

Answers to exercises

man is your friend !

- Commands and their meaning :
`ls -F`
→ a file type indicator is appended at the end of the filename (“*” for an ordinary file, “/” for a directory,...)
`date --reference=/etc/passwd`
→ print the last modification time of the specified file
- To display the current date in “dd/mm/yyyy” format :
`date +%d/%m/%y`
- To display current date in Unix time (number of seconds that have elapsed since “epoch”, i.e. since 1970-01-01 00:00:00 UTC) :
`date +%s`

Doing research in file content with grep

- To download the file :
`wget http://access... --no-check-certificates`
- To extract the file :
`tar xvzf 20171002.tgz`
- To count occurrences of the word 'LOG_ERROR' :
`grep LOG_ERROR 20171002 | wc -l`
- To count events with 'Exit_status=0' :
`grep 'Event_status=0' 20171002 | wc -l`
- To count events not with 'Exit_status=0' :
`grep -v 'Event_status=0' 20171002 | wc -l`

Sorting files with sort

- To download the archive and extract its content :
`wget http://access... --no-check-certificates`
`tar xvzf cpusecnodes.tgz`
- Lines are formatted liked this : <number>;<hostname>
- To sort the file on the first column :
`sort -n cpusecnodes > cpusecnodes.sorted`

Sorting numbers with sort

- To create and edit the file with nano :

```
nano file_to_sort.txt
```

And then, write the content, save the changes with CTRL+O, and exit nano with CTRL+X. You will be asked to confirm the name of the file in which to save changes : just type ENTER to validate the default name.

- Trying to sort numerically with :

```
sort file_to_sort.txt
```

will not work because alphabetically, “2” comes after “12” !

- If you want to consider the numerical value of each line, then add the option “-n” :

```
sort -n file_to_sort.txt
```

- To remove duplicate lines, add option “-u” :

```
sort -u -n file_to_sort.txt
```

Monitoring long process with time and top

Sorting and filtering command results thanks to pipes

- To sort the output of “ps aux” by user name :

```
ps aux | sort
```

- To print the output of “ps aux” without the processes belonging to root :

```
ps aux | grep -v 'root'
```

Comparing files with diff

- To download the 2 files :

```
wget --no-check-certificate http://access...
```

- To find which machines were added or removed :

```
diff 20161001 20171001
```

File searching with find

- To find files in “/bin” with size bigger than 100K :

```
find /bin/ -size +100k
```

There are two traps in this exercise :

- the path in which the search is done must be ended by “/”;
- the unit symbol must be a lower case “k”.

- To find files in “/etc” whose name is ending in “.conf” :
`find /etc/ -name "*.conf"`
- The command :
`find ~/ -name "*.txt" -mtime -60 -exec cat {} \;`
 will print the content of files in your home directory whose name is ending in “.txt” and which have been modified during the last 60 days.

Vertical slicing with cut

- To edit the file :
`nano fruits.txt`
 The command to print out the first 3 letters of each line is :
`cut -c 1-3`
- To edit the file :
`nano beatles.txt`
 The command to print out the second column is :
`cut -d: -f2 beatles.txt`

A script that prints to a file a report of all the processes

- To check the result of the script, you can verify that the report was well created like this :
`ls -al report_*`
 and then check the content of the report :
`cat report_*`

Using string checking operators in “if” commands

- The '-z' is a unary operator that checks if its operand is null. The condition :
`[-z $string]`
 is true if \$string is null.
- The role of the line 'exit 1' is to stop the execution of the script with an exit code equals to 1. Remember that an exit code not equal to 0 means that there was no error.

Lab 1 : Exploring a big text file

- Getting the text file...
- To get the size in human readable format and the permissions :
`ls -lh 4300-0.txt`
- To get the number of words :
`wc -w 4300-0.txt`

To get the number of lines :

```
wc -l 4300-0.txt
```

- Parsing the file with less :

```
less 4300-0.txt
```

and then, just use arrows (↑ and ↓) and/or PgUp and PgDn keys to explore the file content. To search for the word “Molly”, type “/Molly”, and then jump from one occurrence to another using “n” key (“n” for “next”).

- Parsing the file with nano :

```
nano 4300-0.txt
```

Searching with nano is bit more difficult : CTRL+W followed by the word to search for, and then ALT+W to jump from one occurrence to another. Note : this search is not case-sensitive !

- Parsing the file with grep :

```
grep molly 4300-0.txt
```

```
grep Molly 4300-0.txt
```

```
grep -i molly 4300-0.txt
```

The last command is case-insensitive (option '-i').