

Task.Linux.2

Pashynskyi Maxim

- 1) Analyze the structure of the `/etc/passwd` and `/etc/group` file, what fields are present in it, what users exist on the system? Specify several pseudo-users, how to define them?

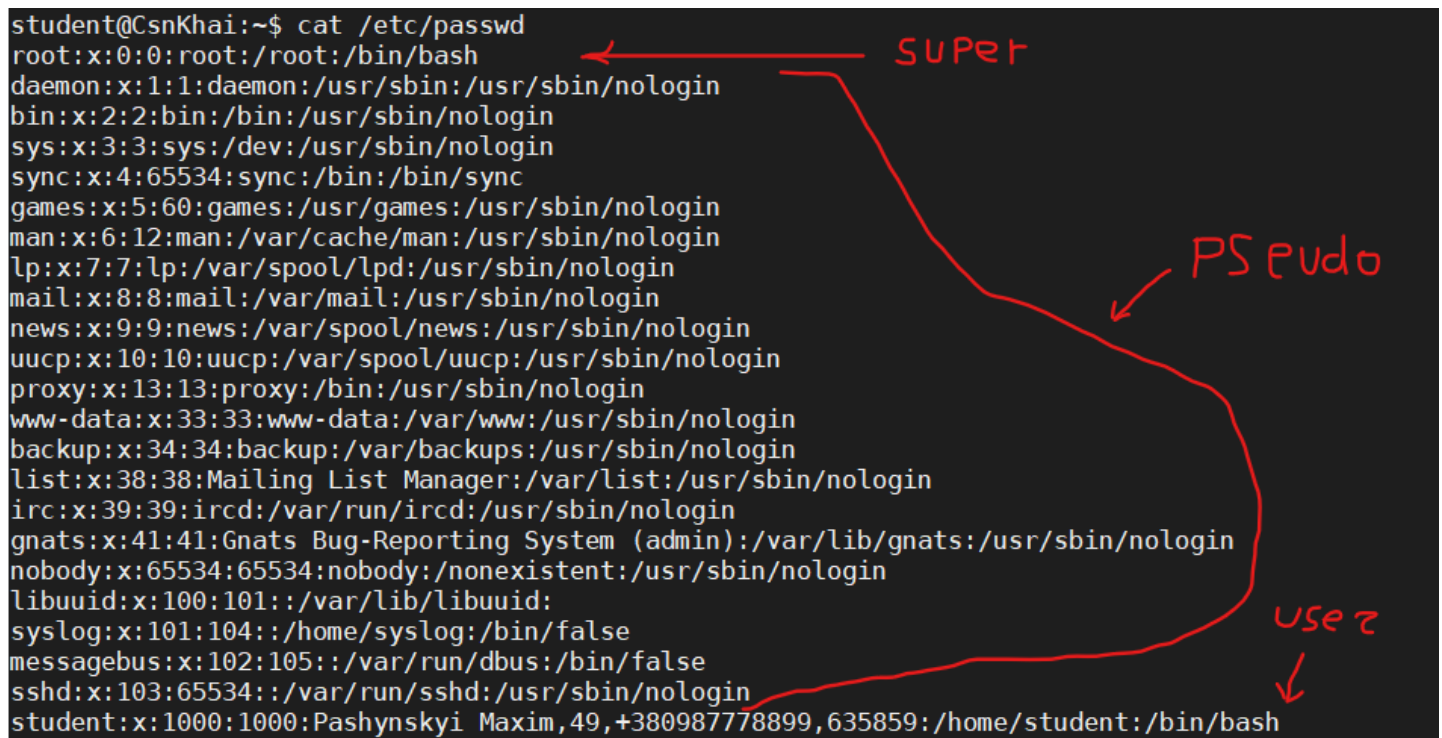
`/etc/passwd` contains information about users in the system and their characteristics. The structure of the user info row in `/etc/passwd` file looks like this:

username:password(x):UID:GID:GECOS personal info:home directory:shell

The `/etc/group` file stores group information. Each line in the file represents a group and is divided into fields:

group name:password(x):GID:users in group

Pseudo-users are users that serve specific purposes and typically don't have interactive login access. They are used to manage system services and other tasks.



A terminal window showing the contents of the `/etc/passwd` file. The output lists system users and services. Handwritten red annotations are present: an arrow points from the word "SUPER" to the `root` entry; another arrow points from the word "PSEUDO" to a bracketed group of users including `daemon`, `bin`, `sys`, `sync`, `games`, `man`, `lp`, `mail`, `news`, `uucp`, `proxy`, `www-data`, `backup`, `list`, `irc`, `gnats`, `nobody`, and `libuuid`; a third arrow points from the word "USER" to the `student` entry at the bottom.

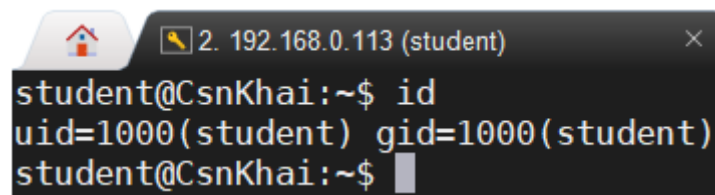
```
student@CsnKhai:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
libuuid:x:100:101::/var/lib/libuuid:
syslog:x:101:104::/home/syslog:/bin/false
messagebus:x:102:105::/var/run/dbus:/bin/false
sshd:x:103:65534::/var/run/sshd:/usr/sbin/nologin
student:x:1000:1000:Pashynskyi Maxim,49,+380987778899,635859:/home/student:/bin/bash
```

- 2) What are the uid ranges? What is UID? How to define it?

UID - user identifier, numeric value assigned to each user in the system.

The common **UID ranges** are 0-999 for root and pseudo-users and 1000+ for normal users.

To see current user UID we can use such command:



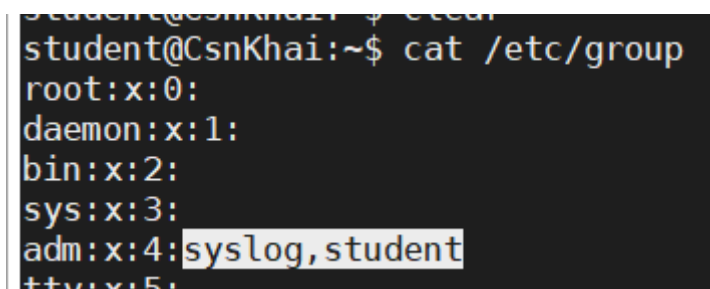
```
student@CsnKhai:~$ id
uid=1000(student) gid=1000(student)
student@CsnKhai:~$
```

3) What is GID? How to define it?

GID - group identifier, numeric value assigned to each group in the system. When a user is created, it is automatically assigned to a group of the same name with a specific **GID**, equal to **UID** by default.

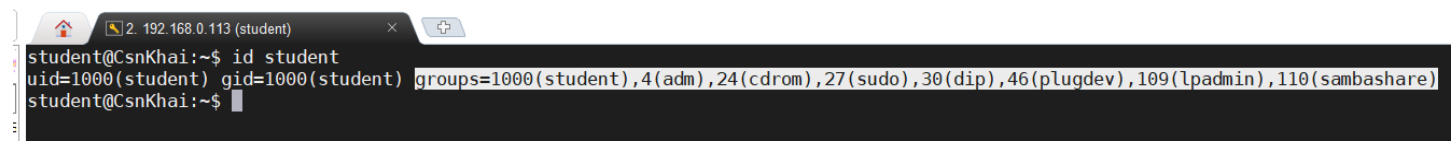
4) How to determine the belonging of a user to a specific group?

Open `/etc/group` file:



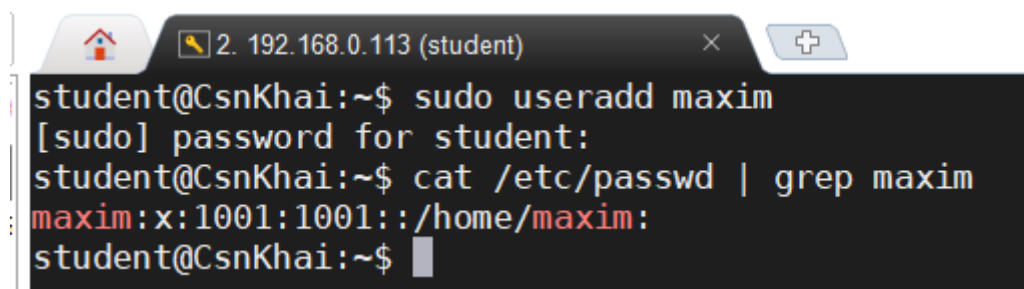
```
student@CsnKhai:~$ cat /etc/group
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:syslog,student
tty:x:5:
```

or use `id` command:



```
student@CsnKhai:~$ id student
uid=1000(student) gid=1000(student) groups=1000(student),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),109(lpadmin),110(sambashare)
student@CsnKhai:~$
```

5) What are the commands for adding a user to the system? What are the basic parameters required to create a user?



```
student@CsnKhai:~$ sudo useradd maxim
[sudo] password for student:
student@CsnKhai:~$ cat /etc/passwd | grep maxim
maxim:x:1001:1001::/home/maxim:
student@CsnKhai:~$
```

6) How do I change the name (account name) of an existing user?

```
student@CsnKhai:~$ sudo usermod -l new_maxim maxim
student@CsnKhai:~$ cat /etc/passwd | grep maxim
new_maxim:x:1001:1001:~/home/maxim:
student@CsnKhai:~$
```

7) What is `skell_dir`? What is its structure?

The *skel* is a template directory containing default config files and user profile settings.

When a new user is created, the contents of the *skel* directory (`/etc/skel`) are copied into the user's home directory. This ensures that the new user starts with a consistent and predefined environment. Users can then customize their settings as needed.

```
student@CsnKhai:~$ ll /etc/skel/
total 20
drwxr-xr-x  2 root root 4096 Sep 15 2015 ./
drwxr-xr-x 83 root root 4096 Aug 17 12:29 ../
-rw-r--r--  1 root root  220 Apr  9 2014 .bash_logout
-rw-r--r--  1 root root 3637 Apr  9 2014 .bashrc
-rw-r--r--  1 root root  675 Apr  9 2014 .profile
student@CsnKhai:~$
```

.bash_logout - commands to run when user logs out;

.bashrc - sets up user shell environment;

.profile - sets up user environment variables and settings.

8) How to remove a user from the system (including his mailbox)?

```
root@CsnKhai:/home/student# userdel -r maxim
userdel: maxim mail spool (/var/mail/maxim) not found
root@CsnKhai:/home/student#
```

9) What commands and keys should be used to lock and unlock a user account?

to lock:

```
root@CsnKhai:/home/student# passwd -l maxim
passwd: password expiry information changed.
root@CsnKhai:/home/student# cat /etc/shadow | grep maxim
maxim:!!$6$loFTn2Az$byVzKUQuP/9h9Xqb90BuuAWt4AgV6Gg8NCvhJILU!tao
root@CsnKhai:/home/student#
```

to unlock:

```
root@CsnKhai:/home/student# passwd -u maxim
passwd: password expiry information changed.
root@CsnKhai:/home/student# cat /etc/shadow | grep maxim
maxim:$6$loFTn2Az$byVzKUQuP/9h9Xqb90BuuAWt4AgV6Gg8NCvhJILU!tao
root@CsnKhai:/home/student#
```

10) How to remove a user's password and provide him with a password-free login for subsequent password change?

to remove password:

```
student@CsnKhai:~$ sudo passwd -d maxim
passwd: password expiry information changed.
student@CsnKhai:~$ sudo cat /etc/shadow | grep maxim
maxim::19586:0:99999:7:::
student@CsnKhai:~$
```

to expire password and force user to change it upon the next login:

```
student@CsnKhai:~$ sudo passwd -e maxim
passwd: password expiry information changed.
student@CsnKhai:~$ su maxim
Password:
su: Authentication failure
student@CsnKhai:~$ su maxim
Password:
You are required to change your password immediately (root enforced)
Changing password for maxim.
(current) UNIX password: 
```

11) Display the extended format of information about the directory, tell about the information columns displayed on the terminal.

```
student@CsnKhai:/etc$ ls -l
total 740
-rw-r--r-- 1 root root 2981 Sep 15 2015 adduser.conf
drwxr-xr-x 2 root root 4096 Aug 15 20:02 alternatives
drwxr-xr-x 3 root root 4096 Sep 15 2015 apm
drwxr-xr-x 3 root root 4096 Sep 15 2015 apparmor
drwxr-xr-x 8 root root 4096 Sep 15 2015 apparmor.d
drwxr-xr-x 6 root root 4096 Sep 15 2015 apt
-rw-r--r-- 1 root root 2177 Apr 9 2014 bash.bashrc
-rw-r--r-- 1 root root 45 Mar 22 2014 bash_completion
drwxr-xr-x 2 root root 4096 Sep 15 2015 bash_completion.d
-rw-r--r-- 1 root root 356 Jan 1 2012 bindresvport.blacklist
```

type/permissions|num of links|owner|group|size|modification time|name

12) What access rights exist and for whom (i. e., describe the main roles)? Briefly describe the acronym for access rights.

Access rights:

- r - read permission. Allows to view the content of a file or list the contents of a directory;
- w - write permission. Allows users to modify the content of a file or create, modify, and delete files in a directory;
- x - execute permission. For files, it allows users to execute the file as a program if it's executable. For directories, it allows users to access the contents of the directory.

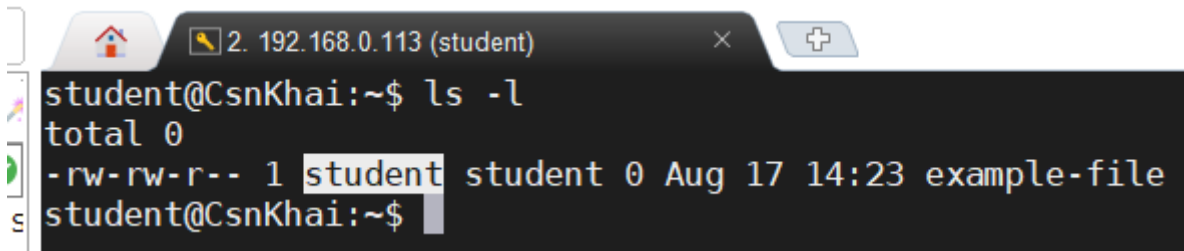
There are 3 main roles: owner (u), group (g) and others (o).

Basic set of permissions to a file looks like this:

rwX(u)rwX(g)rwX(o)

13) What is the sequence of defining the relationship between the file and the user?

First we need to define a file owner:

A terminal window titled '2. 192.168.0.113 (student)' showing the command 'ls -l' being executed. The output shows a file named 'example-file' owned by 'student' with permissions '-rw-rw-r--'.

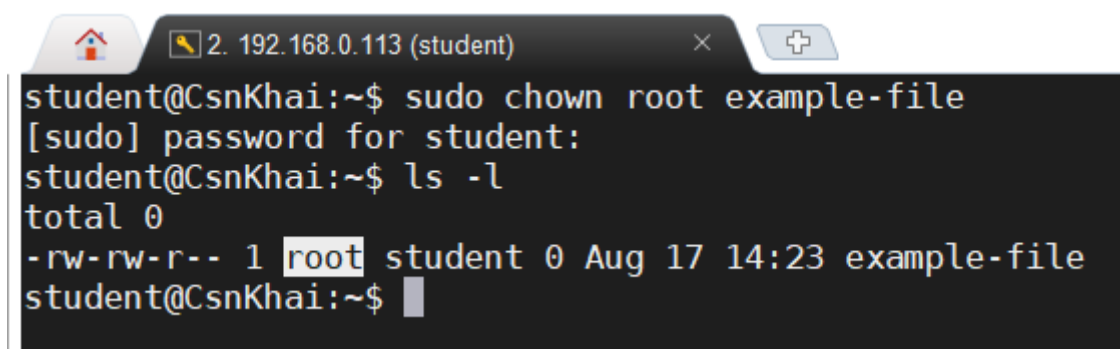
```
student@CsnKhai:~$ ls -l
total 0
-rw-rw-r-- 1 student student 0 Aug 17 14:23 example-file
student@CsnKhai:~$
```

Then we need to look at the set of permissions for the owner user. In this example the owner has privileges to read file content and to modify it, without permission to execute.

Other users have permission only to read that file.

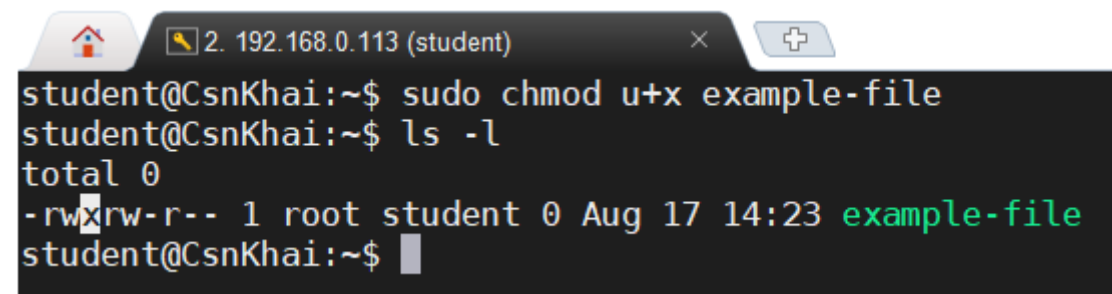
14) What commands are used to change the owner of a file (directory), as well as the mode of access to the file? Give examples, demonstrate on the terminal.

to change owner:

A terminal window titled '2. 192.168.0.113 (student)' showing the command 'sudo chown root example-file' being executed. The prompt asks for the password for 'student'. After running 'ls -l', the output shows 'example-file' is now owned by 'root' with permissions '-rw-rw-r--'.

```
student@CsnKhai:~$ sudo chown root example-file
[sudo] password for student:
student@CsnKhai:~$ ls -l
total 0
-rw-rw-r-- 1 root student 0 Aug 17 14:23 example-file
student@CsnKhai:~$
```

to change permissions:

A terminal window titled '2. 192.168.0.113 (student)' showing the command 'sudo chmod u+x example-file' being executed. After running 'ls -l', the output shows 'example-file' is owned by 'root' and has permissions '-rwxr--r--'.

```
student@CsnKhai:~$ sudo chmod u+x example-file
student@CsnKhai:~$ ls -l
total 0
-rwxr--r-- 1 root student 0 Aug 17 14:23 example-file
student@CsnKhai:~$
```

15) What is an example of octal representation of access rights?
Describe the umask command.

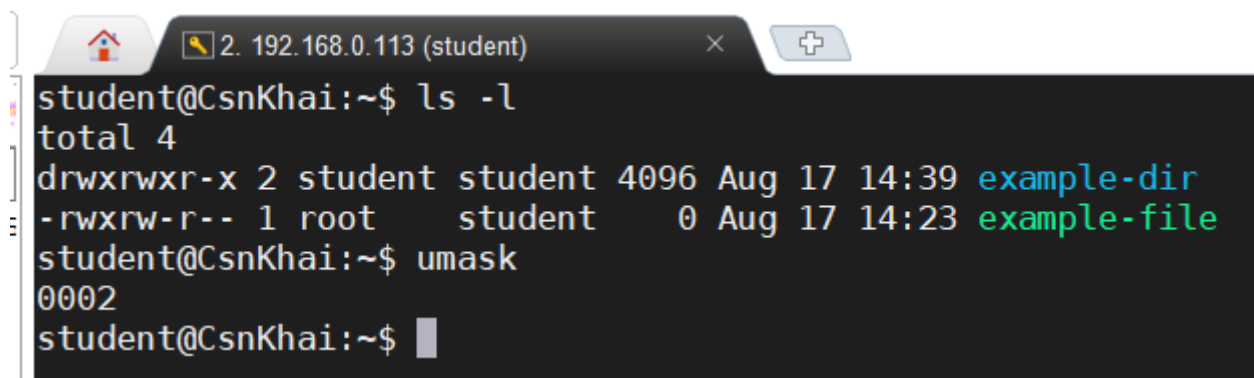
In octal representation of permissions values from 0 to 7 equals a set of rwX permissions for a single role. For example:

4 = r--, 7 = rwx, etc.

Full set of permissions will look like this in octal:

rw-rw-r = 664

umask command defines a default set of permissions associated with newly created files and directories.

A terminal window titled '2. 192.168.0.113 (student)' showing a user named 'student' at a host 'CsnKhai'. The user runs 'ls -l' and the output shows two items: 'example-dir' with permissions 'drwxrwxr-x' and 'example-file' with permissions '-rwxrw-r--'. Then the user runs 'umask' and the output is '0002'.

```
student@CsnKhai:~$ ls -l
total 4
drwxrwxr-x 2 student student 4096 Aug 17 14:39 example-dir
-rwxrw-r-- 1 root    student  0 Aug 17 14:23 example-file
student@CsnKhai:~$ umask
0002
student@CsnKhai:~$
```

As we can see on this screenshot, default perms for files are 664, and default perms for directories are 775.

16) Give definitions of sticky bits and mechanism of identifier substitution. Give an example of files and directories with these attributes.

Sticky bit (t) - permission attribute that can be set on directories. When the sticky bit is set on a directory, only the owner of a file within that directory and the directory's owner can delete or rename the file.

When the **SUID (u+s)** bit is set on an executable file, it allows the user who executes the file to temporarily gain the permissions of the file's owner.

When the **SGID (g+s)** bit is set on an executable file or directory, it allows users who execute the file or create files within the directory to gain the group ownership of the file or directory's group.


```
student@CsnKhai:~$ ls -l
total 4
drwxrwxr-t 2 student student 4096 Aug 17 14:39 example-dir
-rwsrw-r-- 1 root      student  0 Aug 17 14:23 example-file
-rw-rwsr-- 1 student  student  0 Aug 17 14:54 sgid-file
student@CsnKhai:~$
```

17) What file attributes should be present in the command script?

Execute (x) permission should be present for sure and SUID or SGID according to the situation.