Linux Workshop

S. T. Balan, L. Whiteway, T. Wilson, A. Palmese

Department of Physics and Astronomy
University College London

15 October 2015

Where to find this presentation

url

https://github.com/Astrophysics-UCL/HPCInfo/blob/master/training/workshops_2015/linux_workshop/slides/linux_workshop_oct_2015.pdf

What will you learn?

- In this talk:
  - Accessing Astrophysics group machines
  - Using the Linux console for your research
- In the next talk:
  - How to run programs on High Performance Computing (HPC) machines

Information on the Web

Astrophysics Wiki

`https://wiki.ucl.ac.uk/display/PhysAstAstPhysGrp/Main+Page`

This Wiki is freely viewable and editable by all members of the department. Please use it to record information that you think will be useful to others (including your future self). Be bold!

UCL Research Computing Platforms

`https://wiki.rc.ucl.ac.uk/wiki/Main_Page`

Stack Overflow

`http://stackoverflow.com/`

Command shell

- You will be using a 'command shell'.
- This is a text-based environment in which you type commands and text output.
- Not GUI! Reflects the hardware limitations current when Unix was created. Low-tech and reliable e.g. for remote access.
- Various command shell programs in use: *bash*, *csh*, *tcsh*,...

Accessing machines remotely

You will need a *username* and *password*

```
# step 1: login to zuserver1
ssh -YC username@zuserver1.star.ucl.ac.uk

# step 2: login to splinter (or other machines)
# from zuserver1
ssh -YC username@splinter-login.star.ucl.ac.uk
```

Directory structure

- Everything is organised around files (which may be data files or program files i.e. instructions to be executed).
- Files live in directories. There is a hierarchical tree structure of directories.
- Sample file name: /share/splinter/ucapwhi/des/foo.txt
- Note use of slash '/', not backslash '\' as in Windows.
- Case sensitivity: 'Foo' and 'foo' are different strings.

Special symbols for directories

Symbol	Meaning
/	Top of the directory tree (the root directory)
.	Current directory
..	Parent of the current directory
~	User's 'home' directory

Structure of commands

Structure

```
# [command] -[option[s]] [argument]
```

Example

```
ls -la
mkdir hello_world
cp hello.cpp new_hello.cpp
```

Linux console cheat sheat I

navigation and help

```
ls -lah dir_name
cd dir_name
cd ..
cd -
man command_name
pwd
exit
```

copy or move

```
cp src dest
cp -r src dest
mv src dest
ln -s src targ
```

create or delete

```
touch file.txt
mkdir dir_name
mkdir -p prt/dir
rm -i file.txt
rm -rf dir_name
```

find or search

```
locate file
whereis file
grep "foo" file
awk 'pattern' file
```

Linux console cheat sheat II

file contents

```
cat file
more file
less file
head file
tail file
nm object_file
readelf shared_obj_file
ldd executable
```

process management

```
ps -e
kill
killall
top
```

ssh

```
ssh usr@host
ssh -YC user@host
scp usr@host:file dest
```

system info

```
uname -a
who
whoami
whois
which
finger
ping
echo $VAR_NAME
```

Linux console cheat sheat III

& ; | ¿ ¡

```
&  # background
;  # combine
\  # next line
|  # combine
*  # wildcard
>  # output
<  # input
```

Text editors

```
emacs
vi
gedit
```

web

```
firefox
google-chrome
wget
curl
```

publishing

```
latex
pdflatex
bibtex
```

Linux console cheat sheat IV

compressed files

```
gzip
gunzip
tar xvzf
tar cvzf
tar xvjf
tar xvJf
```

images

```
eog
xfig
gimp
gthumb
convert
```

development

```
make
cmake
python
gcc
g++
gfortran
```

scientific

```
gnuplot
R
matlab
IDL
```

Exercises I

1. Go to your home directory and create a directory called `linux_hpc_workshop`.
2. Change directory to `linux_hpc_workshop`.
3. Find the name of the present working directory.
4. Make a directory `level_1/level_2`, and move to `level_1/level_2` in one command.
5. Move back to the previous directory.
6. Remove the directory `level_1` (and its contents).
7. In the current directory make a symbolic link to `usr/lib` called `my_sybolic_link`.
8. Create a file called `foo.txt` with contents "This file contains the word foo".
9. Add another line in `foo.txt` with contents "This is the second line".
10. Check to see if it worked.
11. Search for the phrase *foo* in `foo.txt`.

Exercises II

1. Find the location of your python installation.
2. Find the installation location(s) of `liblapack.a`.
3. Find whether an object `daxpy` is in `liblapack.a`.
4. Find the value the environment variable `PATH` and `LD_LIBRARY_PATH`.
5. Set the environment variable `MY_LINUX_HPC_VAR` to equal the absolute path to `linux_hpc_workshop`.
6. Add (i.e append) to the `PATH` the absolute path to `linux_hpc_workshop`.
7. Use the *source* command do the last two steps from source file.
8. Use the *man* command to find the option of `ls` that shows the output in Kilobyte,Megabyte.

Exercises III

1. Find hostname, processor type and operating system version and write this info into a text file called `info.txt`.
2. List the people who are currently logged into the system.
3. Find the process that is taking most of the CPU at the moment.
4. Find the IDs of the processes that you are running.
5. Make a directory called `to_be_compressed`. Add the files `hello.cpp` and `hello.py` in this dir. Then compress this directory using *tar* and *zip*.
6. Delete the directory `to_be_compressed` and extract the files from `to_be_compressed.tar.gz`.
7. Use *wget* to download files from `ftp://heasarc.gsfc.nasa.gov/software/fitsio/c/cfitsio3370.tar.gz`.
8. Find the size of the item you just downloaded in MB.
9. Extract all files from this downloaded archive file.
10. In the extracted files, find the number of occurrences of the word `table` in all the files with extension `.h`.
11. Remove all the files with extension `.h`.
12. Copy the files with extension `.c` into a new directory `c_files`.