School of Computer Science

COMP47470

Lab 1
CLI (Bash) for Big Data

Teaching Assistant:	Leandro Batista de Almeida
Coordinator:	Dr Anthony Ventresque
Date:	Monday 21st January, 2019
Total Number of Pages:	11

Linux is an Operating System that is widely used in industry - in particular a lot of "Big Data" jobs run or use Bash commands, i.e., commands based on the command line interpreter often used by Linux/Unix systems. In this lab, you will play with some basic bash commands that we will use later to run Big Data jobs.

1 Connecting to your Linux VM

Mac OSX and Linux This is very straightforward, we simply open up the **Terminal** application and type:

```
$ ssh username@hostName
```

That's it!

Windows For windows, using ssh is slightly trickier, we need to download a piece of software that will allow us to connect. For a tutorial with visuals take a look at this: `https://mediatemple.net/community/products/dv/204404604/using-ssh-in-putty-`

- Download Putty from: `http://www.putty.org/`
- When you have downloaded it, find the putty.exe (the executable file) and double-click.
- Once it is installed, run the application.
- Under 'Host Name' enter the host name that was given to you.
- For 'Port', insert '22'
- The connection type will be SSH
- Hit open. (Note: You can actually save configurations with Putty which may be useful for you going forward.)
- Enter the password for the VM when prompted and your connected!

Another option, which might be easier, is to enable a recent Windows 10 update. Have a look at this tutorial:
`https://www.howtogeek.com/336775/how-to-enable-and-use-windows-10s-built-in-ssh-commands/`.

2 (Optional) Using Bash on your Laptop

Bash is the name of the command line interpreter (CLI) that you will be using in this course. Today, in fact, you will be using some Bash commands. Bash can be used on different operating systems, however the way you obtain it will depend on your current operating system. See below for details if you would like to have bash on your local machine.

Use Mac OSX' terminal app

Use Linux terminal app

Install Ubuntu as an app on Windows: `https://www.microsoft.com/en-ie/store/p/ubuntu/9nblggh4msv6`

Install Bash for Windows: `https://www.howtogeek.com/249966/how-to-install-and-use-the-linux`

(not recommended)

3 Using man to Get Help

Nearly every command and application in Linux will have a man (manual) page, so finding them is as simple as typing `man command` to bring up a longer manual entry for the specified command.

- For example, `man mv` will bring up the mv (move) manual.
- Move up and down the man file with the arrow keys, and get back to the command prompt with "q".
- `man man` will bring up the manual entry for the man command, which is a good place to start!
- `man intro` is especially useful - it displays the "Introduction to user commands" which is a well-written, fairly brief introduction to the Linux command line.
- Some software developers prefer info to man (for instance, GNU developers), so if you find a very widely used command or app that doesn't have a man page, it's worth checking for an info page.
- Virtually all commands understand the `-h` (or `--help`) option which will produce a short usage description of the command and its options, then exit back to the command prompt. Try `man -h` or `man --help` to see this in action.
- man pages can be lengthy, so if you are looking for a specific option etc. it could be useful to look up some word using the syntax `/word` – and then hit the key `n` to move to the next occurrence. For example, try to find the option `-t` in the man page of python.
- If you aren't sure which command or application you need to use, you can try searching the manual pages. Each manual page has a name and a short description.
  - If you know part of the command name, use the following command: `whatis -r <string>`. For example try with the following: `whatis -r cpy`
  - To search the names or descriptions for `<string>` enter: `apropos -r <string>`. For example, `apropos -r "copy files"` will list manual pages whose names or descriptions contain copy files.

4 File & Directory Commands

The Linux hierarchy is typical of Unix systems (with some variations depending on the specific distributions). For the moment you just need to know that the file system is a tree that starts at the root (represented with the symbol `/`). Note that if you are familiar with DOS/Windows the path delimiter is the foward slash and not the backward slash... A path then looks like this in Linux: `/var/log/auth.log`. This leads to the file `auth.log` in the repository `log` in the repository `var` which is right after the root of the file system.

- The tilde (~) symbol stands for your home directory. If you are anthony, then the tilde (~) stands for `/home/anthony`.
- `pwd`: The pwd command will allow you to know in which directory you're located (pwd stands for "print working directory"). Example: "pwd" in the Desktop directory will show `~/Desktop`.
- `ls`: The ls command will show you ('list') the files in your current directory. Used with certain options, you can see sizes of files, when files were made, and permissions of files. Example: `ls ~` will show you the files that are in your home directory. Check some of the options of `ls`.
- `cd`: The cd command will allow you to change directories. When you open a terminal you will be in your home directory. To move around the file system you will use cd. Examples:
  - To navigate into the root directory, use `cd /`
  - To navigate to your home directory, use `cd` or `cd ~`
  - To navigate up one directory level, use `cd ..`
  - To navigate to the previous directory (or back), use `cd -`
  - To navigate through multiple levels of directory at once, specify the full directory path that you want to go to. For example, use, `cd /var/log` to go directly to the `/log` subdirectory of `/var/`. As another example, `cd ~/Desktop` will move you to the Desktop subdirectory inside your home directory.
- `cp`: The cp command will make a copy of a file for you. Example: `cp file foo` will make an exact copy of "file" and name it "foo", but the file "file" will still be there. If you are copying a directory, you must use `cp -r directory foo` (copy recursively). (To understand what "recursively" means, think of it this way: to copy the directory and all its files and subdirectories and all their files and subdirectories of the subdirectories and all their files, and on and on, "recursively")
- `mv`: The mv command will move a file to a different location or will rename a file. Examples are as follows: `mv file foo` will rename the file "file" to "foo". `mv foo ~/Desktop` will move the file "foo" to your Desktop directory, but it will not rename it. You must specify a new file name to rename a file.
- `rm`: Use this command to remove or delete a file in your directory.
- `rmdir`: The rmdir command will delete an empty directory. To delete a directory and all of its contents recursively, use `rm -r` instead.

- `mkdir`: The mkdir command will allow you to create directories. Example: `mkdir music` will create a directory called ”music”.

Let’s start with some exercises!

Exercise

- Explore the filesystem tree using cd, ls and pwd. Look in /bin, /usr/bin, /sbin, /tmp and /boot. What do you see?
- Make a directory called BigData (mkdir) in your home directory and then rename it to COMP47470 (mv)
  Solution:
  mkdir OperatingSystems
  mv BigData COMP47470
- Make a sub-directory lab1 and in it start a text file lab1_commands.txt using vi (or nano if you are not familiar with vi) where you’ll write all the examples and exercises you’ll do today.
  Solution:
  mkdir lab1
  cd lab1
  vi lab1_commands.txt **(can also use nano if you don’t like vi)**
- Copy the file /etc/passwd into your home directory (cp).
  Solution:
  cp /etc/passwd ˜
- Move it into the subdirectory COMP47470 and create a symbolic link to /proc/meminfo called meminfo_link (look at man ln).
  Solution:
  mv passwd COMP47470
  ln -s /proc/meminfo meminfo_link
- use the following command `ls /var/log/ | wc -w` to get the number of files in `/var/log/`. How does `wc` work? (check this in the `man` page corresponding to `wc`). We will spend some time on the `|` symbol later – but do you have an idea of what it does? try `ls /var/log/` alone to give you an idea of what happens.

5 Some System Information Commands

- `df`: The df command displays filesystem disk space usage for all mounted partitions. `df -h` is probably the most useful - it uses megabytes (M) and gigabytes (G) instead of blocks to report. (-h means ”human-readable”)
- `du`: The du command displays the disk usage for a directory. It can either display the space used for all subdirectories or the total for the directory you run it on. Two

interesting options are -s (”Summary”) and -h (”Human Readable”)

- `free`: The free command displays the amount of free and used memory in the system. `free -m` will give the information using megabytes, which is probably most useful for current computers.
- `top`: The top (’table of processes’) command displays information on your Linux system, running processes and system resources, including CPU, RAM & swap usage and total number of tasks being run. To exit top, press ”q”.
- `htop`: has my preference over `top`.
- `uname -a`: The uname command with the -a option prints all system information, including machine name, kernel name & version, and a few other details. Most useful for checking which kernel you’re using.
- `lsb_release -a`: The lsb_release command with the -a option prints version information for the Linux release you’re running.

Examining files. *Why use more if you have less*

”cat” stands for conCATenate. You can use this command to dump the entire text file to the screen.

```
$> cat /etc/passwd
```

If the text file is too long, you might find that it scrolls past too quickly and you cannot see the beginning of the file anymore. In which case, you can use either the ”more” or ”less” command.

```
$> more /etc/passwd
```

Or

```
$> less /etc/passwd
```

Both these commands perform similar functions as they allow to see the text file one page at a time. You use the spacebar to continue paging, `<enter>` key will move down one line, and ”q” to quit. ”less” has actually more features than ”more” :) The most useful feature is that it can scroll backwards (or up) whereas ”more” cannot. Press ”h” (while in the program) to see more options. Another interesting option is the search option, which is similar to the one you’ve seen when we presented `man`. If you are looking for a specific string in a text file use the syntax `/string` - and then hit the key n to move to the next occurrence. For example, try to find your login name in `/etc/passwd`.

Show part of files. *Can you make heads or tails of it?*

Note: `/etc/passwd` is a file storing some login info for a user: user number, group number, shell at login, home directory etc. It also contains OS processes considered as ”users”. Use commands `whoami`, `id`, `groups` to try to get some of the information contained in `/etc/passwd`. `head` and `tail` are two opposite commands, showing the beginning or the end of a file respectively. Try the following commands – what do they give you?

```
$> head /etc/passwd
```

```
$> tail /etc/passwd
```

Both commands have various options that can be powerful – and a little complicated:

- `$> head -n 3 /etc/passwd` shows the first 3 lines
- `$> tail -n 5 /etc/passwd` shows the last 5 lines
- `$> tail -n +4 /etc/passwd` shows from the fourth line to the end
- `$> head -n -1 /etc/passwd` shows from the second line to the tenth line

A classical use of `tail` consists in showing the content of a dynamic file, that is evolving over time. `/var/log/` contains a lot of files that store log messages, i.e., information about what's happening in the system. Try `tail -f /var/log/syslog`.

Exercise

- Combine head and tail using a pipe to print 15th line to 20th line in /etc/passwd.
  Solution:
  tail -n +15 /etc/passwd | head -n 6
- Combine head and tail using a pipe to print the 7th line in /etc/passwd
  Solution:
  tail -n +7 /etc/passwd | head -n 1

Echo and Special Characters.

The goal of this section is to play with special characters and to show how they impact the execution of commands, and can be disabled conveniently. If you need to interrupt/stop the execution of a command (if you're stuck), you can use the combination of keys *control* + *c*.

- print on the screen a simple word, for instance "hello" (use echo).
- echo can print several strings - print "hello everybody"
- you can also use echo to print non alphabetical characters on screen (as long as they do not belong to the list of special characters). Use echo to print "Hello to the 2 of you!"
- Disable the special character # in the following string (using the character \) and print it on screen using echo: "# is a very useful character in bash"
- Likewise for the following: "# is more useful than \ in bash"
- Print the following using echo: "# is more useful than \ in bash but less than *"
- When you need to disable a lot of special characters, you can use simple and double quotes to disable all (simple quotes) or some (double quotes - do not disable $, ' and \) of them. Print the following two sentences using simple then double quotes: "The $PATH variable is very important."

Selecting Columns. *Charles I of England?*

Linux command cut is used for text processing. You can use this command to extract portions of text from a file by selecting columns.

Option -cN extracts only column (character) N of each line from a file:

```
$> cut -c2 /etc/passwd
```

will extract only the second character of each line (the 2nd column of each line).

Range of characters can also be extracted from a file by specifying start and end position delimited with -.

```
$> cut -c2-5 /etc/passwd
```

Either start position or end position can be passed to the cut command with the -c option.

The following command specifies only the start position before the '-'. This example extracts from the 10th character to the end of each line.

```
$> cut -c10- /etc/passwd
```

The following commad specifies only the end position after the '-'. This example extracts 10 characters from the beginning of each line.

```
$> cut -c-10 /etc/passwd
```

Instead of selecting x number of characters, if you like to extract a whole field, you can combine options -f and -d. Option -f specifies which field you want to extract, and option -d specifies the field delimiter that is used in the input file.

The following example displays only the first field of each line from the /etc/passwd file using the field delimiter : (colon). In this case, the 1st field is the username.

```
$> cut -d":" -f1 /etc/passwd
```

You can also extract more than one fields from a file. The example below displays the username and home directory of users.

```
$> cut -d":" -f1,6 /etc/passwd
```

To display a range of fields specify start field and end field as shown below. In this example, we are selecting fields 1 through 4, 6 and 7 (fields 1, 2, 3, 4, 6 and 7).

```
$> cut -d":" -f1-4,6,7 /etc/passwd
```

Sorting. *The Last Shall Be First*

The sort command rearranges the lines in a text file so that they are sorted lexicographically.

```
$> sort /etc/passwd
```

These are the default sorting rules:

- a number is before a letter.

- letters follow their order in the alphabet.
- a lowercase letter is before a same uppercase letter.

You can change the default sorting rules by providing a suitable parameter. E.g..

- Reverse sort:

```
$> sort -r /etc/passwd
```

- Ignore case:

```
$> sort -f /etc/passwd
```

You can also concatenate and sort several files at once:

```
$> sort /etc/passwd /etc/group
```

You can also save the result of the sort in a another file:

```
$> sort /etc/passwd -o result.txt
or
$> sort /etc/passwd > result.txt
or
$> sort /etc/passwd >> result.txt
```

The -o options is no available for all commands. Using > and >> works for any command. > will overwrite result.txt if it already exists while >> will append the output to result.txt if it already exists. Both create the file if it does not exist.

Duplicates. *Castor Troy: It's like looking in a mirror, only not.*

This command filters duplicate adjacent lines.

```
$> uniq /etc/passwd
```

It can filter them even by ignoring the case:

```
$> uniq -i /etc/passwd
```

You can report the duplicate lines by:

```
$> uniq -d /etc/passwd
```

You can also print the number of occurrences of each line:

```
$> uniq -c /etc/passwd
```

Grep. *Look it up!*

Grep prints out the lines containing a certain string.

```
$> grep root /etc/passwd
```

Options:

- -c: only gives the number of matching lines
- -v: Shows only the lines that do not match the pattern. Inverted search.
- -i: ignore case
- -n: gives the line number as well as the matching lines.

Getting files on and from the server

At times you will need to upload files from your own machine to the server, or download some of the files you have on the server to your local machine. Here are some ways of doing this.

scp

The scp command works just as cp but lets you transfer files from one machine to another one. You can use it if you have a bash terminal on your machine. The syntax is as follows:

```
$> scp source target
```

For example, if we want to download the file unirank.csv from the home folder ($\tilde{}$) of user aventresque on the csserver to the directory we are currently in (./), we would write:

```
$> scp aventresque@csserver.ucd.ie:~/unirank.csv ./
```

If we want to upload unirank.csv from the current folder to the home folder of user aventresque on the csserver then :

```
$> scp unirank.csv aventresque@csserver.ucd.ie:~/
```

To copy directories, we use he -r option.

FileZilla

FileZilla offers a graphical interface that allows you to do the same thing as scp. It is available on windows, macOS and linux at `https://filezilla-project.org/download.php`.

Using FileZilla is pretty straightforward. If needed you can find instructions at `https://wiki.filezilla-project.org/FileZilla_Client_Tutorial_(en)`

public_html

You might need to share files with others who do not have access to your home directory. In order to do this you can put the files in a directory called public_html and they will be publicly available at the URL `http://csserver.ucd.ie/~username/fileName`.

Exercise 1

- What is the output of the commands:
  - `echo {0..9}`
  - `echo 1.{0..9}`
  - `echo {A..C}{0..2}`
- Use sort to sort the content of `/etc/passwd`
- Use grep to list only the line from `/proc/meminfo` that shows total memory (the name of the variable is MemTotal).
- Use grep to find all the lines with ”to” in the `syslog` file (`/var/log/alternatives.log.1`). Then redirect this output into a file in your home directory.
- List all of the files in a directory in reverse date order, so that the most recent is at the bottom (ls).

Solutions

```
sort /etc/passwd

grep MemTotal /proc/meminfo

grep to /var/log/syslog > ~/lab2.txt #(Overwrite any changes)
# or
grep to /var/log/syslog >> ~/lab2.txt #(Append to file)

# You can do this entirely with ls
ls -tr

# -t : 'sort by modification time, newest first'
# -r : 'reverse order while sorting'
# Optionally, you can use -l for the long listing format
ls -ltr
```

Exercise 2

In this exercise, we will be exploring a dataset containing information regarding universities. Follow these steps using the commands described above.

1. Download the dataset as unirank.csv using wget (capital o option):

```
wget -O unirank.csv http://csserver.ucd.ie/~aventresque/unirank.csv
```

2. Explore the file and its structure (`cat`, `head`, `tail`, `less`, ...)
3. Find all the ”colleges” in the list (`grep`).
4. List and group all the universities by state (`cut`, `sort`).
5. Find the number of institutes per state (`cut`, `sort`, `uniq`). (which state has the most institutes in the dataset?)
6. Create a list (x, y) of couples containing the Tuitions and Fees (x) and the rank (y) for each institute. Plot the values using Excel, gnuplot, etc. What do you notice?

Solutions

```
grep -i college unirank.csv | grep -v -i university
cat unirank.csv | cut -d, -f1,3 | sort -k 2 -t ","
cat unirank.csv | cut -d, -f3 | sort | uniq -c
cat unirank.csv | cut -f4,6 -d, > plot.csv
```