

# Introduction to Linux

**dr Karina Kubiak – Ossowska**  
**High Performance Computing Support Team**  
**Department of Physics**

**e-mail: [karina.kubiak@strath.ac.uk](mailto:karina.kubiak@strath.ac.uk)**

**UNIX** is an operating system which was first developed in the 1960s, and has been under constant development ever since. By operating system, we mean the suite of programs which make the computer work. It is a stable, multi-user, multi-tasking system for servers, desktops and laptops.

UNIX systems also have a graphical user interface (GUI) similar to Microsoft Windows which provides an easy to use environment. However, knowledge of UNIX is required for operations which aren't covered by a graphical program, or for when there is no windows interface available, for example, in a telnet session.

There are many different versions of UNIX, although they share common similarities. The most popular varieties of UNIX are Sun Solaris, GNU/Linux, and MacOS X.

The shell is a command line interpreter (CLI). It interprets the commands the user types in and arranges for them to be carried out. The commands are themselves programs: when they terminate, the shell gives the user another prompt

Filename Completion - By typing part of the name of a command, filename or directory and pressing the [**Tab**] key, the tcsh shell will complete the rest of the name automatically. If the shell finds more than one name beginning with those letters you have typed, it will beep, prompting you to type a few more letters before pressing the tab key again. To see previous commands press [**up arrow**].

## Usefull links:

<http://www.ee.surrey.ac.uk/Teaching/Unix/unix8.html>

<http://linuxfordummies.org/8-free-linux-manualspdf-format/>

## cd (change directory)

cd .. : changes to the parent directory (..)  
cd ../../ : goes up two directory levels  
cd ../../job-scripts : goes up two directory levels then into a sub-directory  
.. : parent directory  
. : current directory

## pwd (print working directory)

pwd : prints the working directory path

## whoami

whoami : prints the username

## ls (list)

ls : on its own lists current directory  
ls ~ : lists the home directory  
ls /lustre/strath/physics/cwb08102 : lists /lustre/strath/physics/cwb08102 directory  
ls -a : lists hidden files and folders  
ls -l : long listing  
ll : as above  
ls -al : long listing with hidden files

## mkdir (make directory)

mkdir *dir1* : creates a new (empty) directory

## rm (remove), rmdir (remove directory)

rmdir *dir1* : deletes an empty directory  
rm *file1* : removes (deletes) a file  
rm *file1 file2 file3* : removes a list of files  
rm *file\** : achieves the same as the previous command  
\* is a “wild-card”. The shell ‘expands’ the wild-card and returns all filenames beginning with ‘file’  
rm -rf *dir1* : deletes a directory and all of its contents  
-f (force): ask no questions!

## cp (copy)

`cp file1 file2` : creates a copy of *file1* and calls it *file2*  
`cp -r dir1 dir2` : create a copy of *dir1* with all of its contents  
`cp /lustre/strath/physics/cwb08102/file1 .` : copies a file *file1* from another directory to the current directory (denoted by `.`)

## mv (move)

`mv file1 file2` : rename *file1* to *file2*  
`mv file1 ..` : moves *file1* to the parent directory (`..`)  
`mv file1 ~` : moves *file1* to the home directory  
`mv dir1 dir2` : renames *dir1* ( `-r` not necessary)  
`mv dir1 ~` : moves *dir1* to home directory  
`mv dir1 /lustre/strath/physics/cwb08102` : moves *dir1* to given *path*

## more / less

These commands writes the contents of a file onto the screen a page at a time.

`more file1`

`less file1`

Press the [**spacebar**] if you want to see another page, type [**q**] if you want to quit reading, type [**b**] if you want to go backwards, type [**f**] if ypu want to go forwards, type [**/**] if you want to search for a term

`head file1` : views the first 10 lines of the file

`tail file2` : views the last 10 lines of the file

## Viewing running processes

`top` : shows the processes (programs) running on the computer  
press [**spacebar**] to update, press [**q**] to quit  
`ps` : produces a static list of your running processes  
`ps -f` : produces a “long” listing  
`ps -ef` : lists all running processes (long format)

## Redirection

`ps -ef > processes.txt` : redirects output from the terminal to a file, `>` creates the named file *processes.txt*  
`ps -ef >> processes.txt` : appends to an existing file  
`ps -ef >& processes.txt` : redirects system errors as well as “standard” output:  
`cat file1.txt file2.txt > bigfile.txt` : concatenates (merges) two files into one

## Searching a File

`grep error output.txt` : find instances of error in a file `output.txt`  
`grep -i error output.txt` : case insensitive search  
`grep error output.txt > error.txt` : will print to the file `error.txt` all lines containing the phrase “error” in the file `output.txt`

## Command Output Re-direction

`ps -ef | grep cwb08102` : find instances of error in “output.txt”  
| re-directs the output of the command “`ps -ef`” and sends it to the second command `grep`

## Kill a process

`top` : will show the process id  
`kill -9 processid` : will kill the process with given ID

It is not possible to kill other users processes (only root can do this)

## Access rights on files

- **r** (or -), indicates read permission (or otherwise)
- **w** (or -), indicates write permission (or otherwise)
- **x** (or -), indicates execution permission (or otherwise)

## Access rights on directories

- **r** allows users to list files in the directory;
- **w** means that users may delete files from the directory or move files into it;
- **x** means the right to access files in the directory. This implies that you may read files in the directory provided you have read permission on the individual files.

So, in order to read a file, you must have execute permission on the directory containing that file, and hence on any directory containing that directory as a subdirectory, and so on, up the tree.

`-rwxrwxrwx` a file that everyone can read, write and execute (and delete)  
`-rw-----` a file that only a user (owner) can read and write

## chmod (changing a file mode)

Only the owner of a file can use chmod to change the permissions of a file. The options of chmod are as follows

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

`chmod go-rwx file1` : will remove read write and execute permissions on the *file1* for the group and others

`chmod a+rw file1` : will give read and write permission on the *file1* to all

## quota

All users are allocated a certain amount of disk space on the file system for their personal files

`quota -v` : will check the current quota and how much has been used

## df

The df command reports on the space left on the file system. For example, to find out how much space is left on the fileserver, type

`df` :will print on the screen how much space is left

## du

The du command outputs the number of kilobytes used by each subdirectory.

`du` :will display sizes of all files

`du -s *` :will display only a summary (total size) for all files and directories

## gzip (file compression)

<code>gzip file1.txt</code>	: will compress <i>file1.txt</i> to <i>file1.txt.gz</i>
<code>gunzip file1.txt.gz</code>	: will uncompress <i>file1.txt.gz</i> to <i>file1.txt</i>
<code>zcat file1.txt.gz</code>	: reads gzipped files without needing to uncompress them first
<code>zcat file1.txt.gz   less</code>	: pipe the output through less
<code>file file1</code>	: classifies the named files according to the type of data they contain, for example ASCII (text), pictures, compressed data ...
<code>file *</code>	: will report on all files in current directory

## diff (difference)

This command compares the contents of two files and displays the differences.

<code>diff file1 file2</code>	: will show the differences between <i>file1</i> and <i>file2</i>
-------------------------------	---

## find

This command searches through the directories for files and directories with a given name, date, size, or any other attribute you care to specify. To see all options available type `man find`.

<code>find . -name "*.txt" -print</code>	: will find in the current directory and all subdirectories all files with the extension .txt and print the results on the screen
<code>find . -size +1M -ls</code>	: will find files bigger than 1MB and display the results as a short listing

## history

The C shell keeps an ordered list of all the commands that have been entered. Each command is given a number according to the order it was entered.

<code>history</code>	: will show command history list
----------------------	----------------------------------

## man pages

<code>man command_name</code>	: help how to use the command (navigation like under <b>more:</b> [spacebar] to page down, [b] to page up, [q] to quit)
e.g. <code>man grep</code>	

## Text Editors:

A text editor is a program for inserting or amending text in a file. A text editor is not a word processor although some text editors do include word processing facilities.

### vi editor – for advanced users

`vi file1.txt` : will open new file `file1.txt` under vi editor

there are two modes in which you use vi:

**Command mode:** this is the mode you are in whenever you begin to use **vi**. In this mode commands are used to move around and edit text objects such as words, sentences and paragraphs. Pressing the [**esc**] key returns you to command mode.

**Insert mode:** this is the mode you use to type (insert) text into a buffer. There are several commands that you can use to enter this mode:

<code>a</code>	: insert text after the cursor
<code>i</code>	: insert text before cursor
<code>A</code>	: append text at the end of the current line
<code>I</code>	: insert text at the start of the current line
<code>o</code>	: open a new line below the current line
<code>O</code>	: open a new line above the current line

Once the insert mode is on the user can type normally. To come back to the command mode press [**esc**].

The most important commands:

<code>:w</code>	: write (“:” has to be at the beginning)
<code>:wq</code>	: write and quit
<code>:q</code>	: quit
<code>:q!</code>	: quit with no changes !
<code>:\$</code>	: go to the end of the file
<code>:100</code>	: go to line 100
<code>yy</code>	: copy the current line
<code>P</code>	: insert the copied line
<code>r</code>	: replace the current character by other. Example: <code>r4</code> will replace the current sign to 4
<code>:r file2.txt</code>	: reads the given file name to current file
<code>x</code>	: delete at the current cursor position
<code>9x</code>	: delete 9 signs starting from the current cursor position

`:set number` : will print the line number  
`:%s/text1/text2/g` : will substitute text1 to text2 in the entire file (global)

## gedit editor

`gedit file1.txt` : will create *file1.txt* and open notepad-like window

`gedit file1.txt &` : will work as above and keep the terminal active

gedit can be open also from Applications Menu

## Remote control

### ssh (secure shell)

ssh is a program for logging into a remote machine and executing commands in the remote machine.

ssh connects and logs into the specified hostname. The user's identity is proved to the remote machine using a password.

`ssh headnode1.eng.strath.ac.uk -l cwb08102` :log user cwb08102 to headnode1.eng.strath.ac.uk

`ssh -X headnode1.eng.strath.ac.uk -l cwb08102` : use X terminal

### scp (secure copy)

scp copies files over the network securely; uses ssh for data transfer, using the same authentication and providing the same security as ssh.

`scp [-p] [-v] [-r] username@host1:/path/file username@host2:/path/file_or_dir`

`scp file1.dat cwb08102@headnode1.eng.strath.ac.uk:/lustre/chemeng/cwb08102/_CPP_PEPTIDE/`

copy file *file1.dat* from current computer and localisation to the directory  
`/lustre/chemeng/cwb08102/_CPP_PEPTIDE/` on headnode1.eng.strath.ac.uk with the same name

`scp file1.dat`

`cwb08102@headnode1.eng.strath.ac.uk:/lustre/chemeng/cwb08102/_CPP_PEPTIDE/file2.dat`

copy file *file1.dat* from current computer and localization to the directory  
`/lustre/chemeng/cwb08102/_CPP_PEPTIDE/` on headnode1.eng.strath.ac.uk with new name *file2.dat*

scp malpka@ce-kk-pc.chem.strath.ac.uk:/media/old\_home/malpka/PHP\_WATER/file1.dat .

Will take the file *file1.dat* from computer named ce-kk-pc.chem.strath.ac.uk to current localisation without changing the name (.).

scp malpka@rc-weir40.chem.strath.ac.uk:/home/malpka/Documents/file1.dat  
malpka@headnode1.eng.strath.ac.uk:/lustre/chemeng/cwb08102/PHP\_PEPTIDE/

will copy file *file1.dat* from computer weir40.... to headnode1....

scp  
cwb08102@headnode1.eng.strath.ac.uk:/lustre/chemeng/cwb08102/PHP\_PEPTIDE/PHP\_WATER/  
LYS\* .

will take all files LYS\* from headnode1 to current localisation without changing names

-r : copy the entire directory together with subdirectories.  
-p : preserve file attributes and timestamps.  
-v : will make scp verbose.

## sftp (secure file transfer program)

**sftp** is a program for transferring files to and from a remote computer. It is interactive, just like **ftp** used to be, but uses the secure encryption of ssh.

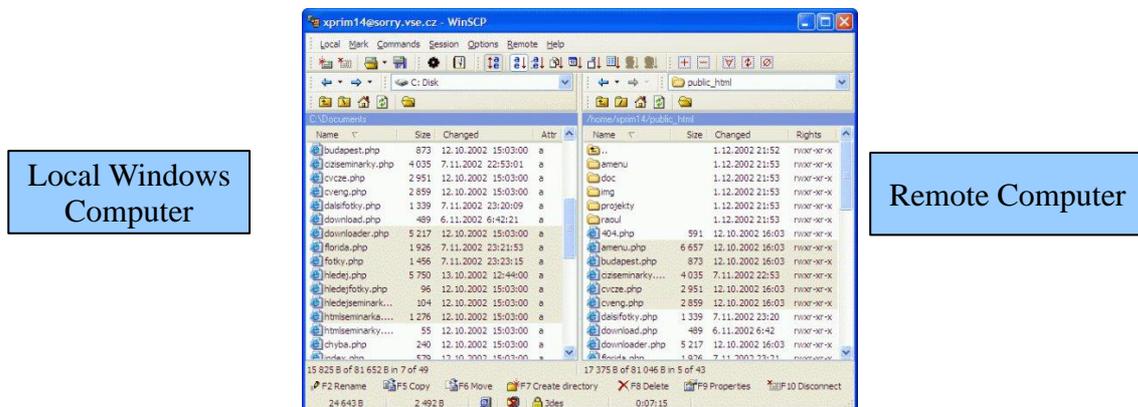
Once you connect with the remote computer, giving a password if necessary, you can type the following interactive commands to change directories and to transfer files between your local computer and the remote computer:

*get filename* : retrieves remote file and stores it to local computer  
*put filename* : upload local file to store on remote computer  
*cd path* : change remote directory to *path*  
*ls* : list remote files  
*pwd* : print remote working directory  
*rename oldfilename newfilename* : rename remote file  
*rm file1* : remove remote file  
*mkdir path* : create remote directory  
*rmdir dirname* : remove remote directory  
*lcd path* : change local directory to *path*  
*lpwd* : print local working directory  
*lls* : list local files

**help** : display help text  
**quit** : quit sftp

## WinSCP (scp for Windows)

WinSCP is a SFTP client and FTP client for Windows. Its main function is the secure file transfer between a local and a remote computer. It uses Secure Shell (SSH) and supports, in addition to Secure FTP, also legacy SCP protocol



Application interface is similar to Norton Commander

## PuTTY

PuTTY is a free implementation (open source) of Telnet and SSH for Windows and Unix platforms, along with an xterm terminal emulator. To download visit <http://www.putty.org/>

## Remote desktop

The best program to have remote desktop connection between computers is NX. It is open source, works for Windows, Mac OS and Linux desktop computer and connect them to remote Linux computer. NX server runs on ARCHIE-WeSt. Users can download the client from [www.nomachine.com](http://www.nomachine.com) . Instructions: <http://www.archie-west.ac.uk/archie-access> To have the remote desktop session please connect directly to **archie-w, archie-e, archie-s or archie-t**. Please do not leave the session, **please log out!**

# Summary

## 1. Home Directory & Changing Directories

<code>~</code>	: represents “Home Directory”
<code>cd</code>	: change directory, e.g.
<code>cd job-scripts</code>	
<code>cd /lustre/strath/physics/cwb08102</code>	
<code>cd ~</code>	
<code>cd ~/job-scripts</code>	
<code>cd</code>	: with no argument takes the user to their home directory
<code>cd ..</code>	: changes to the parent directory
<code>cd ../../</code>	: goes up two directory levels
<code>cd ../../job-scripts</code>	: goes up two directory levels then into a sub-directory
<code>pwd</code>	: lists the present working directory
<code>whoami</code>	: returns the username

## 2. Listing Files & Directories

<code>ls</code>	: on its own lists current directory
<code>ls ~</code>	: lists the home directory
<code>ls /lustre/strath/physics/cwb08102</code>	: lists /lustre/strath/physics/cwb08102 directory
<code>ls -a</code>	: lists hidden files and folders
<code>ls -l</code>	: long listing
<code>ll</code>	: as above
<code>ls -al</code>	: long listing with hidden files

## 4. The “Shell”

`~/.bashrc` : used to execute commands upon login

`~/.bash_profile` : can be used to set environment variable

Filename completion:

- Press [**Tab**] key to complete filenames as you type
- [**Up Arrow**] allows you to access command history

## 5. Creating & deleting directories/files

`mkdir dir1` : creates a new (empty) directory

`rmdir dir1` : deletes an empty directory

`rm file1` : removes (deletes) a file

`rm file1 file2 file3` : removes a list of files

`rm file*` : achieves the same as the previous command  
\* is a “wildcard”. The shell ‘expands’ the wildcard and returns all filenames beginning with ‘file’

`rm -rf dir1` : deletes a directory and all of its contents  
-f: ask no questions!

## 6. Copying files & directories

`cp file1 file2` : creates a copy of *file1* and calls it *file2*

`cp -r dir1 dir2` : create a copy of *dir1* with all of its contents

`cp /lustre/users/acs03114/file1 .` : copies a file *file1* from another directory to the current directory (denoted by . )

## 7. Moving files and directories

`mv file1 file2` : rename *file1* to *file2*

`mv file1 ..` : moves *file1* to the parent directory

`mv file1 ~` : moves *file1* to the home directory

`mv dir1 dir2` : renames *dir1* ( -r not necessary)

`mv dir1 ~` : moves *dir1* to home directory

`mv dir1 /lustre/strath/physics/cwb08102` : moves *dir1* to given *path*

## 8. Viewing files in the terminal

`more file1`

- [**spacebar**] advances through file
- [**b**] for going backwards
- [**q**] to quit
- [**/**] to search for a term

`head file1` : views the first 10 lines of the *file1*

`tail file1` : views the last 10 lines of the *file1*

## 9. Viewing running processes

`top` : shows the processes (programs) running on the computer

- press [**spacebar**] to update
- press [**q**] to quit

`ps` : produces a static list of your running processes

`ps -f` : produces a “long” listing

`ps -ef` : lists all running processes (long format)

## 10. Redirection

`ps -ef > processes.txt` : redirects output from the terminal to a file  
: > creates the named file *processes.txt*

`ps -ef >> processes.txt` : appends to an existing file

`ps -ef >& processes.txt` : redirects system errors as well as “standard” output:

`cat file1.txt file2.txt > bigfile.txt` : Concatenates (merges) two files into one

## 11. Searching a File

`grep error output.txt` : find instances of error in *output.txt*  
`grep -i error output.txt` : case insensitive search

## 12. Command Output Re-direction

`ps -ef | grep cwb08102` : find instances of error in “output.txt”  
| re-directs the output of the command “`ps -ef`” and sends it to the second command `grep`

## 13. Kill a process

`top` : will show the process id  
`kill -9 processid` : will kill the process with given ID

## 14. Changing permissions

`chmod go-rwx file1` : will remove read write and execute permissions on the *file1* for the group and others  
`chmod a+rw file1` : will give read and write permission on the *file1* to all

## 15. quota

`quota -v` : will check the current quota and how much has been used

## 16. df & du

`df` :will print on the screen how much space is left  
`du` :will display sizes of all files  
`du -s *` :will display only a summary (total size) for all files and directories

## 17. File compression

`gzip file1.txt` : will compress *file1.txt* to *file1.txt.gz*  
`gunzip file1.txt.gz` : will uncompress *file1.txt.gz* to *file1.txt*  
`zcat file1.txt.gz` :reads gzipped files without needing to uncompress them first  
`zcat file1.txt.gz | less` : pipe the output though less

## 18. File information

<code>file file1</code>	: classifies the named files according to the type of data they contain, for example ASCII (text), pictures, compressed data ...
<code>file *</code>	: will report on all files in current directory
<code>diff file1 file2</code>	: will show the differences between <i>file1</i> and <i>file2</i>
<code>find . -name "*.txt" -print</code>	: will find in the current directory and all subdirectories all files with the extension .txt and print the results on the screen
<code>find . -size +1M -ls</code>	: will find files bigger than 1MB and display the results as a short listing

## 19. History and help

<code>history</code>	:will show command history list
<code>man command_name</code>	: help how to use the command (navigation like under <b>more:</b> [spacebar] to page down, [b] to page up, [q] to quit)

## 20. gedit editor

<code>gedit file1.txt</code>	: will create <i>file1.txt</i> and open notepad-like window
<code>gedit file1.txt &amp;</code>	: will work as above and keep the terminal active

## 21. ssh, scp

`ssh headnode1.eng.strath.ac.uk -l cwb08102` :log user cwb08102 to headnode1.eng.strath.ac.uk  
`ssh -X headnode1.eng.strath.ac.uk -l cwb08102` : use X terminal

`scp [-p] [-v] [-r] username@host1:/path/file username@host2:/path/file_or_dir`

`scp file1.dat cwb08102@headnode1.eng.strath.ac.uk:/lustre/chemeng/cwb08102/PHP_PEPTIDE/`  
copy file *file1.dat* from current computer and localisation to the directory  
`/lustre/chemeng/cwb08102/PHP_PEPTIDE/` on headnode1.eng.strath.ac.uk with the same name

scp *file1.dat*

cwb08102@headnode1.eng.strath.ac.uk:/lustre/chemeng/cwb08102/\_CPP\_PEPTIDE/*file2.dat*

copy file *file1.dat* from current computer and localization to the directory  
/lustre/chemeng/cwb08102/\_CPP\_PEPTIDE/ on headnode1.eng.strath.ac.uk with new name *file2.dat*

scp malpka@ce-kk-pc.chem.strath.ac.uk:/media/old\_home/malpka/ CPP\_WATER/*file1.dat* .

Will take the file *file1.dat* from computer named ce-kk-pc.chem.strath.ac.uk to current localisation  
without changing the name (.).

scp malpka@rc-weir40.chem.strath.ac.uk:/home/malpka/Documents/*file1.dat*

malpka@headnode1.eng.strath.ac.uk:/lustre/chemeng/cwb08102/ CPP\_PEPTIDE/

will copy file *file1.dat* from computer weir40.... to headnode1....

scp

cwb08102@headnode1.eng.strath.ac.uk:/lustre/chemeng/cwb08102/ CPP\_PEPTIDE/ CPP\_WATER/  
LYS\* .

will take all files LYS\* from headnode1 to current localisation without changing names

- r : copy the entire directory together with subdirectories.
- p : preserve file attributes and timestamps.
- v : will make scp verbose.