Linux Commands - Part II

1. Elevate your user access to root;

```
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.90.1-microsoft-standard-WSL2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

This message is shown once a day. To disable it please create the
/home/andrijanasharkoska/.hushlogin file.
andrijanasharkoska@Andrijana:~$ sudo -i
[sudo] password for andrijanasharkoska:
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.90.1-microsoft-standard-WSL2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

This message is shown once a day. To disable it please create the
/root/.hushlogin file.
root@Andrijana:~#
```

sudo -i –> to elevate access to the root user or also known as the superuser

2. Add a new user to your Linux OS and set a password for it;

```
This message is shown once a day. To disable it please create the
/root/.hushlogin file.
root@Andrijana:~# useradd seconduser
root@Andrijana:~# passwd seconduser
New password:
Retype new password:
passwd: password updated successfully
root@Andrijana:~#
```

With the command ***useradd <user name>*** we can add a new user
passwd <user name> → to set up the password for the new user

3. Test if you can log in using that user;

```
andrijanasharkoska@Andrijana:~$ su seconduser
Password:
$ whoami
seconduser
$
```

su <user name>→ to switch account and log in with another user account
whoami → to check if I've switched to the other user account

4. Using grep command check if the user is created;

```
andrijanasharkoska@Andrijana:~$ grep seconduser /etc/passwd
seconduser:x:1001:1001::/home/seconduser:/bin/sh
```

- From what I've researched, each created user is stored in the ***etc/passwd*** text file, which stores essential information for login.
- Thus, I used the ***grep <user name>/etc/passwd*** command to search for the user

5. grep the UID of each user;

```
andrijanasharkoska@Andrijana:~$ id | grep id
uid=1000(andrijanasharkoska) gid=1000(andrijanasharkoska) groups=1000(andrijanasharkoska),4(adm),20(dialout),24(cdrom)
5(floppy),27(sudo),29(audio),30(dip),44(video),46(plugdev),116(netdev),117(docker)
```

- In this case, it matches the "id" string in the output, however, I still got the ID number of the user

```
andrijanasharkoska@Andrijana:~$ id seconduser
uid=1001(seconduser) gid=1001(seconduser) groups=1001(seconduser),17(newgroup),120(group1)
```

```
andrijanasharkoska@Andrijana:~$ id thirduser
uid=1002(thirduser) gid=1002(thirduser) groups=1002(thirduser),17(newgroup),120(group1)
```

- And I tried this approach, using only the **grep** command with the *username* and the **/etc/passwd** command (not sure if it's the right approach)

```
andrijanasharkoska@Andrijana:~$ grep andrijanasharkoska /etc/passwd
andrijanasharkoska:x:1000:1000:,,,:/home/andrijanasharkoska:/bin/bash
andrijanasharkoska@Andrijana:~$ grep seconduser /etc/passwd
seconduser:x:1001:1001::/home/seconduser:/bin/tcsh
andrijanasharkoska@Andrijana:~$ grep thirduser /etc/passwd
thirduser:x:1002:1002::/home/thirduser:/bin/sh
andrijanasharkoska@Andrijana:~$
```

6. Find out the GID of the created user;

```
andrijanasharkoska@Andrijana:~$ id -g seconduser
1001
```

id -g <user name>→ finds the GID of the created user

7. Change the password of the user and force it to change the pass on his next login;

```
andrijanasharkoska@Andrijana:~$ sudo -i
[sudo] password for andrijanasharkoska:
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.90.1-microsoft-standard-WSL2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage


This message is shown once a day. To disable it please create the
/root/.hushlogin file.
root@Andrijana:~# passwd --expire seconduser
passwd: password expiry information changed.
root@Andrijana:~# chage -l seconduser
Last password change                                    : password must be changed
Password expires                                        : password must be changed
Password inactive                                       : password must be changed
Account expires                                         : never
Minimum number of days between password change          : 0
Maximum number of days between password change          : 99999
Number of days of warning before password expires       : 7
root@Andrijana:~#
```

- I had to elevate the access to root so I could change the password for the created user. First I had to set its expiration, so the next time the user logs in, will be forced to change the password.

passwd –expire <user name> → to set expiration on the password
chage -l <user name> → to check the password expiry information

```
root@Andrijana:~# su seconduser
You are required to change your password immediately (administrator enforced).
Changing password for seconduser.
Current password:
New password:
Retype new password:
$
```

- The next time the user tries to log in, a prompt will appear that they have to change the password immediately

8. Add a new user and set an expiration date for it, with a five-day warning period;

```
andrijanasharkoska@Andrijana:~$ sudo -i
[sudo] password for andrijanasharkoska:
root@Andrijana:~# useradd thirduser
root@Andrijana:~# passwd thirduser
New password:
Retype new password:
passwd: password updated successfully
root@Andrijana:~#
```

```
andrijanasharkoska@Andrijana:~$ sudo -i
[sudo] password for andrijanasharkoska:
root@Andrijana:~# useradd thirduser
root@Andrijana:~# passwd thirduser
New password:
Retype new password:
passwd: password updated successfully
root@Andrijana:~# sudo -e 2023-3-13 thirduser
sudo: 2023-3-13 unchanged
sudo: thirduser unchanged
root@Andrijana:~# sudo useradd -e 2023-3-14 thirduser
useradd: user 'thirduser' already exists
root@Andrijana:~# sudo usermod -e 2023-3-14 thirduser
root@Andrijana:~# sudo chage -l thirduser
Last password change                                    : Mar 08, 2023
Password expires                                        : never
Password inactive                                       : never
Account expires                                         : Mar 14, 2023
Minimum number of days between password change          : 0
Maximum number of days between password change          : 99999
Number of days of warning before password expires       : 7
root@Andrijana:~# sudo chage thirduser --warndays 5
root@Andrijana:~# sudo chage -l thirduser
Last password change                                    : Mar 08, 2023
Password expires                                        : never
Password inactive                                       : never
Account expires                                         : Mar 14, 2023
Minimum number of days between password change          : 0
Maximum number of days between password change          : 99999
Number of days of warning before password expires       : 5
root@Andrijana:~#
```

- I was not sure whether the expiration was pertaining to the user account or the password, so I set up both. You can see with the ***sudo chage -l <user name>*** command the user information, moreover, the expiration policy, and when the account and the password will expire.

9. Create a new group;

```
root@Andrijana:~# groupadd -g 17 newgroup
root@Andrijana:~#
```

groupadd -g <ID of the group> <name of the group> → adding a new group

10. Assign the two new users to that group;

```
root@Andrijana:~# sudo usermod -a -G newgroup seconduser
root@Andrijana:~# sudo usermod -a -G newgroup thirduser
root@Andrijana:~# getent group newgroup
newgroup:x:17:seconduser,thirduser
root@Andrijana:~#
```

sudo usermod -a -G <group name> <user name> → to add an existing user to another group
getent group <group name> → to check whether the users have been added to the group

11. Lock one of the user accounts;

```
root@Andrijana:~# passwd -l thirduser
```

```
root@Andrijana:~# passwd --status thirduser
thirduser L 03/08/2023 0 99999 5 -1
```

passwd -l <user name> → to lock user's account
passwd –status <user name> → check whether the account was locked --> L is showing that the locking was successful.

12. Change the shell of one user to tcsh;

```
root@Andrijana:~# cat /etc/shells
# /etc/shells: valid login shells
/bin/sh
/bin/bash
/usr/bin/bash
/bin/rbash
/usr/bin/rbash
/usr/bin/sh
/bin/dash
/usr/bin/dash
/usr/bin/tmux
/usr/bin/screen
root@Andrijana:~# grep seconduser /etc/passwd
seconduser:x:1001:1001::/home/seconduser:/bin/sh
root@Andrijana:~# chsh --shell /bin/tcsh seconduser
chsh: Warning: /bin/tcsh does not exist
root@Andrijana:~#
```

- First, I am listing the available shells, and I need to have root access to be able to see the information
- Then I tried to change the shell to tcsh to one of the users, and I got a warning that it does not exist. The command I used for changing the shell is ***chsh –shell /bin/tcsh***

- By doing a bit of a research, I found out that I can install the **tcsh** shell in my Linux environment by typing tcsh into the terminal. However, that wasn't the case, and I had to install it via the ***apt install tcsh*** command. Obviously, I had to do more extensive research, or maybe not in this case, since I got the answer right in the terminal (in front of my eyes)

```
root@Andrijana:~# tcsh
Command 'tcsh' not found, but can be installed with:
apt install tcsh
root@Andrijana:~# apt install tcsh
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  tcsh
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 422 kB of archives.
After this operation, 1351 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy/universe amd64 tcsh amd64 6.21.00-1.1 [422 kB]
Fetched 422 kB in 10s (43.3 kB/s)
Selecting previously unselected package tcsh.
(Reading database ... 24593 files and directories currently installed.)
Preparing to unpack .../tcsh_6.21.00-1.1_amd64.deb ...
Unpacking tcsh (6.21.00-1.1) ...
Setting up tcsh (6.21.00-1.1) ...
update-alternatives: using /bin/tcsh to provide /bin/csh (csh) in auto mode
Processing triggers for man-db (2.10.2-1) ...
root@Andrijana:~#
```

- ***chsh –shell /bin/<name of shell> <name of user>*** → to change the shell
- Finally, I was able to change the shell and check whether it was really changed

```
root@Andrijana:~# chsh --shell /bin/tcsh seconduser
root@Andrijana:~# grep seconduser /etc/passwd
seconduser:x:1001:1001::/home/seconduser:/bin/tcsh
root@Andrijana:~#
```

==========

13. Make sure your home directory has "execute" access enabled for group and other.

```
andrijanasharkoska@Andrijana:~$ chmod go+x home
```

chmod g0+x home → created a home directory and set up the "execute" access with the ***x*** and ***go*** is used for specifying that the access is for the group and others

14. Change to your home directory, and create a directory called labs;

```
andrijanasharkoska@Andrijana:~$ cd home
andrijanasharkoska@Andrijana:~/home$ mkdir labs
```

15. Create an empty file in labs directory

```
andrijanasharkoska@Andrijana:~/home$ cd labs
andrijanasharkoska@Andrijana:~/home/labs$ touch myfile.txt
```

16. Change permissions of file to rwx-rwx-rwx

```
andrijanasharkoska@Andrijana:~/home/labs$ chmod 777 myfile.txt
```

chmod 777 <filename> → changes the file permissions to rwx-rwx-rwx

17. List the file. What color is the file?

```
andrijanasharkoska@Andrijana:~/home/labs$ chmod 777 myfile.txt
andrijanasharkoska@Andrijana:~/home/labs$ ls -l
total 0
-rwxrwxrwx 1 andrijanasharkoska andrijanasharkoska 0 Mar  9 01:25 myfile.txt
andrijanasharkoska@Andrijana:~/home/labs$
```

- The color of the file is green

18. Change the permissions back to rx-rw-rw

```
andrijanasharkoska@Andrijana:~/home/labs$ chmod 566 myfile.txt
andrijanasharkoska@Andrijana:~/home/labs$ ls -l
total 0
-r-xrw-rw- 1 andrijanasharkoska andrijanasharkoska 0 Mar  9 01:25 myfile.txt
andrijanasharkoska@Andrijana:~/home/labs$
```

- I used this table as a reference when typing the numbers for changing the permissions

r/w/x	binary	octal
---	000	0
--x	001	1
-w-	010	2
-wx	011	3
r--	100	4
r-x	101	5
rw-	110	6
rwx	111	7

19. Check what owners does the file have.

```
andrijanasharkoska@Andrijana:~/home/labs$ ls -l myfile.txt
-r-xrw-rw- 1 andrijanasharkoska andrijanasharkoska 0 Mar  9 01:25 myfile.txt
```

- My username shows as the owner of the file (andrijanasharkoska)

20. Change the user ownership of the file to another user;

```
andrijanasharkoska@Andrijana:~/home/labs$ chown seconduser myfile
chown: cannot access 'myfile': No such file or directory
andrijanasharkoska@Andrijana:~/home/labs$ chown seconduser myfile.txt
chown: changing ownership of 'myfile.txt': Operation not permitted
andrijanasharkoska@Andrijana:~/home/labs$ sudo chown seconduser myfile.txt
[sudo] password for andrijanasharkoska:
andrijanasharkoska@Andrijana:~/home/labs$ ls -l
total 0
-r-xrw-rw- 1 seconduser andrijanasharkoska 0 Mar  9 01:25 myfile.txt
andrijanasharkoska@Andrijana:~/home/labs$
```

- First, I used the ***chown <user name> <file name>*** command to change the user, however, the operation was not permitted
- Thus, upon further research, I found out I can use the keyword **sudo** before since the **sudo/root** user has the permission to change system settings like changing ownership, adding or removing users, etc.
- **sudo chown <user name> <file name>** → to change the ownership of the file
- **ls -l** to list the files in order to verify the owner was actually changed

21. Create a group called group1 and assign two users to the group;

```
root@Andrijana:~# groupadd -g 120 group1
root@Andrijana:~# sudo usermod -a -G group1 seconduser
root@Andrijana:~# sudo usermod -a -G group1 thirduser
```

- I've added the existing group members for the sake of not creating additional users

22. Create a file called group1.txt and redirect below input into the file: "This is our group test file".

```
root@Andrijana:~# touch group1.txt
root@Andrijana:~# nano group1.txt
root@Andrijana:~# cat group1.txt
"This is our group test file".
```

23. Change the group of the file to one of your users;0

```
root@Andrijana:~# chown :seconduser group1.txt
root@Andrijana:~# ls -l
total 4
-rw-r--r-- 1 root seconduser 31 Mar  9 01:53 group1.txt
```

chown :<user name> <file name> → to change the group of the file to one of the existing users

- Then I am listing the file information to confirm the changes have taken place

24. Give members of the group group1 read/write access to this file?

```
Try 'chmod --help' for more information.
root@Andrijana:~# chmod 006 group1.txt
root@Andrijana:~# ls -l
total 4
-------rw- 1 666 seconduser 31 Mar  9 01:53 group1.txt
```

Read by owner	400
Write by owner	200
Execute by owner	100
Read by group	040
Write by group	020
Execute by group	010
Read by others	004
Write by others	002
Execute by others	001

Sum all the accesses you wish to permit. For example, to give write and execute privileges to the owner of `grop1.txt`(200+100=300)

- In this case, we sum up the read by others and write by others (002 + 004 - 006) and get the **rw** permissions for the users.

0Other is **everyone that is not the owner or in the group**. For example, if you have a file that is root:root then root is the owner, users/processes in the root group have group permissions, and you are treated as other.