

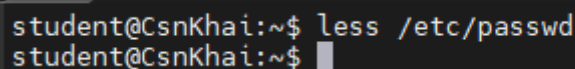
Struna V.R.

Task №2

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?

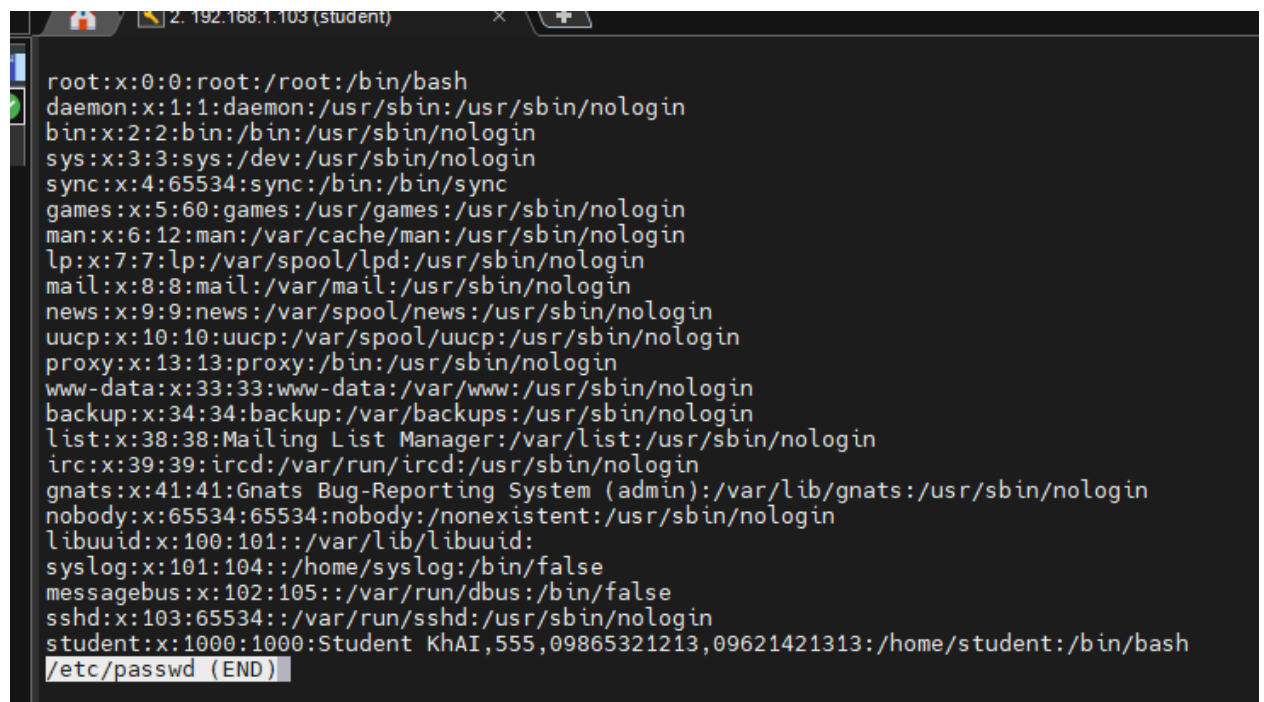
The `/etc/passwd` file stores information about all registered users.

The `/etc/group` file contains the names of the groups present in Linux and lists the members of each group.



```
student@CsnKhai:~$ less /etc/passwd
student@CsnKhai:~$
```

less /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:Student KhAI,555,09865321213,09621421313:/home/student:/bin/bash
/etc/passwd (END)
```

The records contain:

- Username;
- Encrypted password;
- UID (Digital User ID);
- GID (Digital User Group Identifier);
- GEGOS (Full username);
- User's home directory and login shell.

```
student@CsnKhai:~$ less /etc/group
```

less /etc/group

```
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:syslog,student
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
uucp:x:10:
man:x:12:
proxy:x:13:
kmem:x:15:
dialout:x:20:
fax:x:21:
voice:x:22:
cdrom:x:24:student
floppy:x:25:
tape:x:26:
sudo:x:27:student
audio:x:29:
dip:x:30:student
www-data:x:33:
backup:x:34:
operator:x:37:
list:x:38:
irc:x:39:
src:x:40:
gnats:x:41:
shadow:x:42:
utmp:x:43:
video:x:44:
sasl:x:45:
plugdev:x:46:student
staff:x:50:
games:x:60:
users:x:100:
nogroup:x:65534:
libuuid:x:101:
netdev:x:102:
crontab:x:103:
syslog:x:104:
messagebus:x:105:
fuse:x:106:
mlocate:x:107:
ssh:x:108:
student:x:1000:
lpadmin:x:109:student
sambashare:x:110:student
/etc/group (END)
```

The records contain:

- Group name - by default, when a new user is created, his group is also created with the same name as the user's registration name;
- An encrypted password or an x to indicate the use of the / etc / gshadow file;
- Group ID - (number) list of members, separated by commas without spaces. It is important here: a user with the same name as the group name can exist in the system and be a member of this group, but not be listed in this list.

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

uid - unique identifier of the user within the system.

uid is a simple numeric designation for an individual user. This is usually a positive number not more

than 65535 (sometimes 32-bit). Some identifiers are reserved for special use. These include 0 (root),

1-999(daemons, pseudo-users, system and reserved users), 1000+ (regular users).

The UID can be determined via the **id** command

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

3) What is GID? How to define it?

gid - unique identifier of the group within the system to which the user belongs

id -g

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

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

id -Gn

```
student
student@CsnKhai:~$ id -Gn
student adm cdrom sudo dip plugdev lpadmin 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?

sudo useradd test

sudo adduser test

username is the login name of the user. This is the only required parameter in all commands.

uid comment is an additional comment about the user with the specified name.

dir - indicates to the user's home directory.

expire - indicates the exact date until registration record are available .

inactive - indicates a continuous number of days without registration in the system before this record is

blocked.

gid - defines the id or name of the group to which the user belongs.

new_username - Replaces the old login name.

shell - defines the shell for the command interpreter for the given user.

skel_dir - contains files which must be copied to the new user's home directory.

uid is the unique user identifier associated with this name.

-m - indicates need to create a new home directory (useradd) or move the current one to a new location

(usermod).

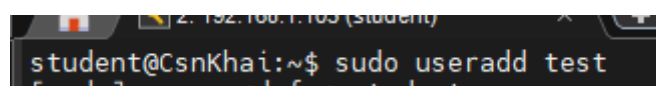
-o - Allows repeating the same user ID.

-g - Select the main group for the login name.

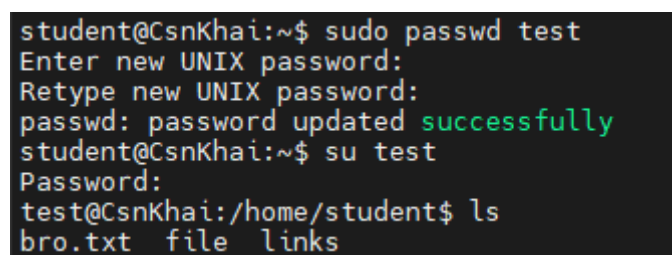
-G - selects additional groups.

-r - Tells the user's home directory to be moved. If the home directory for the registration entry is out of

date, existing files will be migrated to the new directory.



```
student@CsnKhai:~$ sudo useradd test
```



```
student@CsnKhai:~$ sudo passwd test
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
student@CsnKhai:~$ su test
Password:
test@CsnKhai:/home/student$ ls
bro.txt  file  links
```

```

student@CsnKhai:~$ sudo adduser test1
Adding user `test1' ...
Adding new group `test1' (1002) ...
Adding new user `test1' (1002) with group `test1' ...
Creating home directory `/home/test1' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for test1
Enter the new value, or press ENTER for the default
    Full Name []: vik
    Room Number []: 14
    Work Phone []: 24124121241
    Home Phone []: 2512312341
    Other []: 2312
Is the information correct? [Y/n] y
student@CsnKhai:~$

```

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

usermod -l newname oldname

```

root@CsnKhai:~# usermod -l test1 test

```

7) What is skell_dir? What is its structure?

The /etc/skel directory contains copies of various installation and other files that can be copied to new user home directories when the useradd program adds the user.

```

root@CsnKhai:~# ls -l /etc/skel/
total 0
root@CsnKhai:~#

```

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

sudo deluser --remove-all-files user

```

student@CsnKhai:~$ clear
student@CsnKhai:~$ sudo deluser --remove-all-files test
[sudo] password for student:
Looking for files to backup/remove ...
/usr/sbin/deluser: Cannot handle special file
Removing user `test' ...
Warning: group `test' has no more members.
Done.

```

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

To block or unblock an account, use the **passwd** command with certain keys:

-l prevent the user from logging in.

-S display account information.

- u - cancels the effect of the -l parameter.

Second character means opened it or locked(P – open; L - locked)

```
student@CsnKhai:~$ sudo passwd -S test1
test1 P 12/23/2021 0 99999 7 -1
student@CsnKhai:~$ sudo passwd -l test1
passwd: password expiry information changed.
student@CsnKhai:~$ sudo passwd -S test1
test1 L 12/23/2021 0 99999 7 -1
student@CsnKhai:~$
student@CsnKhai:~$ sudo passwd -u test1
passwd: password expiry information changed.
student@CsnKhai:~$ sudo passwd -S test1
test1 P 12/23/2021 0 99999 7 -1
student@CsnKhai:~$
```

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

sudo passwd -d user

sudo usermod -s /usr/sbin/nologin user

sudo chage -l user

```
student@CsnKhai:~$ sudo passwd -d test1
passwd: password expiry information changed.
student@CsnKhai:~$ sudo usermod -s /usr/sbin/nologin test1
usermod: no changes
student@CsnKhai:~$ sudo chage -l test1
Last password change                : Dec 23, 2021
Password expires                    : never
Password inactive                    : never
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
student@CsnKhai:~$
```

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

ls -l /etc/ | less

```
student@CsnKhai:~$ ls -l /etc/ | less
```

```

total 740
-rw-r--r-- 1 root root 2981 Sep 15 2015 adduser.conf
drwxr-xr-x 2 root root 4096 Sep 15 2015 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
-rw-r--r-- 1 root root 321 Apr 16 2014 blkid.conf
lrwxrwxrwx 1 root root 15 Aug 5 2015 blkid.tab -> /dev/.blkid.tab
drwxr-xr-x 3 root root 4096 Sep 15 2015 ca-certificates
-rw-r--r-- 1 root root 7773 Sep 15 2015 ca-certificates.conf
drwxr-xr-x 2 root root 4096 Sep 15 2015 calendar
drwxr-s--- 2 root dip 4096 Sep 15 2015 chatscripts
drwxr-xr-x 2 root root 4096 Sep 15 2015 console-setup
drwxr-xr-x 2 root root 4096 Sep 15 2015 cron.d
drwxr-xr-x 2 root root 4096 Sep 15 2015 cron.daily
drwxr-xr-x 2 root root 4096 Sep 15 2015 cron.hourly
drwxr-xr-x 2 root root 4096 Sep 15 2015 cron.monthly
-rw-r--r-- 1 root root 722 Feb 9 2013 crontab
drwxr-xr-x 2 root root 4096 Sep 15 2015 cron.weekly
drwxr-xr-x 4 root root 4096 Sep 15 2015 dbus-1
-rw-r--r-- 1 root root 2969 Feb 23 2014 debconf.conf
-rw-r--r-- 1 root root 11 Feb 20 2014 debian_version
drwxr-xr-x 2 root root 4096 Sep 15 2015 default
-rw-r--r-- 1 root root 604 Nov 7 2013 deluser.conf
drwxr-xr-x 2 root root 4096 Sep 15 2015 depmod.d
drwxr-xr-x 4 root root 4096 Sep 15 2015 dhcp
drwxr-xr-x 2 root root 4096 Sep 15 2015 dictionaries-common
drwxr-xr-x 2 root root 4096 Sep 15 2015 discover.conf.d
-rw-r--r-- 1 root root 346 Dec 29 2013 discover-modprobe.conf
drwxr-xr-x 4 root root 4096 Sep 15 2015 dpkg

```

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

Briefly describe the acronym for access rights.

Initially, each file had three access parameters:

- Read - allows to receive the contents of the file, but not to write. To access the list of files and directories used in it;
- Write - allows you to write new data to a file or violate, and also allows you to create and fix files and directories;
- Execution - You cannot execute a program if it does not have an execution flag. This file is installed for all programs and scripts.

Each file has three categories of users, for which you can set different combinations of access rights:

- Owner - a set of rights for the owner of the file, the user who created or is now set by its owner. Usually the owner has all the rights, read, write and execute.

- Group - any user group that exists in the system and is associated with a file. However, this can only be one group, and it is usually the owner's group, although a different group can be assigned to a file.
- Others - all users except the owners and users of the file group.

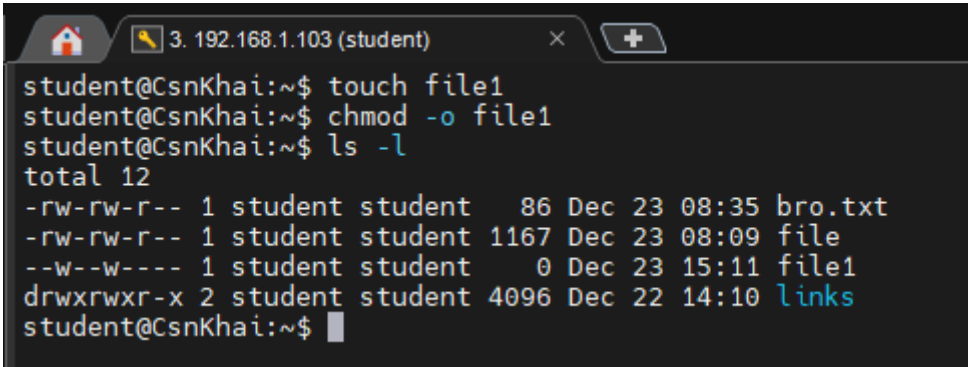
13) What is the sequence of defining the relationship between the file and the user?

The Linux operating system is a multi-user system that provides tremendous opportunities for manipulating data access for each user separately. This allows you to flexibly regulate the relationship between users, combining them into groups, which will protect the data of one user from unwanted interference from others.

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.

For example, I created file and change mode of access to it.

`chmod -o file`



```

student@CsnKhai:~$ touch file1
student@CsnKhai:~$ chmod -o file1
student@CsnKhai:~$ ls -l
total 12
-rw-rw-r-- 1 student student 86 Dec 23 08:35 bro.txt
-rw-rw-r-- 1 student student 1167 Dec 23 08:09 file
--w--w---- 1 student student 0 Dec 23 15:11 file1
drwxrwxr-x 2 student student 4096 Dec 22 14:10 links
student@CsnKhai:~$

```

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

You can use an octal value to change the permissions of a file:

- x - 1;
- w - 2;
- r - 4;
- All - 7.

`chmod 700,710,720,740 file`


```

drwxrwxr-x 2 student student 4096 Dec 22 14:10 links
student@CsnKhai:~$ chmod 700 file1
student@CsnKhai:~$ ls -l
total 12
-rw-rw-r-- 1 student student 86 Dec 23 08:35 bro.txt
-rw-rw-r-- 1 student student 1167 Dec 23 08:09 file
-rwx----- 1 student student 0 Dec 23 15:11 file1
drwxrwxr-x 2 student student 4096 Dec 22 14:10 links
student@CsnKhai:~$ chmod 740 file1
student@CsnKhai:~$ ls -l
total 12
-rw-rw-r-- 1 student student 86 Dec 23 08:35 bro.txt
-rw-rw-r-- 1 student student 1167 Dec 23 08:09 file
-rwxr----- 1 student student 0 Dec 23 15:11 file1
drwxrwxr-x 2 student student 4096 Dec 22 14:10 links
student@CsnKhai:~$ chmod 720 file1
student@CsnKhai:~$ ls -l
total 12
-rw-rw-r-- 1 student student 86 Dec 23 08:35 bro.txt
-rw-rw-r-- 1 student student 1167 Dec 23 08:09 file
-rwx-w---- 1 student student 0 Dec 23 15:11 file1
drwxrwxr-x 2 student student 4096 Dec 22 14:10 links
student@CsnKhai:~$ chmod 710 file1
student@CsnKhai:~$ ls -l
total 12
-rw-rw-r-- 1 student student 86 Dec 23 08:35 bro.txt
-rw-rw-r-- 1 student student 1167 Dec 23 08:09 file
-rwx--x--- 1 student student 0 Dec 23 15:11 file1
drwxrwxr-x 2 student student 4096 Dec 22 14:10 links
student@CsnKhai:~$ █

```

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

Sticky bit is mainly used for directories to protect files in them. From such a directory, the user can delete only those files that he owns. An example is the /tmp directory, in which the entry is open for all users, but it is undesirable to delete other people's files. The attribute is set using the chmod utility.

The octal values for SUID and SGID are 4000 and 2000.

Symbolic: u + s and g + s.

chmod 2722 file

```

drwxrwxr-x 2 student student 4096 Dec 22 14:10 links
student@CsnKhai:~$ chmod 2722 file1
student@CsnKhai:~$ ls -l
total 12
-rw-rw-r-- 1 student student 86 Dec 23 08:35 bro.txt
-rw-rw-r-- 1 student student 1167 Dec 23 08:09 file
-rwx-wS-w- 1 student student 0 Dec 23 15:11 file1
drwxrwxr-x 2 student student 4096 Dec 22 14:10 links
student@CsnKhai:~$ █

```

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

Scripts are text, a set of commands that are read by the interpreter and executed in the programming environment that the user is using.

Like any text, it can be stored in files. Files in Linux can have a flag that is responsible for the fact that the file can be run. The script itself can be written in different languages: python, PHP, or in bash itself. If the script is written in a language that the interpreter does not read by default, then it must be specified in the script itself or in the terminal line.