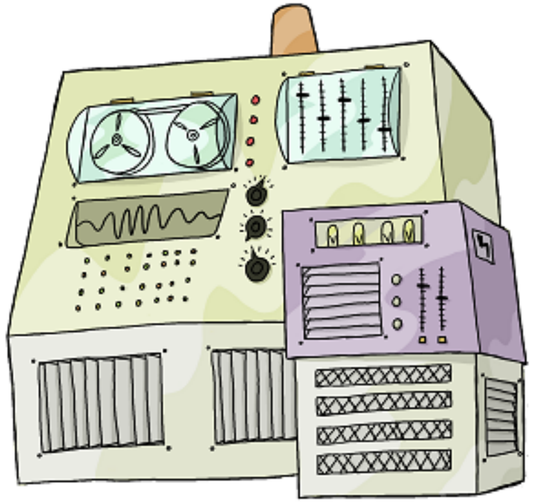
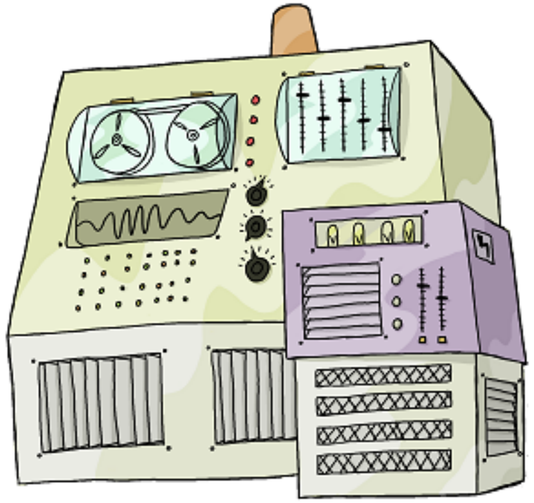
Introduction to Unix Shell

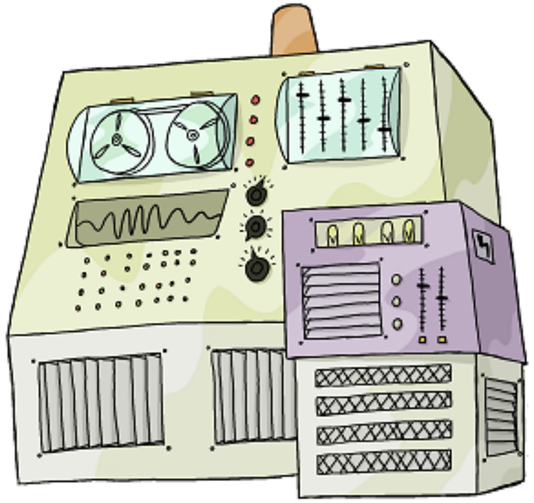
David Castillo, Marco Di Stefano, Marc A. Marti-Renom
Genome Biology Group (CNAG)
Structural Genomics Group (CRG)





Run

Programs

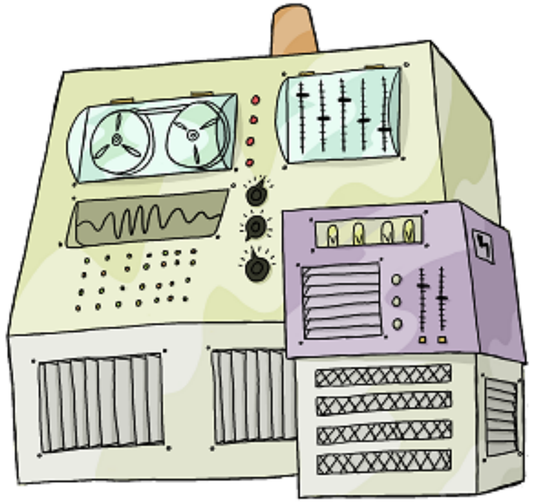


Run

Programs

Store

Data



Run

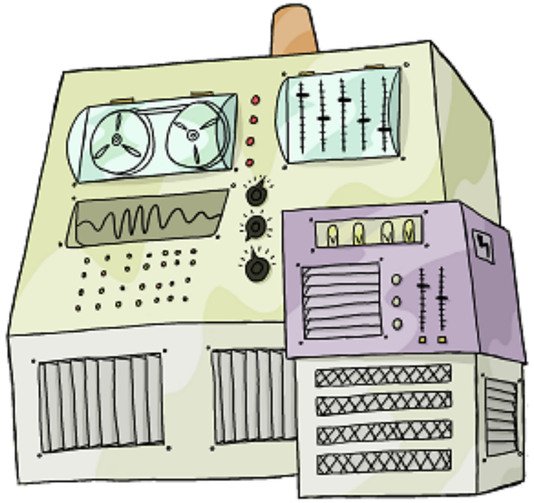
Programs

Store

Data

Communicate

with each other



Run Programs

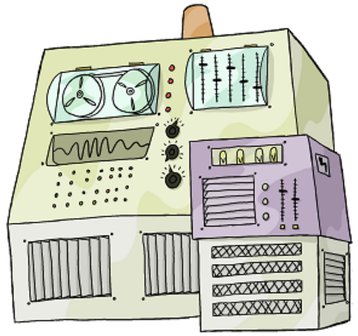
Interact with us

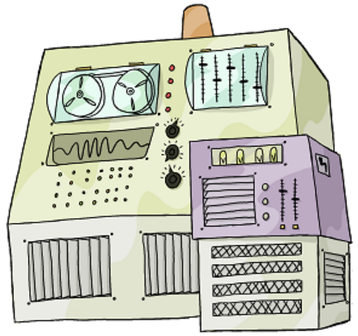
Store Data

Communicate with each other

Interact

with us





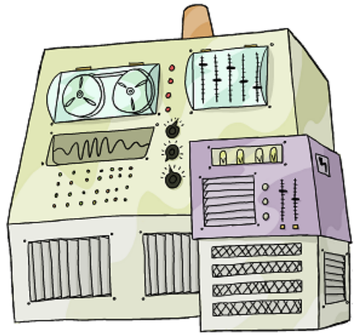
Interact

with us

WIMP

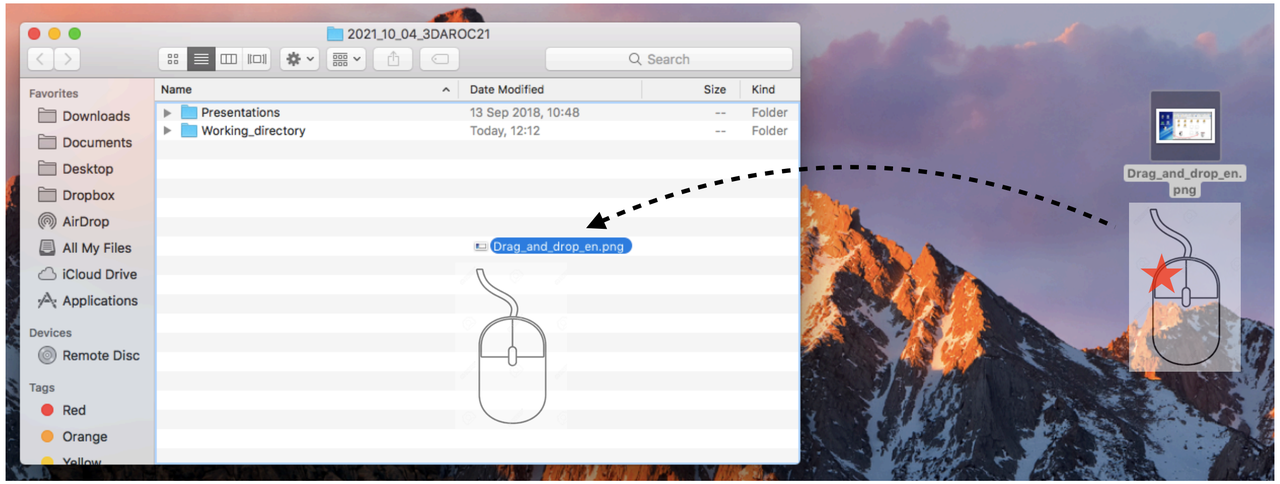
(windows, icons, menus, pointers)

(windows, icons, mice, pointers)



Interact

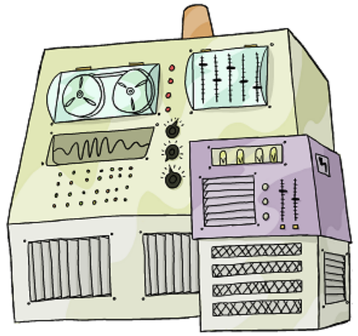
with us



WIMP

(windows, icons, menus, pointers)

(windows, icons, mice, pointers)



Interact

with us

Typewriter

WIMP

(windows, icons, menus, pointers)

(windows, icons, mice, pointers)

user logs in

user logs in

user types command

user logs in

user types command

computer executes command

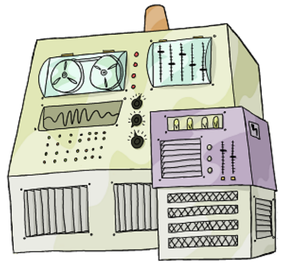
and prints output

user logs in
user types command
computer executes command
  and prints output
user types another command

user logs in
user types command
computer executes command
  and prints output
user types another command
computer executes command
  and prints output

user logs in
user types command
computer executes command
  and prints output
user types another command
computer executes command
  and prints output
⋮
user logs off

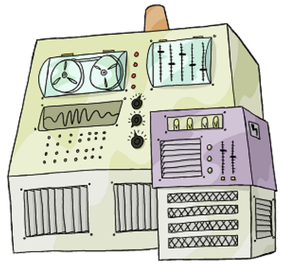
user logs in
user types command
computer executes command
  and prints output
user types another command
computer executes command
  and prints output
⋮
user logs off



user logs in
user types command
computer executes command
and prints output
user types another command
computer executes command
and prints output
⋮
user logs off

Uses characters and numbers to write

Uses binary numbers to operate

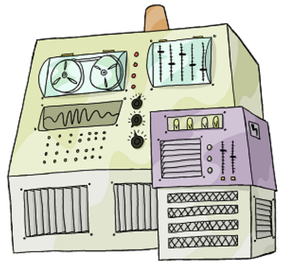


user logs in
user types command
computer executes command
  and prints output
user types another command
computer executes command
  and prints output
⋮
user logs off

Uses characters and
numbers to write

?

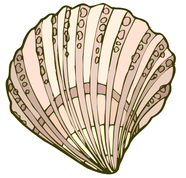
Uses binary numbers
to operate



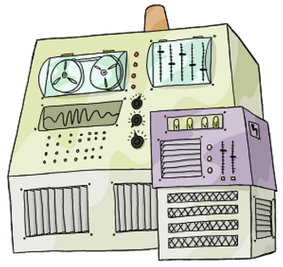
user logs in
user types command
computer executes command
and prints output
user types another command
computer executes command
and prints output
⋮
user logs off

Uses characters and numbers to write

shell



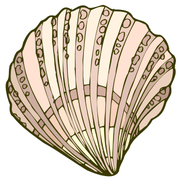
Uses binary numbers to operate



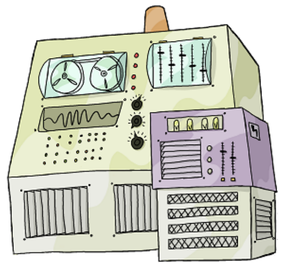
Uses characters and numbers to write

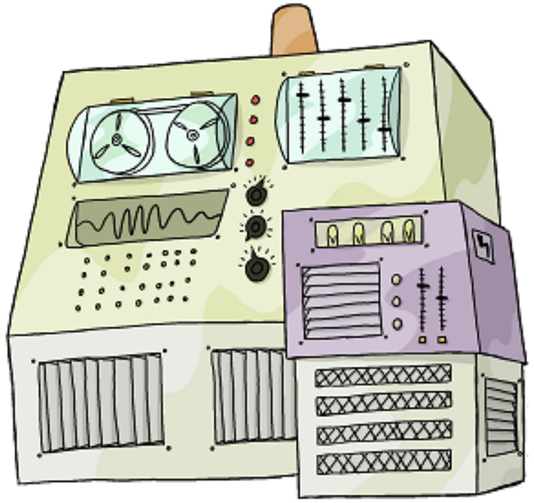
user logs in
user types command
computer executes command
and prints output
user types another command
computer executes command
and prints output
⋮
user logs off

shell



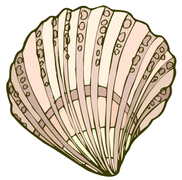
Uses binary numbers to operate





Run Programs

shell



Interact with us

Store Data

Communicate with each other

login: vlad

password: ******

$ shell prompt

login: vlad

password: ******

$

shell prompt

like Python's >>> and …

```
login: vlad
password: ********
$ whoami
```

check user ID

login: vlad

password: ******

$ whoami

check user ID

shell finds the **whoami** program

login: vlad

password: ******

$ whoami

check user ID

shell finds the **whoami** program

runs it

```
login: vlad
password: ********
$ whoami
vlad
```

check user ID

shell finds the **whoami** program

runs it

prints its output

```
login: vlad
password: ********
$ whoami
vlad
$
```

check user ID

shell finds the **whoami** program

runs it

prints its output

displays a new prompt

```
login: vlad
password: ********
$ whoami
vlad
$ pwd
```

what is the *working directory*

```
login: vlad
password: ********
$ whoami
vlad
$ pwd
```

what is the *working directory*

the directory used when no other directory is explicitly specified

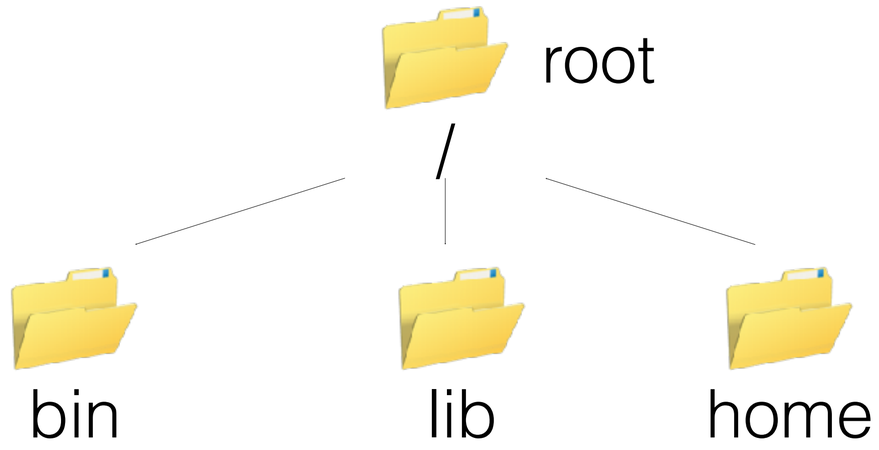
```
login: vlad
password: ********
$ whoami
vlad
$ pwd
/home/vlad
$
```

```
login: vlad
password: ********
$ whoami
vlad
$ pwd
/home/vlad
$
```

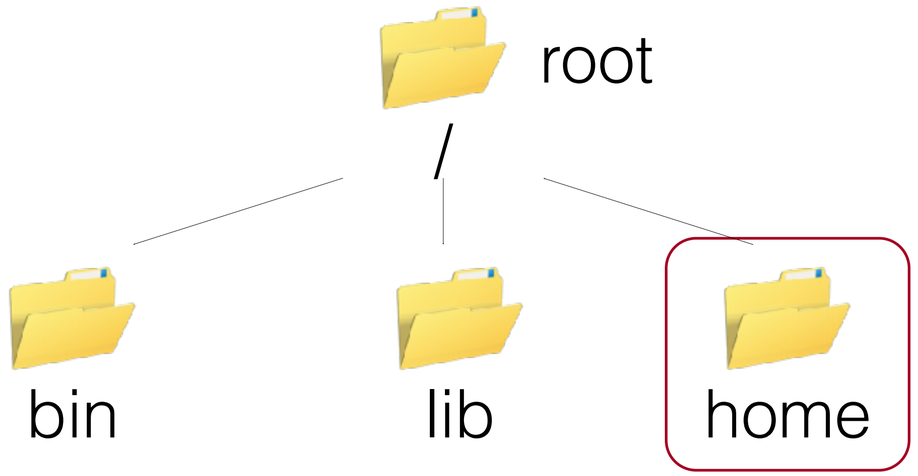
root

/

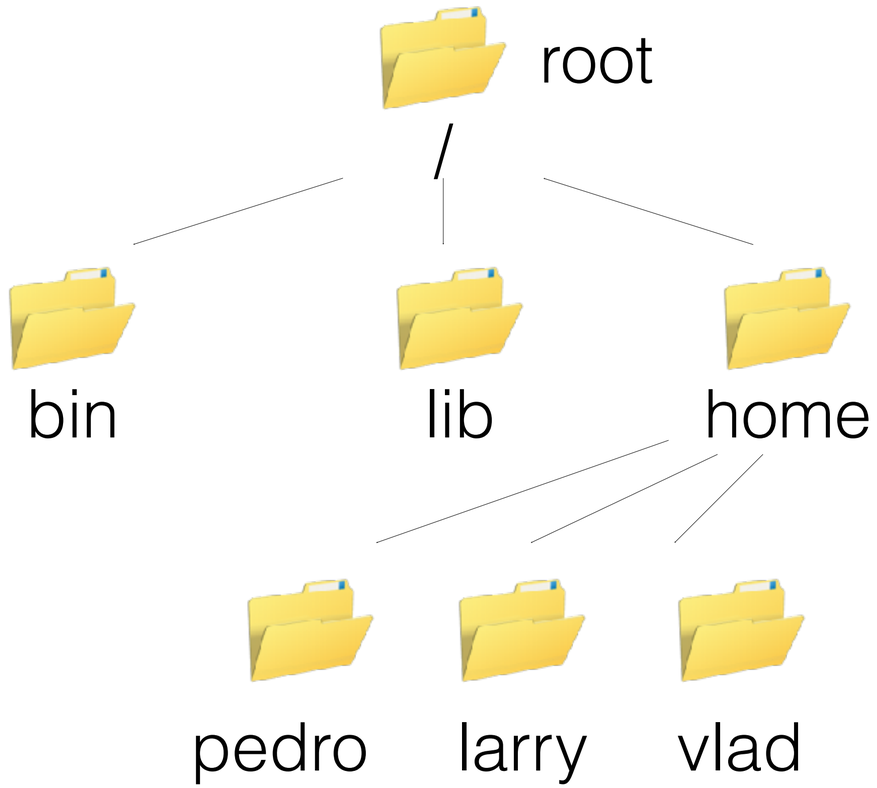
```
login: vlad
password: ********
$ whoami
vlad
$ pwd
/home/vlad
$
```



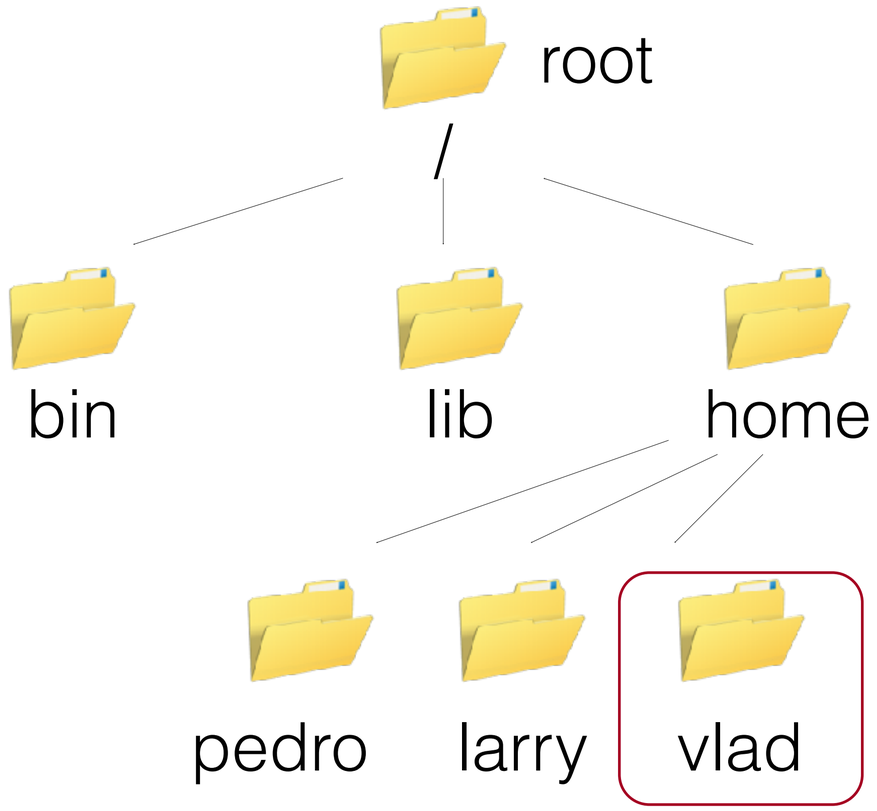
```
login: vlad
password: ********
$ whoami
vlad
$ pwd
/home/vlad
$
```



```
login: vlad
password: ********
$ whoami
vlad
$ pwd
/home/vlad
$
```



```
login: vlad
password: ********
$ whoami
vlad
$ pwd
/home/vlad
$
```



```
login: vlad
password: ********
$ whoami
vlad
$ pwd
/home/vlad
$ ls
```

stands for "listing"

```
login: vlad
password: ********
$ whoami
vlad
$ pwd
/home/vlad
$ ls
```

stands for "listing"
sadly more memorable than
most command names

```
login: vlad
password: ********
$ whoami
vlad
$ pwd
/home/vlad
$ ls
bin        data     mail     music
notes.txt  papers
$
```

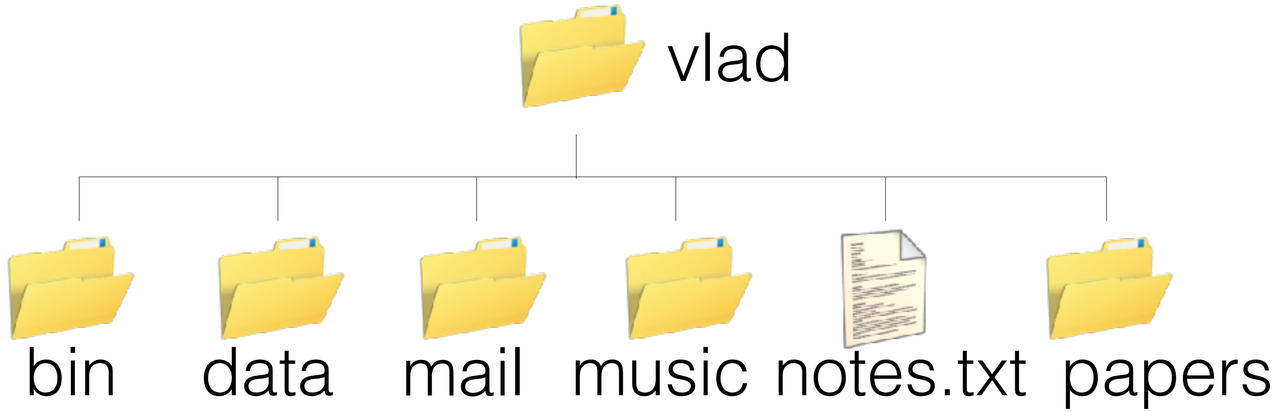
```
login: vlad
password: ********
$ whoami
vlad
$ pwd
/home/vlad
$ ls -F
bin/        data/     mail/      music/
notes.txt   papers/
$
```

an *argument* or *flag* modifying the command's behavior

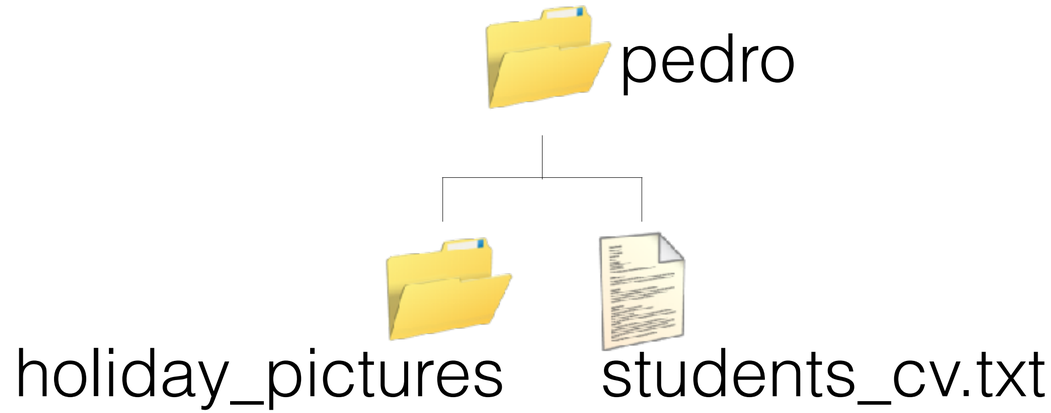
```
login: vlad
password: ********
$ whoami
vlad
$ pwd
/home/vlad
$ ls -F
bin/        data/     mail/     music/
notes.txt   papers/
$
```

adds a trailing '/' to directory names

```
$ ls -F
bin/        data/    mail/     music/
notes.txt   papers/
```



```
$ pwd
/home/vlad
$ ls /home/pedro -F
holiday_pictures/  students_cv.txt
```




```
$ ls ../pedro -F
holiday_pictures/  students_cv.txt
```

Lets start!!

Command	Description	Example	Action
`pwd`	print working directory	`pwd`	path and name of working dir. I am in now
`ls`	list contents of directory	`ls`	list contents of current dir.
		`ls test/`	list contents of the test dir. that hangs from the working dir.
		`ls -lh`	vertical list of dir. contents
`cd`	change directory	`cd`	go to home directory (same as `cd /home/user` or `cd ~/`)
		`cd /home/user/Docs`	go to the Docs directory
		`cd ..`	go to parent directory
`mkdir`	make directory	`mkdir test`	creates directory test/
`rmdir`	remove directory	`rmdir test`	remove test/ if empty
`rm`	remove file	`rm test.txt`	remove test.txt file
`cp`	copy	`cp fileA fileB`	copy fileA to fileB
`mv`	move or rename file or directory	`mv a b`	change name from a to b
		`mv a ..`	move a to parent directory
`more`	see file content	`more a.txt`	see contents of a.txt page by page
`cat`	see file content	`cat a.txt`	see contents of a.txt page, all the file at once
`head/tail`	see first/last lines of a file	`head -n 10 a.txt`	see the first 10 lines a file a.txt (last 10 would be with `tail`)
`zcat`	like cat but fir zipped files	`zcat a.txt.gz | head`	see first 10 lines of a compressed file a.txt.gz
`nano`	simple text editor!	`nano a.txt`	edit a.txt (ctrl-X to exit)
`firefox`	a web browser	`firefox a.html or firefox a.jpg`	use web browser to view file
`ssh`	connect to a remote server	`ssh student@172.17.133.110`	go to the home folder of “student” in a given server
`scp`	copy from a remote server	`scp -r student@172.17.133.110:~/test .`	copy test directory from remote server at current pwd
`man`	manual on a command	`man ls`	manual page for the 'ls' command