TRINITY COLLEGE DUBLIN, SCHOOL OF MATHEMATICS

Assignment #1, Module: MA5621

Gustavo Ramirez

October 24, 2016

1 PROBLEM DESCRIPTION (NOTE: MOST OF THE SOLUTIONS WERE TESTED IN UBUNTU 16.04 LTS)

1. Name 6 different Linux distributions, and state which major packaging format they use (e.g. RPM or DEB).
2. Give a brief overview of the different sections of the Man Pages (note: not the parts of an individual page NAME, SYNOPSIS etc).
3. Describe how you would change the shell prompt to make it BOLD and display in the colour red.
4. Given a directory of files, how would you list the files while sorting them in file size in descending order? And in ascending order?
5. Starting in your home directory (on 'chuck'), list at least two ways to change to the '/tmp' directory.
6. On the command-line, give at least 2 examples of how would you delete a file called '-i' (without the qoutes).
7. Give at least 2 ways to show the contents of a directory called "My Docs" using 'ls -l'. I.e. how do you get around SPACEs in a file/folder name using the shell.
8. Using the 'date' command, how would you display the current date/time in the following format: YYYYMMDD-DHHMMSS. How would you display the date in that format from exactly 1 week ago from the time you run the command?
9. What is the Unix Epoch? How would you display it using the 'date' command?
10. Using the 'cal' command, how would you display the full calendar year, with weeks starting in a Monday?
11. What is the purpose of the PATH environment variable. How would you add a new location to it?
12. Using shell wildcards (fileglobs), how would you list:
    - All files starting with the letter k?
    - All files starting with the letter k, with a .txt extension?
    - All files starting with an upper case letter?
    - All files with 4 characters, where the third character is a number?
13. What is the purpose of the umask? How would you ensure that every new file that you create has the following permissions: full access for you, read-only access for your group, and no access for anyone else?
14. How would you change the behaviour of the delete command to prompt for confirmation before deleting files?

1.1

- Ubuntu (packaging format: DEB)
- Debian (packaging format: DEB)
- Red Hat Linux (packaging format: RPM)
- Elementary OS Freya (packaging format: DEB)
- Fedora (packaging format: RPM)
- Linux Mint (packaging format: DEB)

1.2

This information is described in the Man Pages for the command *man*, with the following descriptions for each of the 9 sections:

1. Executable programs or shell commands
2. System calls (functions provided by the kernel)
3. Library calls (functions within program libraries)
4. Special files (usually found in /dev)
5. File formats and conventions eg /etc/passwd
6. Games
7. Miscellaneous (including macro packages and conventions), e.g. man(7), groff(7)
8. System administration commands (usually only for root)
9. Kernel routines [Non standard]

1.3

There are four variables for customizing the shell prompt: PSi (with i going from 1 to 4). In specific, changes are to be made here to variable PS1, for enabling bold letters and modifying the colour to red. The specification of the value of PS1 is in the ~/.bashrc file (with permissions: rw-r--r--).

In that specific PS1 definition, some special scape characters are to be used. Particularly, to add a new property, it's necessary to use \[\e [X \] at the beginning and \[\e [0m \] at the end, where X is the specification of the properties. For example, if X = 1;91m, means bold (due to the 1) and red (due to the 91m).
In conclusion, if the pattern in PS1 is complicated, just look for the appearances of the characters [0M ; NNm], where M = 1 for bold, and M = 0 for light, and NN denotes the color (NN = 91 for red, for example)
There are other ways of changing the color and bold properties of the shell prompt, but those are more dependent on the distribution and specific shell.

1.4

From the man pages of the ls command, the -S flag represents sorting by file size.
Also from the man pages of ls, the -r flag is for reversing the list being displayed.
Therefore, for listing in ascending order:

```
$ ls -rS DIR/
```

and in descending order:

```
$ ls -S DIR/
```

where DIR/ is the directory for which the files are being sorted.

1.5

Specifying an absolute path:

```
$ cd /tmp
```

or specifying a relative path:

```
$ cd ../../../../tmp/
```

1.6

First, to create the file named "-i", one way is to do the following:

```
$ touch ./-i
```

Then, in order to delete it, one way is the following:

```
$ rm ./-i
```

Other way is to add a – signal at the end of all flags, which disables further option processing by shell. Like this:

```
$ rm -- -i
```

1.7

The first way, suggested by autocomplete in the terminal:

```
$ ls My\ Docs/
```

and the other is making use of quotes:

```
$ ls 'My Docs'
```

1.8

The way the modify the output of the date command, is writing it in the structure:

```
$ date +FORMAT
```

so, in this specific case:

```
$ date +%Y%m%d%k%M%S
```

On the other hand, according to the man page for the date command, specifically for the -d flag: "display time described by STRING, not 'now'". The -d flag can receive an argument specifying the number of days, like this:

```
$ date -d '8 days'
```

and in the case requested (also associating the requested format):

```
$ date -d '-7 days' +%Y%m%d%k%M%S
```

1.9

According to the man page of the date command, at the EXAMPLES section, the first example is:

```
$ date --date='@2147483647'
```

and the description of that example there is: "Convert seconds since the epoch (1970-01-01 UTC) to a date".

Then, the Unix Epoch is the number of seconds that have passed since 1970-01-01 UTC (at 00:00:00).
In the description of FORMAT, also at the man page of date, one of the options is %s which is "seconds since 1970-01-01 00:00:00 UTC". Then, to display it using date:

```
$ date +%s
```

1.10 (TESTED IN SCIENTIFIC LINUX)

In Ubuntu, the -M flag (for weeks starting on Monday) doesn't work with cal, so, Scientific Linux was used for this specific question. But apparently, in Scientific Linux the cal command already displays weeks starting on Monday.
For setting the weeks to start on Monday, the -M flag is used. The -y flag is for displaying the full year. Then:

```
$ cal -M -y
```

1.11

PATH is[1] an environmental variable in Linux and other Unix-like operating systems that tells the shell which directories to search for executable files (i.e., ready-to-run programs) in response to commands issued by a user.
To be able to modify PATH, first is good to be able to access it. To print PATH in the terminal:

```
$ echo $PATH
```

and to modify it:

```
$ export PATH=$PATH:EXTRA_PATH
```

where EXTRA_PATH is the addition path which wants to be included in the PATH variable.
If this is to be made permanent for future sessions, it's necessary to add it to .bashrc.

1.12 (TESTED IN SCIENTIFIC LINUX)

There are 3 wildcards: *, ? and []. The first one stands for a string of any size, the second one for a single character, and the third one selects one character out of a set.
The previous paragraph implies that the requested lists are implemented like this (in the order specified in the description of the problem):

[1] Taken from: http://www.linfo.org/path_env_var.html

1. ls k* | grep -v ^d
2. ls k*.txt | grep -v ^d
3. ls -d [[:upper:]]* | grep -v ^d
4. ls -d ???? | grep -v ^d

and in the third previous command, first an ls is made such that all the files and directories are displayed, starting with a capital letter, and then they are filtered with the use of grep such that only the files are displayed (important: the -d flag in ls, prevents ls to display the content of subdirectories that match the pattern).

1.13

The command *umask* determines the settings of a mask that controls how file permissions are set for newly created files. Then, when a new file is created, the default permissions will be the ones associate to the settings specified by *umask*.
The permissions specified and required in the assignment, are: rxw | r-- | ---; but because *umask* is a mask, and does not actually set permissions, these permissions have the octal reprentation: 640 (i.e. *umask* doesn't interfere with execution permissions for files, although it does for directories). Then, if we perform the subtraction to obtain the umask: 666 - 640 = 026. Finally, to set this umask:

```
$ umask 026
```

1.14

Two options are:

- for the regular user or for root (depends on which case wants to be implemented; if some users have been given access to root, then this can be implemented in /root/.bashrc as a precaution), in the file .bashrc, add the line: alias rm = 'rm -i'.
- when executing the rm command, simply add the -i flag, and then something like "**rm: remove regular empty file 'kkajshdkflhs'?**" will be prompted.