24.04
- suggested for executing lab exercises and final project
- connect using ssh key (~~or password ??~~)
- graphical interface via X2Go (<https://wiki.x2go.org/doku.php>)



Exercise 2 - Familiarizing with GNU/Linux CLI

- Create directory
- Rename directory
- Move directory
- Delete directory
- ...

Submit via e-class assignment

<https://eclass.uoa.gr/modules/work/index.php?course=DI425&id=62479>

OR by email at alexdem@di.uoa.gr

<https://eclass.uoa.gr/modules/document/file.php/DI425/2024-25/exercises/ITBI2024-exercise2-ACD17102024.pdf>

DEADLINE 31/10/24



Questions?

