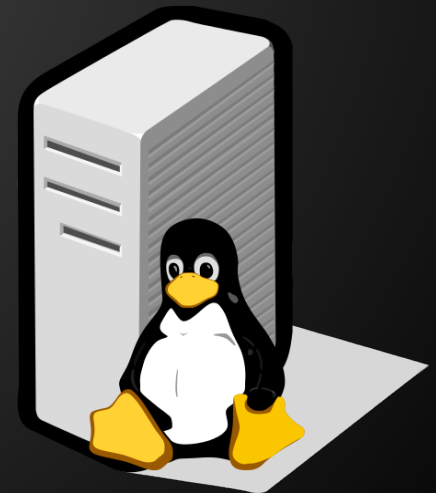


Intro to Linux

BecA Advanced Bioinformatics Workshop

Alan Orth
Nairobi, Kenya
September, 2015



What is “Linux” anyways?

- Linux is an operating system — like Windows or Mac OS X
- Created in 1991 by Linus Torvalds in Finland
- Microsoft’s DOS was too limiting
- UNIX was expensive and restrictive
- Linux was born!

Why Linux for Bioinformatics?

- Bioinformatics is the *application of information technology and computer science to the field of molecular biology*
- Data sets are getting bigger and we need more processing power — computers with that kind of power run Linux :)
- Linux is efficient, stable, and has excellent tools for text processing

Getting your feet wet

Use an *SSH* client like putty or MobaXterm to connect to our Linux server from Windows.

Server: `hpc.ilri.cgiar.org`

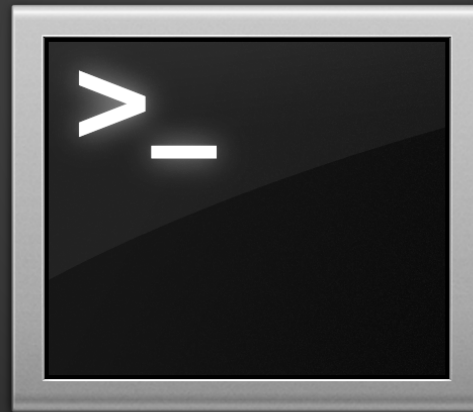
Username: `user1`

Password: `user1`



Getting familiar with the shell

Linux has a graphical environment like Windows, but the real power lies in its command line interface (CLI) or “shell”.



It's very simple: you type a command and press *Enter* to run it.

Command structure

Linux commands come in various forms. Some are simple and can be used by themselves, for example:

whoami

cal

ls

date

Command structure

... other times you can add “arguments” to change the behavior of the command.

Arguments are separated by one or more spaces, for example:

```
cal 09 2015
```

Some commands *require* arguments because they don't make sense to run by themselves.

Commands and their arguments

Examples of commands and their arguments...

`ls -la` (“long” list of files)
`ls -l .bashrc` (“long” list of .bashrc)
`mv file1 file2` (rename file1 to file2)
`cp file filecopy` (copy file to filecopy)
`rm file` (delete file)
`mkdir data` (create folder called data)

Common pitfalls with Linux commands

- Case sensitive (Ls vs ls)
- Attention to detail (ls --l vs ls -l)
- Pasting from Word (“curly” quotes, etc!)
- Missing spaces
- Using Windows-isms (\ vs /)

Google is your friend!

Linux file system hierarchy

Files and folders are organized in a *hierarchical* fashion similar to an upside-down tree, so we call the top of a directory structure the “root”.

The “root” of your home folder, for example:

```
/home/user1
```

Navigating the file system

A few commands to help you navigate and manipulate the file system's directory structure:

`pwd` — *print working directory* (“where am I?”)

`ls` — list contents of the current directory

`cd` — *change directory*

`mkdir` — *make directory*

FYI: “directory” is just a fancy name for folder...

Navigating the file system

Create some directories and get the hang of moving around them:

```
mkdir one  
mkdir two  
mkdir two/three  
cd one
```

How do we get to two?

Navigating the file system

If we want to move to the directory “two” we have to first move back up in the directory hierarchy. Once we move back to “user1” we will be able to move into “two.”

```
cd ..
```

```
cd two
```

Special directories

- . “here”
- .. parent directory, aka “up one”
- ~ home, aka /home/user1

Creating and editing text files

Editing text in the Linux CLI is trickier than you think, but it's still easy...

Move to the root of your home directory and use the command `nano` to create a new file called "hello":

```
cd ~  
nano hello
```

Type a simple message and then type `Ctrl-O` to save the file, and `Ctrl-X` to quit. BTW: in Linux "`^`" means `Ctrl`.

Working with text files

Check the contents of your new text file:

```
wc -l hello  
cat hello  
cp hello hello2  
less hello
```

Press “q” when you're done to quit less.

cat simply prints a file to the screen, while less is used to interactively view a text file one page at a time. Programs like less are called “pagers.”

Anything else?

Questions?