Introduction to the LINUX Environment: Basic commands and scripting

Kenneth Kim
Core Facility for Bioinformatics
Philippine Genome Center

PGC
PHILIPPINE GENOME CENTER

DO NOT FEAR THE COMMAND LINE

What is Unix/Linux?

UNIX isn’t free.

Linus Torvalds, a young man studying computer science at the university of Helsinki, thought it would be a good idea to have some sort of freely available academic version of UNIX, and promptly started to code.

LINUX

What is a kernel and a shell?

The kernel

The kernel is the hub of the operating system. It allocates time and memory to programs and handles the filestore and communications in response to system calls.

The shell

Shell is a program that takes commands from the keyboard and gives them to the operating system to perform.

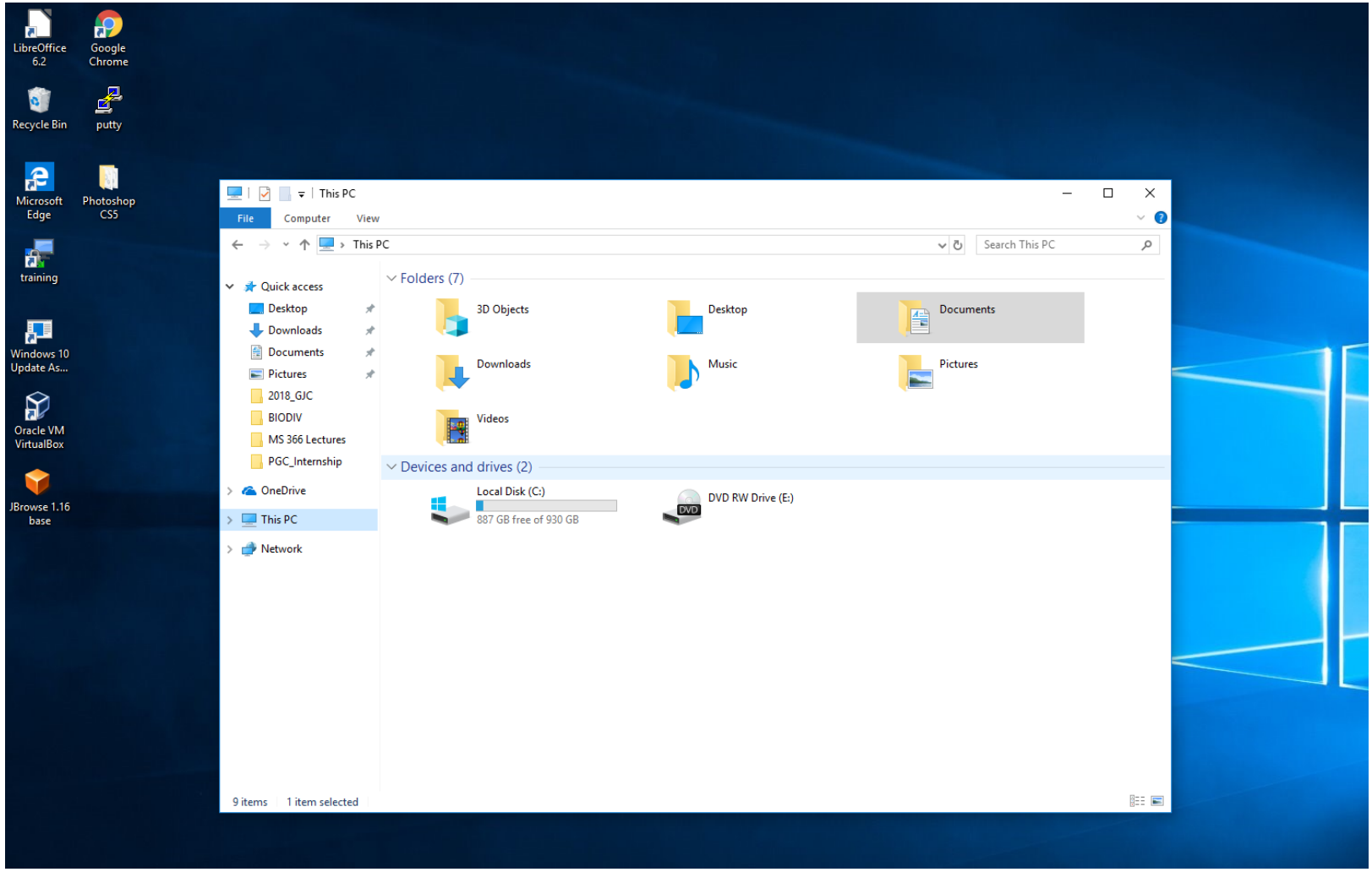
On most Linux systems a program called bash (which stands for Bourne Again SHell) acts as the shell program.

The Linux shell

- The shell is a command-line interpreter that lets you interact with Linux
- The shell takes what you type and "decides" what to do with it

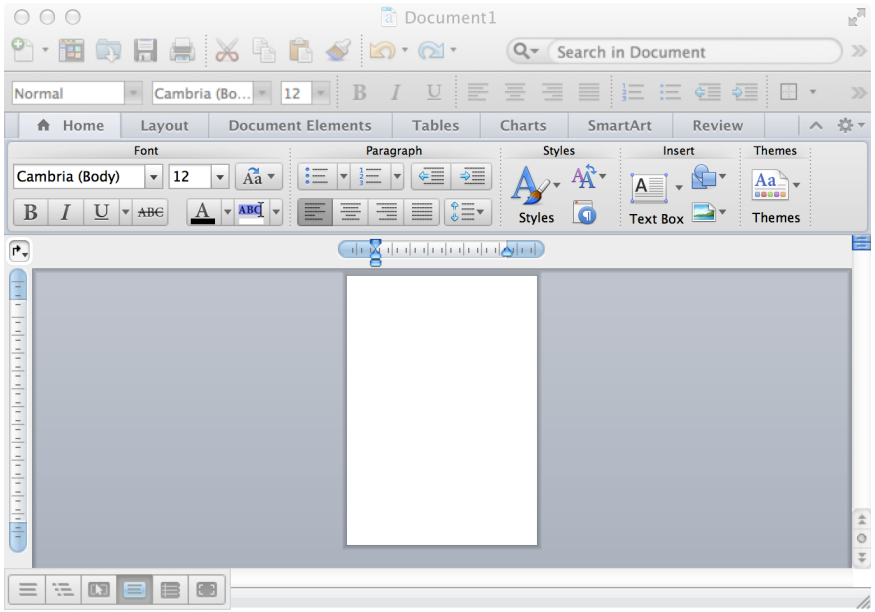
What is the graphical interface?

graphical interface

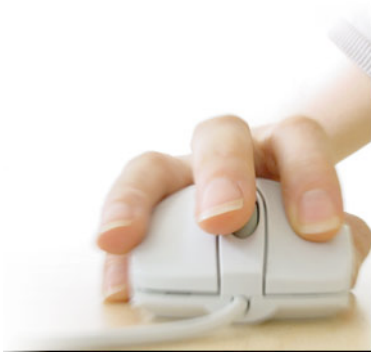


What is the command line interface (or Terminal)?

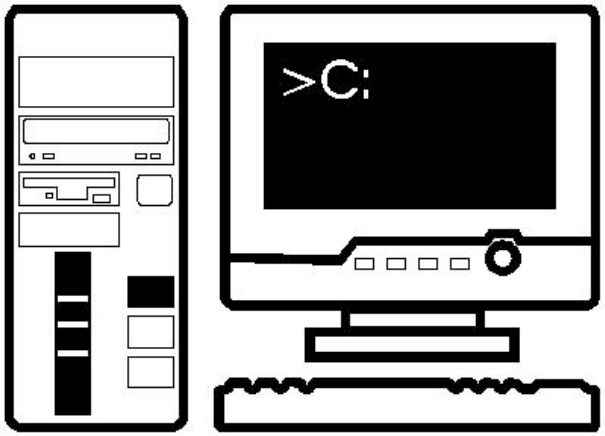
What happens when you double click on the icon of an application?

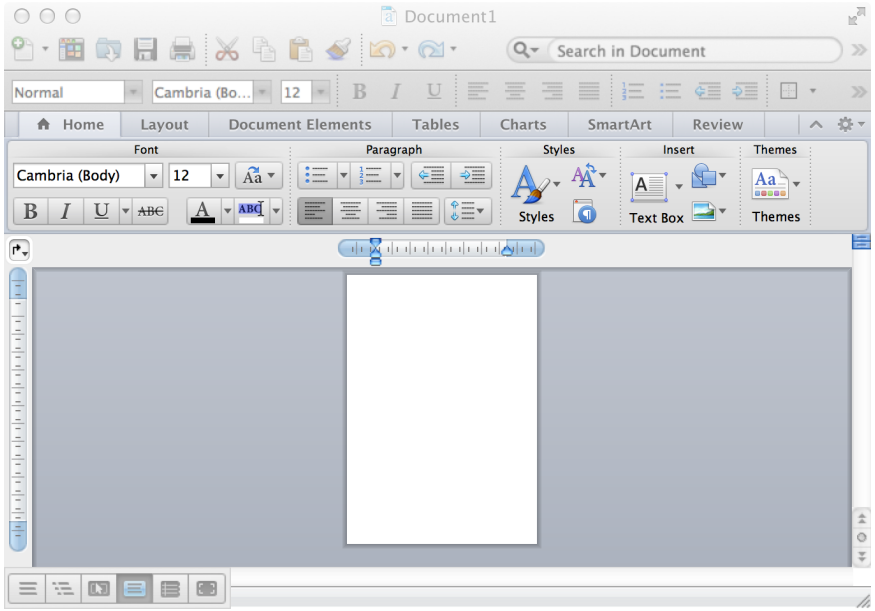


What happens when you double click on the icon of an application?



command (shell)





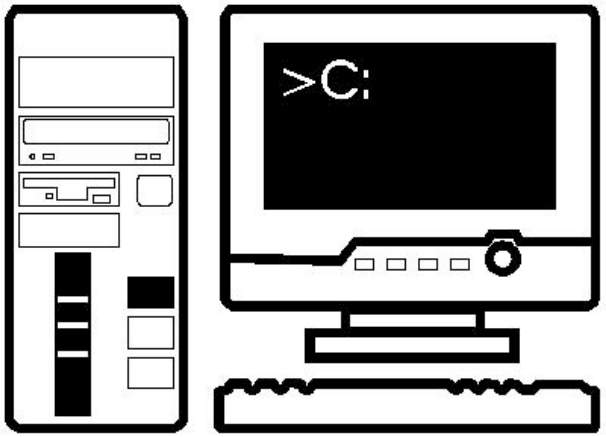
Open a command-line terminal on your computer

You can type a program name at the terminal prompt and then type **[Return]**

Terminal — csh — 68×21

```
~% python
```

command



Terminal — python2.7 — 68×21

```
~% python
Python 2.7.9 |Anaconda 2.2.0 (x86_64)| (default, Dec 15 2014, 10:37:
34)
[GCC 4.2.1 (Apple Inc. build 5577)] on darwin
Type "help", "copyright", "credits" or "license" for more informatio
n.
Anaconda is brought to you by Continuum Analytics.
Please check out: http://continuum.io/thanks and https://binstar.org
>>>
```

The command-line interface (terminal) allows you:

- to send typed instructions to the computer (i.e., run programs, move/view files, etc.)
- to see the output that results from those instructions.

Every time you type any Unix command and **press enter**, the computer will attempt to follow your instructions and then, when finished, return you to the **command prompt**.

Type the Unix command ‘ls’ at the command prompt

What happens?

How can you navigate the filesystem?

What do you need to be able to do to navigate the filesystem?

- Find out where you are in the filesystem
- Change directory
- Find your way home
- Identify the location of a file/directory
- Move one directory up

What is the ***path*** of a file or a directory?

Slashes separate parts of the directory path:

/home/PGC/Documents/Training/materials

What do you need to be able to do in order to do/manage stuff in the filesystem?

- Think of what you need in, e.g., Windows or Mac OSX

Example: Make a new directory

In groups

What do you need to be able to do in order to do/manage stuff in the filesystem?

- Think of what you need in, e.g., Windows or Mac OSX

    - Make a new directory
    - Remove a directory
    - Copy a file to another file
    - Rename a file/directory
    - Create a file
    - Open/close a file
    - Remove a file
    - Run programs

Writing and **running** scripts in Linux

Lets write a shell script

Open a text editor by typing nano

```
#!/bin/bash
echo “Hello world!”
echo “The current shell is:”
echo $SHELL
echo “My username is:”
echo $USER
```

Write the commands in a file, save and exit

#!/bin/bash

"Aha, you want to use the program located at /bin/bash to interpret all the instructions that follow"

How can we run programs on Linux?

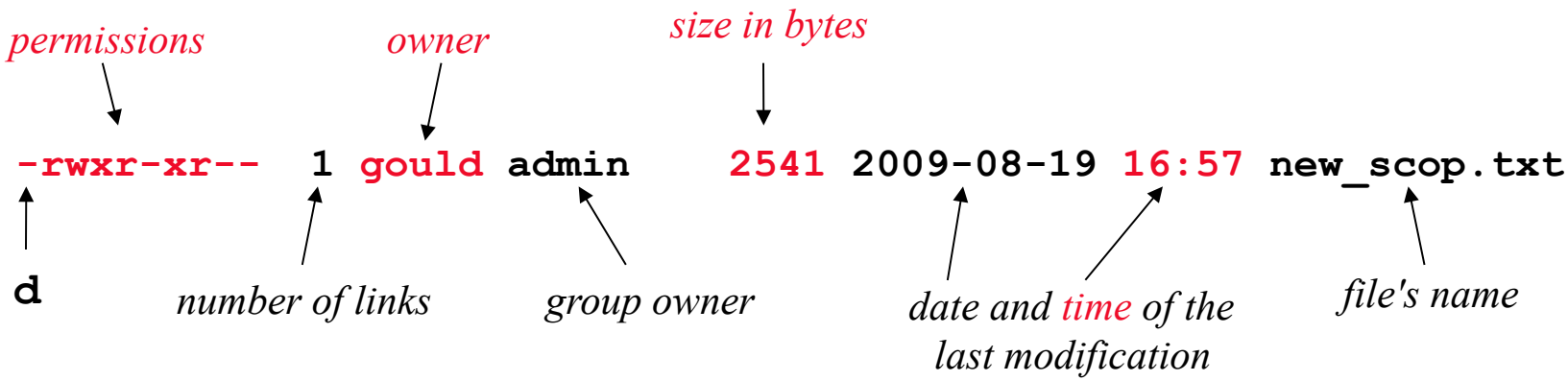
Prerequisites to run a program

1. The program must be somewhere on your computer
2. The program must be **executable**
3. You have to tell the **shell** which "**interpreter**" will read and execute the program AND where it will find it
4. You must be in the same directory as the program you want to run OR….
5. ….you can prefix its name with a path OR…
6. …the path to your program must in the **PATH environment variable**

Is my script executable?

File system security (access rights)

Each file (and directory) has associated access rights, which may be found by typing **`ls -l`**



Access rights on directories

r allows users to list files in the directory
w allows users to delete files from the directory or move files into it
x allow users to access files in the directory

How can I make my script executable?

Changing access rights: chmod

```
%chmod go-rwx myfile.txt
%chmod a+x my_script
```

Symbol	Meaning
`u`	user
`g`	group
`o`	other
`a`	all
`r`	read
`w`	write (and delete)
`x`	execute (and access directory)
`+`	add permission
`-`	take away permission

Now you want to execute the script

You have to tell Linux where it can find it

Where Linux searches for programs?

Where Linux searches for programs?

- You can always run a program by prefixing its name with a path:

  ~/Documents/[scriptname].sh

- Or you have to be in the same directory as the one of the program and type:

  ./[scriptname].sh

Where Linux searches for programs?

- If you simply type (at the prompt):

shell_commands.sh

Linux will check through **a list of predefined directories** to see if that program exists in any of those locations.

- If it finds a match, it will try to run the program and stop looking in any other directory.

- If it cannot find a match, it will print "command not found"

If the system returns a message saying "**command**: **Command not found**", this indicates that <u>either the command doesn't exist at all on the system or it is simply not in your path</u>.

- Any program in ~/my_scripts can be run from **anywhere** in the filesystem (as long as the program file is executable)
- You can use tab-completion
- Your scripts will be treated like any Linux command

```
# for shells in the bash family
export PATH=$PATH:~/[filename]
```

Non-interactive download of files from the Web

`wget [`*`option`*`]... [`*`URL`*`]...`

- Non-interactive means that it can work in the background, while the user is not logged on.
- This allows you to start a retrieval and disconnect from the system, letting Wget finish the work.
- By contrast, most of the Web browsers require constant user's presence, which can be a great hindrance when transferring a lot of data.

Listing files and directories

`ls`	list files and directories
`ls -a`	list all files and directories
`mkdir`	make a directory
`cd` *`directory`*	change to named directory
`cd`	change to home-directory
`cd ~`	change to home-directory
`cd ..`	change to parent directory
`pwd`	display the path of the current directory

The directories '.', '..', and '~'

```
% ls -a [Enter]

% cd . [Enter]

% cd .. [Enter]

% ls ~/oeiras
```

Handling files and directories

cp *file1 file2*	copy file1 and call it file2
mv *file1 file2*	move or rename file1 to file2
rm *file*	remove a file
rmdir *directory*	remove a directory
cat *file*	display a file
more *file*	display a file a page at a time
head *file*	display the first few lines of a file
tail *file*	display the last few lines of a file
grep *'keyword' file*	search a file for keywords
wc *file*	count number of lines/words/characters in file

more **less** **clear**

Redirection

command > file	redirect standard output to a file
command >> file	append standard output to a file
command < file	redirect standard input from a file
cat *file1 file2 > file0*	concatenate file1 and file2 to file0
sort	sort data
who	list users currently logged in