BASH cheat sheet - Level 1

File and directory handling commands

cd *dir* change directory to *dir*
cd – Go back to the previous directory
cd .. Go to the parent directory
cd Go to the user's home directory (same as **cd ~** or **cd /home/*user*/**)

pwd Show the current durectory

cp *file1 file2* Copy *file1* to *file2.* Use **–r** option for copying a directory.

scp *source_file target_ file*
Secure **c**o**p**y over a network connection. A remote ***source*** or/and ***target*** must follow that syntax: ***user@IP*:/home/*user/path*.** Use **–r** option for copying a directory.

rsync –L *source_file target_ file*
Remote file copy. Transfer just the differences between two sets of files. **–L** option transform symlink into referent file/dir.

mv *file1 file2* Rename or move *file1* to *file2* if *file2* is an existing directory, moves *file1* into directory *file2*.
mkdir *dir* Create a directory *dir.*

rm *file* Delete *file*. Use **–r** option for a directory.
ln –s *file link* Create symbolic link *link* to *file.*

unlink *link* Remove a link.

df Show disk space usage (byte). To symplify the lecture use the –h option (human readable).
du –h *dir* Show directory space usage.

ls *dir* Directory listing. Use the **–l** option to list in **l**ong format.

chmod *who=permission file*
Change the *who* permission of *file* to *permission. Who* can be **u**,**g** or/and **o** (**u**ser, **g**roup or/and **o**ther) and *permission* can be **-**,**x**,**w** or/and **r**.
chmod *OctalOctalOctal file*
Change the user, group and other permissions as defined by each octal.

- 4 read (r)
- 2 write (w)
- 1 execute (x)
- 0 none (-)

chmod 754 *file*
Read,write,execute for user, read and execute for group and read only for other.
chmod ug=rwx *file*
Read, write, execute for user and group.

tar czf *file.tar.gz dir* Create a tar with Gzip **c**ompression of *dir*.
tar xzf *file.tar.gz* E**x**tract a tar using Gzip.

gzip *file* Compress file and rename it to *file.gz.* Use the **–d** option to **d**ecompress *file.gz* back to *file.*

File information handling commands

wc –l *file* Display the number of **l**ines *file*. (**-w** for the number of **w**ords, **-c** for the number of **c**haracters). **wc** without option gives these three information.

cat *file* Display the content of file(s).

more *file* Display the content of file one screen at a time.
less *file* Similar as more but allows backward and forward movement.

head –*N file*
Display the ***N*** first lines of *file(s).* Without any option 10 first lines are displayed.

tail –*N file*
Display the ***N*** last lines of *file(s).* Without any option the 10 last lines are displayed.

command* > *file
Redirect *command* output into a *file*. If the file exists, it is overwritten.

command* >> *file
Redirect *command* output and append into *file*.

cut –d';' -f1,4 *file*
Cut out selected portions of each line of a file. The **–d** option defines the field delimiter character (here **;**) instead of the default tab character. The **–f** option specifies the field(s) displayed (here the first and the fourth).

sort *file*
Sort lines of text *file* alphabetically. Use **–n** option to take into account the numbers (ascending order). Use **-k *N*** option to specified to sort according to the field *N*.

uniq *file*
Filter out consecutive repeated lines in file. **–u** displays only not repeated lines while **-d** displays only repeated ones.

split –l*N file prefix*
Split *file* into pieces of **N l**ines labelled by ***prefix***.
split –b Nm *file*
Split *file* into pieces of **N m**egabytes. Use **k** instead of **m** for **k**ylobyte and none for byte.

grep "*pattern*" *file*
Print lines of *file* that match the *pattern.* Use the **–E** option to interpret the *pattern* as a regular expression (see Level 2 for regular expression).

man *command* : display the *command*'s manual page

Jacques Dainat - 2015

fold –w 60 *file* Break the lines to have a maximum of 60 columns width (80 by default).

Miscellaneous

history Display the history list with line numbers.

which *command* Locate a command.

ssh *user@IP* Log into a remote machine.

find *path* -name *name**

Find in *path* and sub-directory recurcively the files or directories matching *name*.

rev *file* Reverse the order of characters in every line of *file*.

join *file1 file2*

Join specified files based on *join-field* and writes the result to the standard output. By default *join* works on the first field.

join -1 2 -2 2 –a1 –a2 –o *list*–e "*value*" *file1 file2*

-1 2 and **-2 2** allows to respectively join on the second field of *file1* and the second field of *file2*.

-a1 and **–a2** option allows respectively to also include the non matching records from *file1* and *file2*. **-o *list*** specify the fields that will be output. e.g the list **0,1.1,2.1** will display respectively the join field, the field1 of *file1* and the field1 of *file2*. When **–o** option is used, the **–e '*value*'** option replace any missing data fields by *value*.

/ !\ Files must be sorted by the field they will be joined to work properly.

File editing

vi
vim Vi Improved
nano
emacs

man *command* : display the *command*'s manual page

Jacques Dainat - 2015