Assingment 3

Angel Camacho CS 480 9/26/2020

I. Man Pages (20 Points)

1. chmod: This command is used to change the permisions of files. What I mean by this is the fact that certain files can be read written or executed. That beaing said this command allows us not only do that but also lets us dictate which group or user. The command can be executed as such: chmod [OPTION] [MODE] [FILE] (A important flag is the -c fag which allows one to see when a change was made)
2. chown: This command is used to modify the accisibility the ownership and group of a file. What I mean by this is that certain groups and owener can access certain files but with this command we can change that. Lets say we want to revoke group A access to use a certain file, and only let group B access it. In this Ocassion this command would be useful. The command can be excuted as such: chown [OPTION] [OWNER] [:[GROUP]]
3. crontab: This command is used to remove, edit and list difrent crontabs. A crontab is ustilized to write difrent jobs that can be schedueld to be excecuted at difrent times of the day, month or year. One can eve Schedule them to be repeated over and over dependat on what the parameters are. The command can be executed as such: crontab [-u user] [-l | -r | -e] [-i] [-s] the -l flag will print the crontabs that are running.
4. getfacl: This command is used to print the accessibility of a certain file. Interestingly enought facl stands for file access control list. So the name is pretty self explanatory. Furthermore the command when used will print the files owener and group and the file name, not to mention it should also print he access control list, which are important since it shows if they can be written, read or excecuted. The command can be excecuted as such: getfacl [-aceEsRLPtpndvh] file
5. ln: This command is used to link files. What I mean by this is the fact that they are essentially like short pcuts like a desk top short cut one could say. Further more this command while creating links between files there can be two kind of links soft links or hard links if I am not mistaken. Soft links will allow you move the files and the link will remain however hard links will not allow you to do this. The command can be executed as such: ln [OPTION] [-T] linkname.
6. login.defs: This command is used to place certain configurations on a password. This is done becuse it defines how a pasword should be encrypted not to mention the type of encryption is set everytime a new password is created. When one executes this command it will show tha passwords configurations.
7. newgrp: This command is used to login to a diffrnet group. Its really straight foward. This wiol essentailly change your current group to a diffrent group, the group that one especifies.
8. passwd: This commang will let the user change ones user password or others user

password. What I mean by this is the user will be able to change their own password. However an important thing to remeber that while root can change their own password thy can also chage other users paswords. When the comand is executed it will first ask for your current password and then your new password. Root is not required to type other users password when changing their password (did not try). If anything root can just delete the users password with passwd -d (username).

9. passwd(file): This if I am not mistaken refers to the fact that there is a file that lets root be able to see all the usres paswords. An important thing to note is the fact that if one is creating a new user and they type an "*" in the file this will allow later on to the user to use passwd and be able to change their own paswwrd.
10. shadow(file): This is a file has the passwordw for users. An important thing to note is the fact that the reason why we have this file is due to the fact that paswords are encrypted and the computer has to know to which paswword it is encrypted to if that makes sense hopefully. Another important thing is that the file also hold the "age" of the passwords meaning it holds how long has this pasword been held for.
11. setfacl: This comand is used very similarly to the get facl they are ssentially the same thing but used for diffrnet things. For ecxample in the case of set facl we willnot be getting the file acceses controll list but instead we will be setting those accesses. Further more this allows the permissions of a firectory or file to be changed not to mething we can also increase/decrease privilages certain users or groups can have for the specified directory/file. The command can be executed as such: setfacl --restore=file but theres other ways to execute it.
12. umask: This command is used to define users umasks. What I mean by this is that users specifically can dictate what permissions a new file may have. An important thing to note is the fact that when you first execute the command it will list the current users and their umask values, and their umask value which I interprete it as a sort of ID. For example a user creates a new directory or file, depending on what the umask was set to the new file/directory created will contain those specific privilages that were set at the beggining.
13. useradd: This command is used to add new users, essentailly we create a new user, it is very straight foward. The command is executed as such: useradd [options] LOGIN when the command is executed it is important to not the fac tthat the username will be decide right there and then. A cool thing about this command is the fact that when a user is created a directory for them is also created or atleast it has the ability to, when I tried it it did not add one, only by using the -D flag will it create a respective directory for them.
14. userdel: This command is very self explanatory, the useradd command si used to add users, the userdel is used to delete users. such command is important should be used with a certain flag in my opinion the -f flag is crucial due to the fact that anything that the user ever did in his directory should and will be deleted. But it is also important that it can be dangerous as maybe some files can be important.
15. usermod: This command stannds for user modification. And as the meaning goes the command does,the usermod command will then be used to modify the users files it essentially lets us be the user and be in the users files. Not only that but with this

command we are able to set certain settings to be changed for this user. The command is executed as such: usermod [options] LOGIN

II. a. Well I think the simplest form would just be to hop onto the users computer and use to command top or htop or ps. But if im root or have his password I can not do it the cave man way and use top -U (username) which should allow me to see their proccesses. Then under $CPU and %MEM I could see which task is taking all the resources. b. I belive I could just use the command kill -STOP (pid) c. kill -CONT (pid) and hope my boss does not fire me. d. kill -9 (pid) I belive this is the highest form of sending a kill signal which should terminate that task it is important to note that its not always good to use this form of kill signal.

III. The users default group is determined when the one creates the new user. One can have a pre determined group that can be changed or modified in the future; Or when the group is created. su root groupadd hwgrp480 usermod -g hwgrp480 bob

IV. su root crontab -u trevor crontab -u trevor -r (at this step just make sure that if he has multiple crontabs just to remove them all) vim /etc/cron.deny (then just type the username tevor write and quit.)

V.

```
use warnings;

my $total_rss = 0;
my $total_sz = 0;

open(PS_F, "ps -AF|");

while (<PS_F>) {
  ($uid, $pid, $ppid, $c, $sz, $rss, $psr, $stime, $tty, $time, $time, $cmd)
  print($sz . " " . $rss . "\n");
  $total_rss = $total_rss + $rss;
  $total_sz = $total_sz + $sz;
}

close(PS_F);

print("TOTAL RSS: " . $total_rss . "\nTOTAL SZ:  " . $total_sz . "\n");
```

These numbers relate to the actual ammount of physical memory and swap space in the way that the VSZ is virtual even though it may be virtual it lies over the actual physical memory, which has a "backing storage" on RAM or RSS memory.

VSZ is the total amount of memory that has been assigned to a process whether or not the process is actually using it and whether or not RSS is the true memory usage. It's the exact amount of memory the process is using

VSZ can also account for the amount of SWAP memory being used. Linux allocates a section of

the harddrive to swap data in and out of ram as it's being used. RSS does not account for SWAP memory it only accounts for the parts of memory active in the RAM.

VI.

```
su root
vim /etc/passwd

#edit it
#(picture unknown)
![](unknown.png)

cd home
mkdir jclass
chown jclass jclass

su jclass
#test it out permissions worked
```

VII.

```
cd hw3playgroud
chmod 1777 tmp

test it out
```

VIII.

```
Incomplete
```

IX.

The only relevant file the the perl file which is on here.

X.

Difficulties: I had to writte everything all over again because my laptop turned off and nothing saved. I sent you an email please check it out thank you.

I had to redo everything from scratch.

XI.

Feedback: The assingment was easy but the time it takes to do is can be unreal. Took me all day to do. Highly interesting I enjoed it even thought I it bothered me that I had to redo more then

hald of my assingment. About 18 hours spent in this assingment technically more because I had to do almost everything twice.