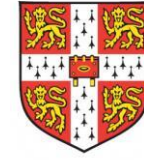




Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin



UNIVERSITY OF
CAMBRIDGE

Introduction to Linux (and the terminal)

27/11/2018

Pierpaolo Maisano Delser

mail: maisano@tcd.ie ; pm604@cam.ac.uk

Outline:

- What is Linux and the terminal?
- Why do we use the terminal?
- Pros and cons
- Basic commands and examples

What is Linux and the terminal?

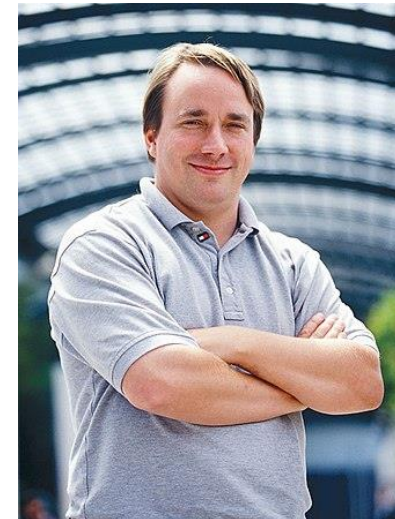
Linux:

- Linux is an operating system (OS):
 - Interaction between applications, computer operator and computer's processor;



Linux:

- Linux is an operating system (OS):
 - Interaction between applications, computer operator and computer's processor;
- History of Linux began in 1991 with Linnus Torvalds (free operating system)



Before Linux:

- In the 80's Microsoft's DOS was the main OS for PCs;
- Apple MAC was better but more expensive;
- UNIX was much better but much more expensive (only for minicomputers for commercial applications);
- DOS, MAC and UNIX were proprietary (high licence fees to modify them);
- Ideally: UNIX-like system but free!

Linux:

- Minix: simplified version of UNIX that runs on PC for class teaching only;
- In 1991 Linnus Torvalds started to develop Linux kernel (freely distributed code);



Linux today:

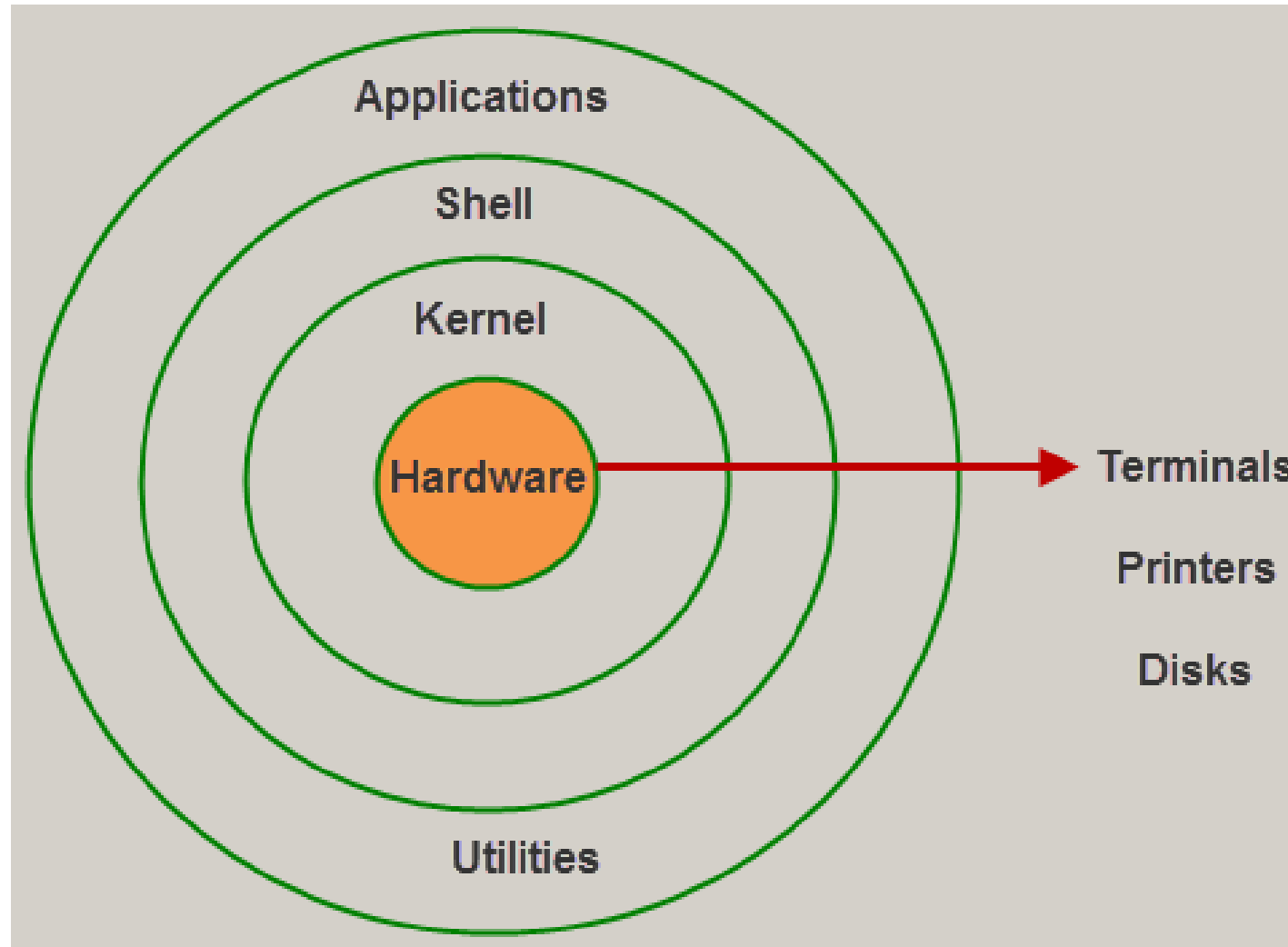
- Not only character user interface but graphical user interface is available;
- Widely used in HPC facilities, smartphones, personal laptop, companies, etc...
- It is a free software:
 - run the program for any purpose (not restricted);
 - you can view and modify the program's source code;
 - you can share the modified code;
 - you can improve features so that others can use them;

Linux today:

- Several versions with extremely good graphical interface (i.e. Ubuntu);
- You can find all corresponding programs that work on Windows and Mac (i.e. Microsoft Office – OpenOffice);
- User-friendly installation and guides;
- Large number of website providing support, help and suggestions;
- Do not be scare to google all your questions/problems!!

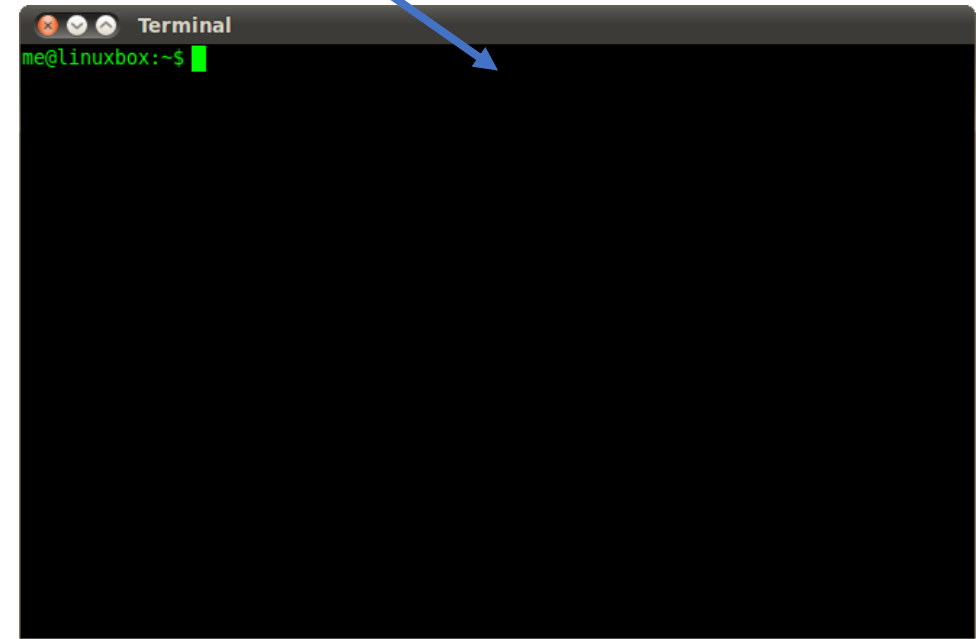
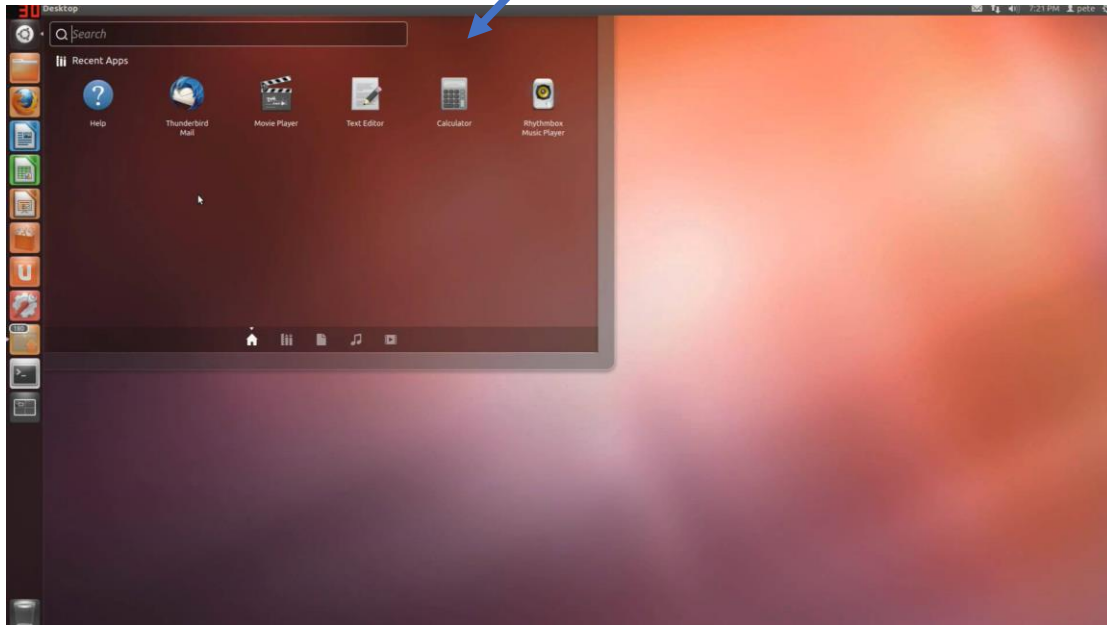
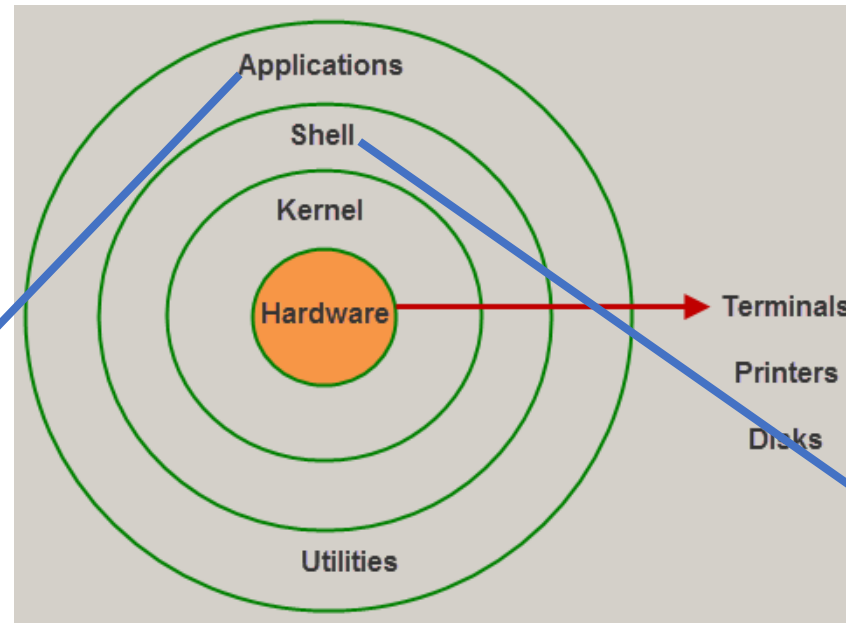
What is Linux and the terminal?

Linux terminal – the shell:



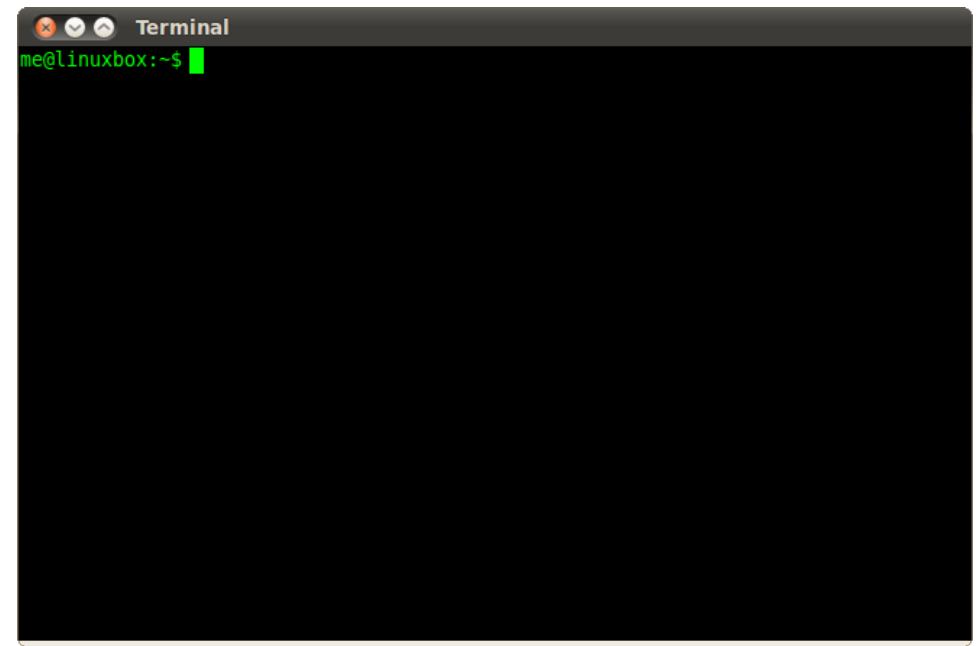
What is Linux and the terminal?

Linux terminal – the shell:



What is Linux and the terminal?

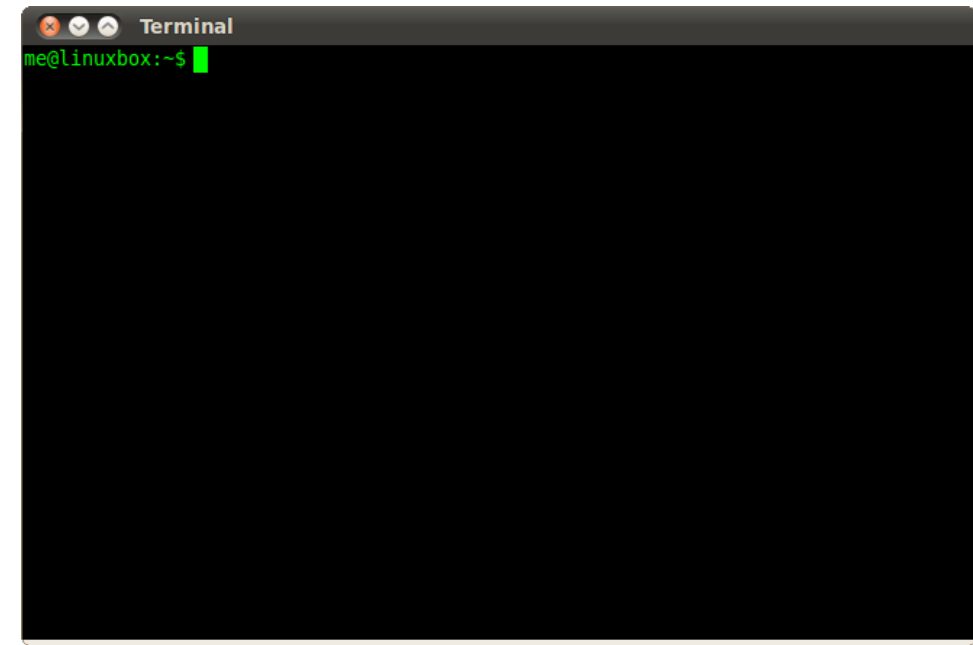
Linux terminal – the shell (“black box”):



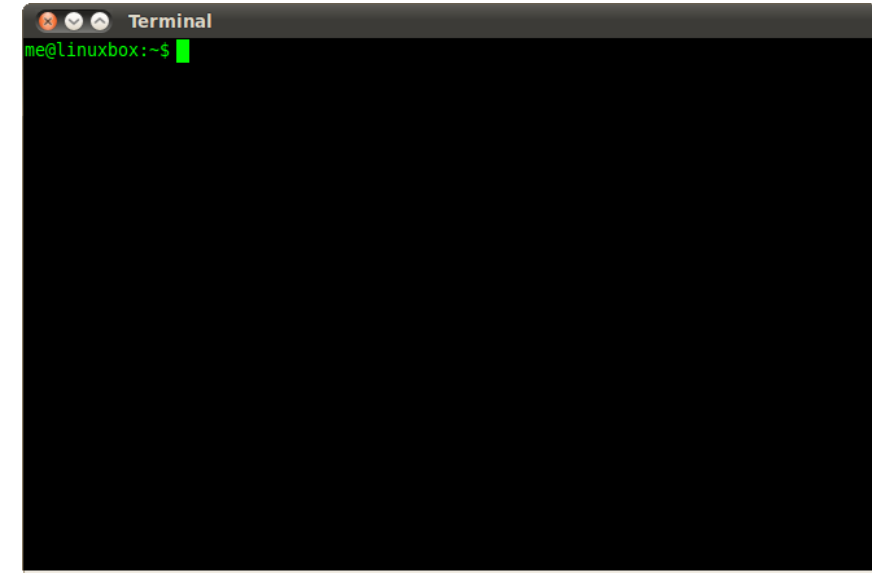
- Slightly different types of shell but the most common one is called **Bash** (Brian Fox called it “Bourne-again shell”);
- Bash shell contains a set of commands for multiple operations;
- Flexible to operate with files and folders (directories);

Bash shell (terminal)...WHY???

- No graphical features (less memory demanding);
- Very powerful and flexible for multiple operations with files and folders;
- It is shared across multiple system – same “language”;
- Can be used to connect to HPC (clusters) and other machines;
- It provides a shared common environment for people to work together;



Bash shell (terminal)...WHY???



- You can have a quick look at files without fully opening them;
- You can copy, paste and transfer files within your computer and among different computers;
- You can modify file, look for patterns, extract specific lines and much more!

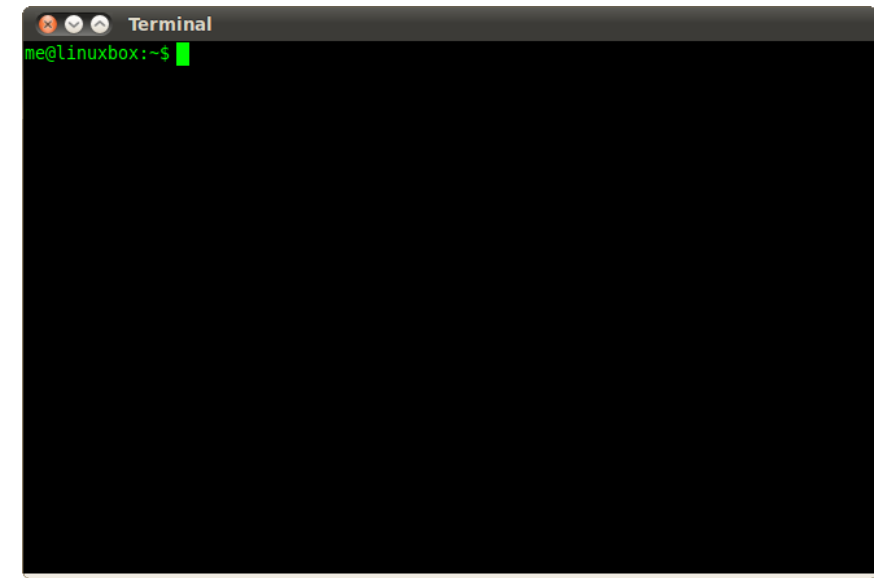
Pros and cons (Linux)

Pros:

- Flexible, open access, great graphical interface and...the terminal!
- Lots of resources and support on line;

Cons:

- Some tools/programs are not at the same level/standard as the corresponding ones in Windows/Mac;
- You have more freedom than in Windows and Mac but you need more time and patience!



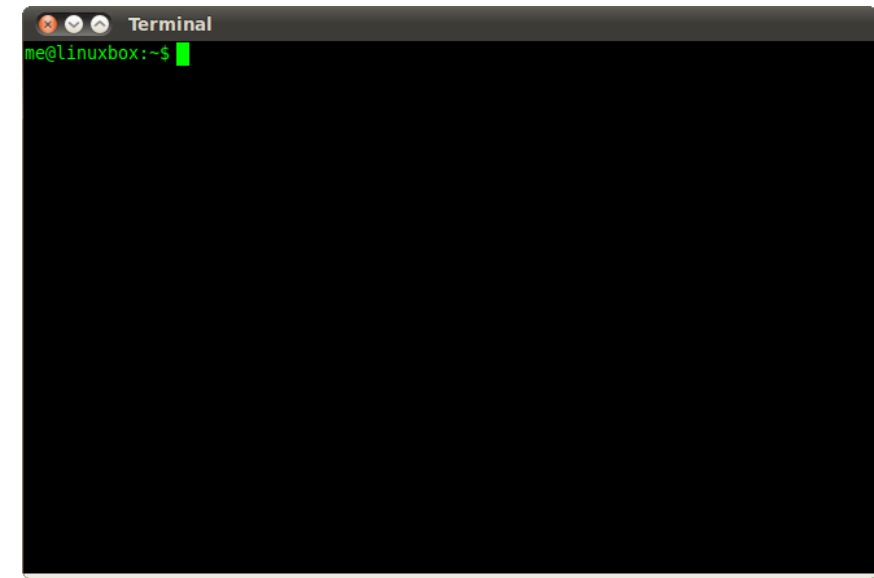
Pros and cons (shell)

Pros:

- Great system for working with remote servers and HPC platforms;
- Lots of resources and support on line;

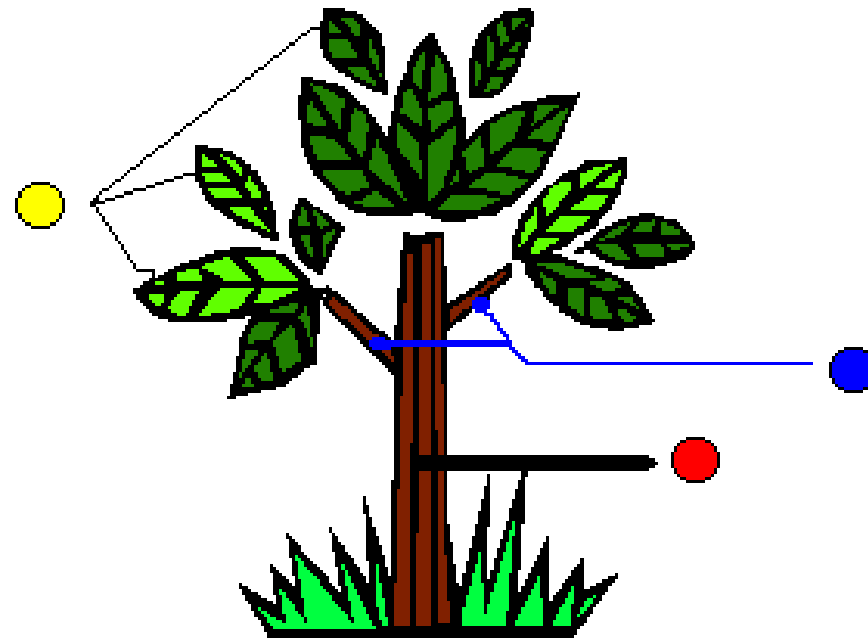
Cons:

- Takes time to get used to it and learn tips and tricks to work faster;
- Overcoming the black-box concept;



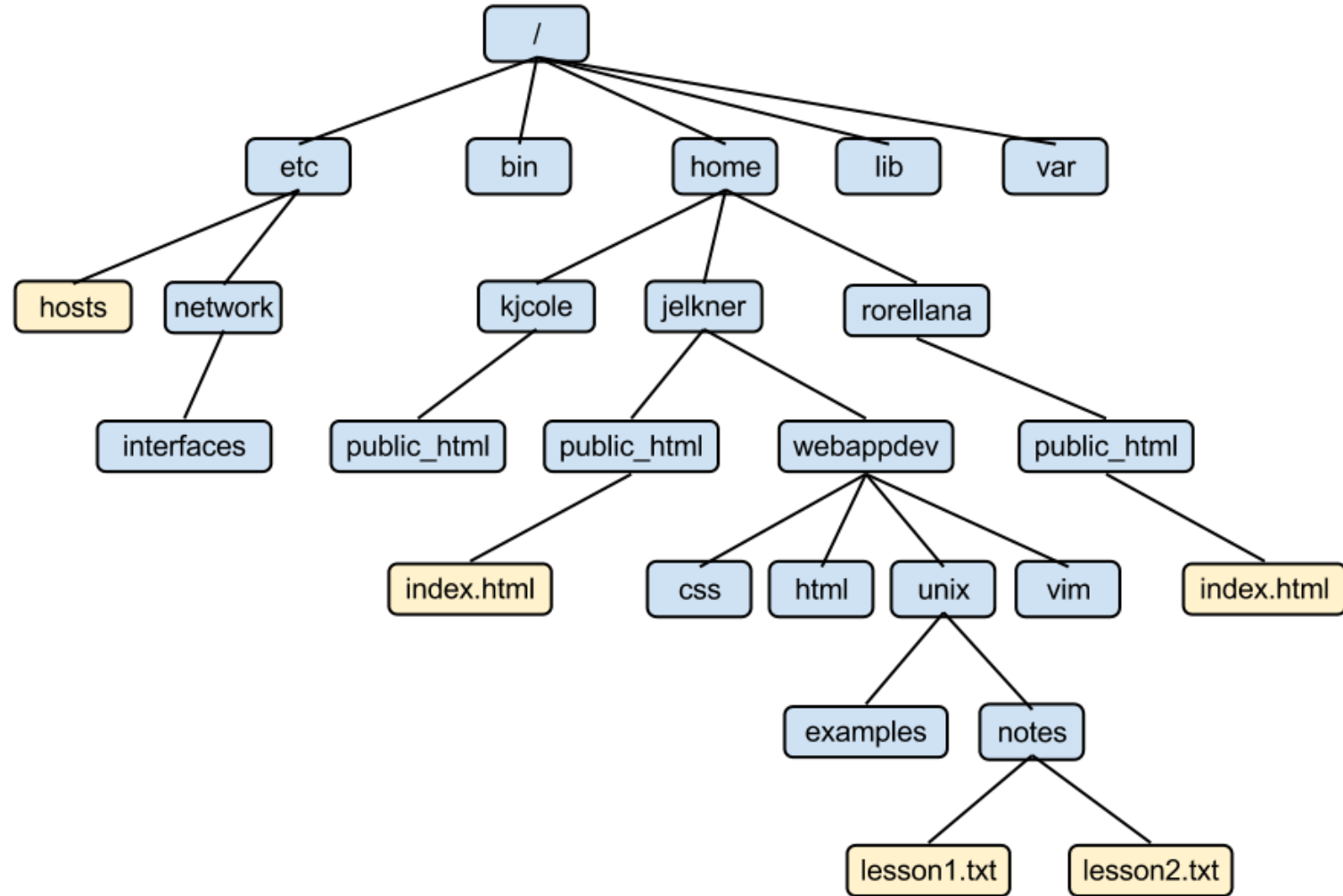
Linux structure – hierarchical structure

- Files
- Subdirectories
(branches of Tree)
- Root



Linux File System is just like a tree

Linux structure – hierarchical structure



```
pwd
```

“Print working directory” ---- Tell me where I am

```
/home/user/pier/Documents
```

```
pwd
```

“Print working directory” ---- Tell me where I am

```
/home/user/pier/Documents
```

```
ls
```

“List directory/files” ---- Tell me which directories/files are in the folder where I am now

```
“Teaching” “Images” “Data”
```

```
pwd
```

“Print working directory” ---- Tell me where I am

```
/home/user/pier/Documents
```

```
ls
```

“List directory/files” ---- Tell me which directories/files are in the folder where I am now

```
“Teaching” “Images” “Data”
```

```
cd “Name of Directory”
```

“Change Directory” ---- enter a different directory

```
cd Teaching
```

Basic commands and examples

```
pwd
```

```
/home/user/pier/Documents/Teaching
```

Basic commands and examples

```
pwd
```

```
/home/user/pier/Documents/Teaching
```

```
cd ..
```

“..” means “the directory above”

```
pwd
```

```
/home/user/pier/Documents
```

Basic commands and examples

```
cd ~
```

“~” means “home directory”

```
/home
```

Basic commands and examples

```
cd ~
```

“~” means “home directory”

```
/home
```

```
cd /
```

Go to the root

```
/
```


Basic commands and examples

```
cd ~
```

“~” means “home directory”

```
/home
```

```
cd /
```

Go to the root

```
/
```

```
cd /home/user/pier/Documents
```

```
cd ../..
```

/home/user/

Basic commands and examples

```
cd /home/user/pier/Documents
```

```
mkdir course
```

Make (create) new directory called “course”

```
ls
```

“Teaching” “Images” “Data” “course”

```
cd /home/user/pier/Documents
```

```
mkdir course
```

 Make (create) new directory called “course”

```
ls
```

“Teaching” “Images” “Data” “course”

```
rm -rf course
```

 Delete a folder and all contents

THERE IS NO “BIN” IN THE TERMINAL...ONCE IT IS DELETED, IT IS GONE FOREVER!!!!

Basic commands and examples

```
echo "Hello, how are you?"
```

Hello, how are you?

```
echo "Hello, how are you?" > output.txt
```

Prints to a file

```
ls
```

"output.txt"

```
rm output.txt
```

THERE IS NO "BIN" IN THE TERMINAL...ONCE IT IS DELETED, IT IS GONE FOREVER!!!!

```
cp lecture1.txt lecture2.txt
```

Copy a file and with a different name

```
ls
```

“lecture1.txt” “lecture2.txt”

Basic commands and examples

```
cp lecture1.txt lecture2.txt
```

Copy a file and with a different name

```
ls
```

“lecture1.txt” “lecture2.txt”

```
rm lesson1.txt
```

Remove/delete lesson1.txt

```
ls
```

“lecture2.txt”

```
mv /home/user/pier/lesson1.txt /home/user/pier/Documents/course/
```

Move a file

```
mv /home/user/pier/lesson1.txt /home/user/pier/Documents/course/
```

Move a file

```
ls
```

“lecture1.txt”

```
mv lecture1.txt lecture2.txt
```

Rename a file

```
ls
```

“lecture2.txt”

Whitespaces matter!

```
cd my results
```

Bash: cd: my: No such file or directory

```
cd my\ results
```

```
cd myResults
```

```
cd my_results
```

```
cd myresults
```

CASE matters!

```
mkdir data
```

```
cd Data
```

```
cd dAta
```

```
cd data
```

```
cd dATa
```

```
cd datA
```

CASE matters!

```
mkdir data
```

```
cd Data
```

ERROR!

```
cd dAta
```

ERROR!

```
cd data
```

CORRECT

```
cd dATa
```

ERROR!

```
cd datA
```

ERROR!

Tab completion can help!

```
cd d[tab] ---- cd data
```

Commands working with files:

```
ls
```

“notes1.txt”

```
cat notes1.txt
```

It will print the content of the all file (all lines) on the terminal

```
head notes1.txt
```

It will print the first 10 lines on the terminal

```
tail notes1.txt
```

It will print the last 10 lines on the terminal

Commands working with files:

```
head -n 25 notes1.txt
```

It will print the first 25 lines on the terminal

```
tail -n 15 notes1.txt
```

It will print the last 15 lines on the terminal

```
tail -n+2 notes1.txt
```

It will print all the lines starting from line 2 (remove header)

Commands working with files:

More, less...less is more??

```
more notes1.txt
```

Scroll downwards, one page at a time

```
less notes1.txt
```

Scroll downwards and upwards

“less” is faster...“more” needs to load the whole file

Less is more!!!

Commands working with files:

Grep:

```
grep "course" notes1.txt
```

It looks for the word "course" and prints out all lines containing the this word

```
grep -v "course" notes1.txt
```

It prints out all lines not containing the word "course"

```
grep -c "course" notes1.txt
```

It counts how many lines contain the word "course" in the document "notes1.txt"

Commands working with files:

Cut: extracts portion of text from a file by selecting columns

```
cut -c2 notes1.txt
```

It displays 2nd character from each line from “notes1.txt”

```
cut -c1-3 notes1.txt
```

It displays the first 3 characters of each line from “notes1.txt”

```
cut -c3- notes1.txt
```

It extracts from 3rd character to end of each line from “notes1.txt”

Commands working with files:

Cut: extracts portion of text from a file by selecting columns

```
cut -c-8 notes1.txt
```

It extracts 8 characters from the beginning of each line from “notes1.txt”

```
cut -d' :' -f1 notes1.txt
```

It displays only first field of each lines from “notes1.txt” using the field delimiter : (colon).

```
cut -d' :' -f1-4,6,7 notes1.txt
```

It displays field 1 through 4, 6 and 7 of each lines from “notes1.txt” using the field delimiter : (colon).

Commands working with files:

Sort: sorts the contents of a text file, line by line.

Rules:

1. Numbers before letters;
2. Alphabetically;
3. Lowercase before uppercase (for same letter);

```
apples  
oranges  
pears  
kiwis  
bananas
```

```
sort data.txt
```



```
apples  
bananas  
kiwis  
oranges  
pears
```

Commands working with files:

Sort: sorts the contents of a text file, line by line.

```
sort data.txt > data_sorted.txt
```

```
sort -o data_sorted.txt data.txt
```

```
apples  
oranges  
pears  
kiwis  
bananas
```

```
sort -r data.txt
```



```
pears  
oranges  
kiwis  
bananas  
apples
```

Commands working with files:

Sort: sorts the contents of a text file, line by line.

Rules:

1. Numbers before letters;
2. Alphabetically;
3. Lowercase before uppercase (for same letter);

```
02  
A  
a  
03  
cat
```

```
sort data.txt
```



```
02  
03  
a  
A  
cat
```

Commands working with files:

Uniq: reports or filters out repeated lines in a file..

```
Cat  
Cat  
Parrott  
Dog  
Dog
```

```
uniq data.txt
```



```
Cat  
Parrott  
Dog
```

```
Cat  
Cat  
Parrott  
Dog  
Dog
```

```
uniq -c data.txt
```



```
2 Cat  
1 Parrott  
2 Dog
```

Commands working with files:

Uniq: reports or filters out repeated lines in a file..

```
Cat  
Cat  
Parrott  
Dog  
Dog
```

```
uniq -d data.txt
```



```
Cat  
Dog
```

```
Cat  
Cat  
Parrott  
Dog  
Dog
```

```
sort data.txt | uniq
```



```
Cat  
Dog  
Parrott
```

Commands working with files:

wc: word count

```
White cat  
Sun  
Blue Parrott  
Cute dog  
Green tree
```

```
wc data.txt
```



```
5 9 47 data.txt
```

```
White cat  
Sun  
Blue Parrott  
Cute dog  
Green tree
```

```
wc -l data.txt
```



```
5 data.txt
```

Commands working with files:

wc: word count

```
White cat  
Sun  
Blue Parrott  
Cute dog  
Green tree
```

```
wc -L data.txt
```



```
13 data.txt
```


Commands working with files:

sed: edits data based on the rules you provide

```
White cat  
Sun  
Blue Parrott  
Cute dog  
Green tree
```

```
sed 's/cat/dog/' data.txt
```



```
White dog  
Sun  
Blue Parrott  
Cute dog  
Green tree
```

```
White cat  
Sun  
Blue Parrott  
Cute dog  
Green tree
```

```
sed -e 's/cat/dog/;  
s/Green/red/' data.txt
```



```
White dog  
Sun  
Blue Parrott  
Cute dog  
red tree
```

Commands working with files:

sed: edits data based on the rules you provide

```
White cat  
Cute cat  
Green tree  
Red tree  
Orange tree
```

```
sed 's/cat/dog/' data.txt
```



```
White dog  
Cute cat  
Green tree  
Red tree  
Orange tree
```

```
White cat  
Cute cat  
Green tree  
Red tree  
Orange tree
```

```
sed 's/cat/dog/g' data.txt
```



```
White dog  
Cute dog  
Green tree  
Red tree  
Orange tree
```

Commands working with files:

sed: edits data based on the rules you provide

```
White cat  
Cute cat  
Green tree  
Red tree  
Orange tree
```

```
sed 's/tree/cat/2' data.txt
```



```
White cat  
Cute cat  
Green cat  
Red cat  
Orange tree
```

```
White cat  
Cute cat  
Green tree  
Red tree  
Orange tree
```

```
sed 's/tree/cat/g' data.txt
```



```
White cat  
Cute cat  
Green cat  
Red cat  
Orange cat
```

Connect to server in Cambridge

Conclusions:

- Linux is user-friendly, the shell needs a bit of more time to be user-friendly;
- The shell/terminal is extremely powerful to work with file, edit text and connect to remote server;

Conclusions:

- Linux is user-friendly, the shell needs a bit of more time to be user-friendly;
- The shell/terminal is extremely powerful to work with file, edit text and connect to remote server;
- There are many commands for the shell...google your questions and you will find lots of support!
- Combine commands together to perform complex operation in one line;

Conclusions:

- Linux is user-friendly, the shell needs a bit of more time to be user-friendly;
- The shell/terminal is extremely powerful to work with file, edit text and connect to remote server;
- There are many commands for the shell...google your questions and you will find lots of support!
- Combine commands together to perform complex operation in one line;
- Do not be scared of the “black-box” ...it will be your best friend!