

COL100 Lab 2

I semester 2017-18

Week 2, 2017

Objective

More familiarisation with Linux and its standard commands

Part 1

1. Login to your system and open a terminal window.
2. Open the web-browser and visit the page <https://moodle.iitd.ac.in> and visit the COL100 course page.

Go to the news forum and from there visit the COL100 course home page

From the course home page, visit the link to Piazza for course discussions. Create a Piazza account and sign-up for the course at Piazza.

3. In this exercise we will learn about the **less** command which is very similar to the **more** command with more features. If you have a file, then the simplest method to view its content will be using the **cat** command. The cat command “concatenates” the contents of all the files given in its command line and prints the results to its standard output, i.e., your terminal window. Type the following command:

```
ls /bin > bin_dir.txt
```

Recall that if a command is followed by a “>” sign and then by a filename then the standard output of the command is sent to the file instead of being displayed on the terminal. In the above command the standard output of the ls command is sent to the file bin_dir.txt in your current directory.

Use the ls command to check that the file bin_dir.txt has been created and then run the following commands

```
cat bin_dir.txt
```

The contents of the file were too much to see in a terminal window. They would have scrolled past. Type man cat to learn more about cat. To see contents of big files we use **more** and **less** commands. In this session we will experiment with the less command. You are already familiar with the **man** command. To get complete details of **more** or **less** commands, you may use the **man** command (which in-turn uses the less command to display the manual pages). Now run the following

command and while the command is active, experiment by pressing the following keys SPACE BAR, f, b, q.

less bin_dir.txt

What do these keys do when the less command is active?

To search forward for a pattern, type / followed by pattern. To search backwards for a pattern, type ? followed by pattern. So to look for the string zip in the file bin_dir.txt, type /zip when less is active. Do the following to illustrate this run the following commands:

less bin_dir.txt

(and without pressing the q, type /zip. What do you see? Press the n and N keys to find the next and the previous matches. Also experiment with typing ?zip and then the keys n, N to move forward and back. Finally type q to quit).

We will now use the man command and search in the man pages. Type the following command to get the manual page on the **less** command.

man less

(Now type /COMMAND to reach to the command section directly and use SPACE and b key to scroll forward and back). Instead of reading the complete manual pages, you can use the search facility to directly get to the topics you are interested in.

Now open your web browser and go to www.google.com, and type the following on the google search bar and follow the links to get more information on the less command

[how to use less command in linux](#)

4. Now we will experiment with the find command which is used to find specific files and directories in the file system. If you have not yet finished the previous lab assignment, please do so now. In the previous lab assignment, you would have experimented with the find command. Now use the **find** command to find all the files with names containing the string zip. How many files are there in the /bin directory with the name containing the string zip. If you haven't figured this out, the command to do this is given below

find /bin -name *zip* -print

How many such files are in the /usr/bin directory?

Now you can place the * appropriately to answer the following questions:

How many files in the /usr/bin directory are there with names starting with zip?

How many files in the /usr/bin directory are there with names ending with zip?

5. Now we will experiment with the grep command which is used to print all lines in a file that contain a given pattern. Type the following command

grep zip bin_dir.txt

This command will print all the lines in the file `bin_dir.txt` that contain the string `zip`. Since the file `bin_dir.txt` was the output of the `ls /bin` command, you should get names of all the files in the `/bin` directory that contain the pattern `zip`.

Now look at the man page of `grep` and search for *word* (remember how to search while `less` command is active) and read the section explaining the `-w` option. Now experiment with the `-w` option of the `grep` command by running the following commands:

```
grep -w zip bin_dir.txt
```

```
grep -w gzip bin_dir.txt
```

The above commands can be run directly without storing the results in the intermediate files using the pipe construct. Without running the commands, write down what you think they will output.

```
ls /bin | grep zip
```

```
ls /bin | grep -w zip
```

```
ls /bin | grep -w gzip
```

```
ls /usr/bin | grep zip
```

```
ls /usr/bin | grep -w zip
```

```
ls /usr/bin | grep -w gzip
```

Now run the above commands and compare your results.

Visit google.com on your browser and type the following to get more help on `grep` command:

[How to use grep command in linux](#)

6. Now we will experiment with the `wc` (word count) command.

Read the manual page of `wc`. Now type the following and carefully observe and explain the output

```
wc bin_dir.txt
```

Construct a single command to count the number of files in `/bin` directory using the commands `ls` and `wc` commands and pipes.

Modify your command to count the number of files in the `/usr/bin` directory.

Construct commands to count the number of files in `/bin` and `/usr/bin` directories whose names contain the letter `a`, using the commands `ls`, `grep` and `wc` commands and pipes

7. Now we will explore the `sort` command which is used to sort the lines in a given file. Read the manual page of the `sort` command.

Construct a command to print the files in `/bin` directory in descending sorted order (filenames starting with `z` should come first followed by those starting with `y` and so on).

```
ls /bin | sort -r | less
```

Now view the contents of `/usr/bin` directory in the descending sorted order.

8. The `tr` (translate) command can be used to replace one set of characters with another. To replace `a` with `e` you may type `tr a e <filename>`. By default `tr` reads standard input and carries out the translation and prints the results to standard output. Fun exercise: for converting all the `a`'s into `e`'s in the manual page of `tr` type the following command

```
man tr | tr a e | less
```

To convert `a` to `e`, `e` to `i`, `i` to `o`, `o` to `u` and `u` to `a` use the following:

```
man tr | tr aeiou eioua | less
```

Rest is up to your imagination and creativity.

Part 2

1. Use the `gedit` command to create a file named `myFirstProgram.c` with the following contents:

```
#include <stdio.h>

int main(int argc, char *argv[]) {
    printf("Who am I?\n");
    return 0;
}
```

2. Using the `chmod` command, change the permission of your file `myFirstProgram.c` so that no one other than you can read this file.

3. Give the full path of your file (using `pwd`) to your neighbour. Ask your neighbour to try to read the content of your file using the `less` command on his/her terminal (and she/he should be unable to do so).

4. Compile the program into machine language by typing the following command:

```
gcc myFirstProgram.c -o myFirstProgram
```

5. Run your program by typing:

```
./myFirstProgram
```

6. Experiment with the following linux commands which are very useful:

- a. `uniq` – to print unique lines in a file
- b. `ps` – to list the currently running processes in the system
- c. `kill` – to terminate a currently running program
- d. `umask` – to set the default file permissions
- e. `nice` – to change the priority of your program
- f. `sed` – to change the contents of files according to a pattern (more advanced than `tr`)
- g. `xargs` – execute one command multiple times with multiple inputs taken from a file.