

UNIX COMMAND LINE EXERCISE

1. Use the `cal` command to find out which day the 31th of December 2002 was on. `cal` takes two parameters, the number of the month (e.g. 1 for January, 8 for August) and the year as a 4 digit number (e.g. 1997).
2. `cal` can also work with 1 parameter, which is the *year*. When given a year, it displays a calendar for the whole year. Use this feature but *redirect* the output of `cal` so that the calendar for 2005 is saved to a file called "yearfile" in the default directory of current user. (Note: investigate `>` operator for redirection of the output and also note `cd` command returns you to default directory)
3. Use the `ls` command to list all the files/directories (including hidden ones) you have in the default directory. You need to pass an additional option to `ls` command to do this. **What are "." and ".." appearing in the list?**
4. Which **file** in /usr/games directory has the smallest size? Use the command `ls -alSr` for this and explain used options referring to `man ls`. `man` command prints the manual for each command supplied.
5. Use redirection to create a file called "thismonth" under default directory, containing only this month's calendar. Then use the `date` command to append the current date to the end of the same file. Investigate `>>` operator for appending purposes. To check the contents of a file, use `cat` command.
6. Copy the file yearfile to yearfile2. Copy the file yearfile to yearfile3. The command for copying is `cp`.
7. Change the name of file yearfile3 to thisyear with `mv` command.
8. Delete the file yearfile with `rm` command. Create an empty file called yearfilenew.
9. Construct the directory structure given under default directory using `cd` and `mkdir` commands. What is the usage of `pwd` command?

```
          demo
          |
  -----+-----
  |         |         |
work      letters    scripts
  |
  -----+-----
  |         |         |
progs  tutorial    misc
```

User Permissions

The `chmod` command is used to change permissions. The simple way to use the `chmod` command is to add or subtract the permission to a file. A simple plus or minus is used to add or subtract the permission.

You may want to prevent yourself from changing an important file then you can remove the write permission of the file "myfile" with the command:

```
chmod -w myfile
```

If you want to make file "myscript" executable

```
chmod +x myscript
```

You can add or remove more than one of these attributes at a time

```
chmod -rwx file  
chmod +wx file
```

You can also use the "=" to set the permission to an exact combination. This command removes the write and execute permission, while adding the read permission:

```
chmod =r myfile
```

Note that if a **directory** has read permission, you can see what files are in the directory. That is, you can do a `ls` command and see the files inside the directory. However, read permission of a directory does not mean you can read the contents of files in the directory. Write permission means you can add a new file to the directory. It also means you can rename or move files in the directory. Execute allows you to use the directory name when accessing files inside that directory. The "x" permission means the directory is "searchable" when searching for executables.

All files have an **owner** and a **group** associated with them. There are three sets of read/write/execute permissions: one set for the user of the file, one set for the group of the file, and one set for everyone else (other).

The `chmod` command has options for user (u), group (g) and other (o) permissions. You can explicitly specify u, g or o in the `chmod` command:

```
chmod u=rw myfile  
chmod g=rw myfile  
chmod ug=rw myfile
```

Following command will add read and remove write permissions for group:

```
chmod g+r-w myfile
```

Finally note that each permission combination has a corresponding octal representation as explained in tables below

rwx	7	Read, write and execute	
rw-	6	Read, write	
r-x	5	Read, and execute	
r--	4	Read,	
-wx	3	Write and execute	
-w-	2	Write	
--x	1	Execute	
---	0	no permissions	

Permission	Octal	Field
rwx-----	700	User
---rwx---	070	Group
-----rwx	007	Other

10. Use various `chmod` combinations on current and newly created files/directories to experiment.

Grep

11. Check out `grep` command and think about some scenarios where it will be useful. This link (<https://www.computerhope.com/unix/ugrep.htm>) will serve as a good starting point as a reference to `grep` utilities.

One of the usages can be finding a particular port that is listened by your computer. Such as the usage of `netstat -tulpn | grep :80` **How can netstat and grep works together?**

Compiling & Running Java Files (WINDOWS)

12. Switch to Windows. Open up a text editor and code a Hello class including main that prints out "Hello CE 306! ". Save the file as Hello.java under Documents folder.

13. Include `javac` in the path via on the command prompt

```
set path=C:\Program Files (x86)\Java\jdk1.8.0_112\bin
```

14. Using `javac Hello.java` on command prompt where the Hello.java resides will compile your file. Use `dir` (equivalent to `ls` in UNIX) to see the compiled .class file in current directory. Then run the program with command `java Hello`.

15. Unix version of compilation & running is the same way as Java is platform-independent.

References:

[1] Linuxzoo.net. (2018). Welcome to linuxzoo. Retrieved 12 February, 2018, from <https://linuxzoo.net/>

[2] Grymoire.com. (2018). Unix/Linux Permissions - An Introduction and Tutorial. Retrieved 12 February, 2018, from <http://www.grymoire.com/Unix/Permissions.html>

[3] Computerhope.com. (2018). Linux grep command help and examples. Retrieved 12 February, 2018, from <https://www.computerhope.com/unix/ugrep.htm>

[4] nixCraft. (2020). Linux Find Out Which Process Is Listening Upon a Port. Retrieved 25.02.2020 from <https://www.cyberciti.biz/faq/what-process-has-open-linux-port/>