Task 5.2

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?

man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
etc/passwd consists of 7 fields :

testman:x:1001:1001:Testman,55,+380993737377,+37903,It is a test user:/home/testman:/bin/sh

login:password:UID:GID:GECOS:home:shell

- login = registration name
- password – x means that encrypted password hash is stored at /etc/shadow
- UID – user's id, regular users have id bigger than 500
- GID – group id, user can join several groups
- GECOS – personal info about user (phone number, address, full name, description)
- home -path to user's home catalog
- shell – command prompt interpreter, typically shell, but can be changed with chsh

/etc/group – list of users with their groups

adm:x:4:syslog,user

name:password:GID: members

Pseudo-users are created for the purpose of administering of the system and program services , for example, for instance "root" (for the superuser account) and "www-data" (for the account which owns and runs web servers). Any Linux system has these pseudo users by default Mail, news, games, Apache FTP, MySQL and sshd are related to the process of Linux system. Pseud- users usually do not need or cannot log in to the system.

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

UID ranges - set of reserved numbers that can be assigned to different kinds of users.

A UID (user identifier) is a number assigned by Linux to each user on the system.

UID 0 (zero) is reserved for the root.
UIDs 1–99 are reserved for other predefined accounts. UID
100–999 are reserved by system for administrative and system accounts/groups.
UID 1000–10000 are occupied by applications account. UID 10000+ are used for user accounts.

3) What is GID? How to define it?

GID 0 (zero) is reserved for the root group.
GID 1–99 are reserved for the system and application use.
GID 100+ allocated for the user's group.

GID - Group identifier, it is an identifier that marks that particular represents a specific group of users.

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

groups username

```
user@usersPC:~$ grep test_group /etc/group
test_group:x:1004:test_user
user@usersPC:~$ groups test_user
test_user : test_user test_group
```

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

sudo useradd username
sudo passwd username
or
sudo adduser gena

```
user@usersPC:~$ sudo adduser gena
Adding user `gena' ...
Adding new group `gena' (1006) ...
Adding new user `gena' (1005) with group `gena' ...
Creating home directory `/home/gena' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for gena
Enter the new value, or press ENTER for the default
        Full Name []: Genadiy
        Room Number []: 566
        Work Phone []: 456
        Home Phone []: 78329
        Other []: it is gena other info
Is the information correct? [Y/n] y
user@usersPC:~$
```

Basic parameters for useradd:
-d – set default home directory
-u – choose user id number for user
- g – choose group id for user
-G – add user to few groups
- m - set home directory if it does not exist
-M – set no home dir
-e – create user with his profile expiry date
-f – create user with password expiry date
-s – create user login shell

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

sudo usermod -l newname oldname

7) What is skell_dir? What is its structure?

/etc/skel is used for initiation of user’s home directory when it was first created.

```
user@usersPC:~$ ls -la /etc/skel
total 28
drwxr-xr-x   2 root root  4096 Jul 31  2020 .
drwxr-xr-x 144 root root 12288 Dec  8 09:59 ..
-rw-r--r--   1 root root   220 Feb 25  2020 .bash_logout
-rw-r--r--   1 root root  3771 Feb 25  2020 .bashrc
-rw-r--r--   1 root root   807 Feb 25  2020 .profile
```

File /etc/default/useradd defines skel directory

```
# The SKEL variable specifies the directory containing "skeletal" user
# files; in other words, files such as a sample .profile that will be
# copied to the new user's home directory when it is created.
# SKEL=/etc/skel
#
```

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

sudo userdel -r username

9) What commands and keys should be used to lock and unlock a user account?
To lock:
sudo passwd -l username or sudo usermod -L username
Unlock:
sudo passwd -u username or sudo usermod -U username

To check user for blocking:
sudo passwd -S username

```
user@usersPC:~$ sudo passwd -l gena
passwd: password expiry information changed.
user@usersPC:~$ sudo passwd --status gena
gena L 12/08/2021 0 99999 7 -1
user@usersPC:~$ sudo passwd -u gena
passwd: password expiry information changed.
user@usersPC:~$ passwd --status newuser
passwd: You may not view or modify password information for newuser.
```

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

chage -l username
or
passwd -e username

```
user@usersPC:~$ sudo passwd -e petya
passwd: password expiry information changed.
user@usersPC:~$ su petya
Password:
You are required to change your password immediately (administrator enforced)
Changing password for petya.
Current password:
```

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

```
user@pc:~/Study/Books$ ls -ilah --author ~/Study/Books
total 65M
29232640 drwxr-xr-x 7 user user user 4.0K Dec 10 16:21  .
28708066 drwxr-xr-x 8 user user user 4.0K Nov 16 10:28  ..
28709895 -rwxr----- 2 user user user 5.9M Feb 25  2021  12_100229_1_98218.pdf
33556248 drwxr-xr-x 2 user user user 4.0K Jan 20  2021  bash_lesson
28709895 -rwxr----- 2 user user user 5.9M Feb 25  2021  hard_link_example
28709795 drwxr-xr-x 2 user user user 4.0K Feb 25  2021  JAVA
29233467 -rw-rw-r-- 1 user user user  164 Mar  2  2021  life.doc
28709811 drwxr-xr-x 2 user user user 4.0K Feb 25  2021  Other
28709897 -rwxr----- 1 user user user 3.5M Feb 25  2021  Stivens_UNIX_razrabotka_setevyih_prilozheniy_57c5f9_291254.fb2
28581527 lrwxrwxrwx 1 user user user   21 Dec 10 16:19  sym_link_example -> 12_100229_1_98218.pdf
29101634 drwxr-xr-x 4 user user user 4.0K Feb 25  2021  TESTS
29232635 drwxr-xr-x 2 user user user 4.0K Feb 25  2021  C
28709898 -rwxr----- 1 user user user  12M Feb 25  2021 'Снейдер Йон Эффективное программирование TCP IP (2009).pdf'
28709899 -rwxr----- 1 user user user  11M Feb 25  2021 'Стивенс. UNIX. Разработка сетевых приложений.djvu'
28709900 -rwxr----- 1 user user user  28M Feb 25  2021 'Уоррен - Алгоритмические трюки для програмистов.pdf'
user@pc:~/Study/Books$
```

First line is total size of directory. Then by coluns:
- unique inode number, which hardlinks has the same as object they points
- file type (d, l, -)
- permissions for file
- number of links to file
- owner and group
- author (--author)
- size
- time of creation
- name

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

rwx rwx rwx
user group others

Each letter group can be trasformed to number by binary format:
r = 4
w = 2
x = 1

rwx = 111 = 2^2+2^1+2^0 = 4+2+1=7

So rw-r--r--will means 644

1st rwx – read write and execute permissions for the file owner
2nd rwx – for the group owner of the file

3rd rwx – for all other users

stat command will show octal representation of file permission

```
user@pc:~/Study/Books$ stat ./
  File: ./
  Size: 4096            Blocks: 8          IO Block: 4096   directory
Device: 801h/2049d      Inode: 29232640    Links: 7
Access: (0755/drwxr-xr-x)  Uid: ( 1000/    user)   Gid: ( 1000/    user)
Access: 2021-12-10 16:21:49.443618807 +0200
Modify: 2021-12-10 16:21:47.675552543 +0200
Change: 2021-12-10 16:21:47.675552543 +0200
 Birth: -
user@pc:~/Study/Books$ stat -c '%a' 12_100229_1_98218.pdf
740
user@pc:~/Study/Books$
```

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

UID (user identifier) is a number assigned by Linux to each user on the system. This number is used to identify the user to the system and to determine which files the user can access.

```
user@user-vm:~/Example$ ls -lan
total 12
drwxrwxr-x  3 1000 1000 4096 грудень 12 18:07 .
drwxr-xr-x 19 1000 1000 4096 грудень 12 18:06 ..
-r--r--r--  1 1001 1000    0 грудень 12 18:06 1.txt
drwxrwxr-x  2 1001 1000 4096 грудень 12 18:07 testdir
user@user-vm:~/Example$
```

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.

```
user@user-vm:~/Example$ ls
1.txt  testdir
user@user-vm:~/Example$ sudo chown test_man 1.txt
user@user-vm:~/Example$ sudo chown -R test_man testdir/
user@user-vm:~/Example$
```

```
user@user-vm:~/Example$ ls -la
total 12
drwxrwxr-x  3 user      user 4096 rpy 12 18:07 .
drwxr-xr-x 19 user      user 4096 rpy 12 18:06 ..
-rw-rw-r--  1 test_man user    0 rpy 12 18:06 1.txt
drwxrwxr-x  2 test_man user 4096 rpy 12 18:07 testdir
user@user-vm:~/Example$ chmod +x 1.txt
chmod: changing permissions of '1.txt': Operation not permitted
user@user-vm:~/Example$ sudo chmod +x 1.txt
user@user-vm:~/Example$ ls -la
total 12
drwxrwxr-x  3 user      user 4096 rpy 12 18:07 .
drwxr-xr-x 19 user      user 4096 rpy 12 18:06 ..
-rwxrwxr-x  1 test_man user    0 rpy 12 18:06 1.txt
drwxrwxr-x  2 test_man user 4096 rpy 12 18:07 testdir
user@user-vm:~/Example$ sudo chmod 444 1.txt
user@user-vm:~/Example$ ls -la
total 12
drwxrwxr-x  3 user      user 4096 rpy 12 18:07 .
drwxr-xr-x 19 user      user 4096 rpy 12 18:06 ..
-r--r--r--  1 test_man user    0 rpy 12 18:06 1.txt
drwxrwxr-x  2 test_man user 4096 rpy 12 18:07 testdir
user@user-vm:~/Example$
```

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

```
user@user-vm:~/Example$  stat -c '%A %a %h %U %G %s %y %n' *
-r--r--r-- 444 1 test_man user 0 2021-12-12 18:06:35.629604163 +0200 1.txt
drwxrwxr-x 775 2 test_man user 4096 2021-12-12 18:07:37.040871796 +0200 testdir
```

umask is used for creation mask of permissions for new files and directories. Unlike chmod umask has inverse mask. It cannot permit all right to file, only rw-rw-rw-, even if it will be 000.

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

A Sticky bit is a permission bit that is set on a file or a directory that lets only the owner of the file/directory or the root user to delete or rename the file. No other user is given privileges to delete the file created by some other user. Last symbol “t” in permissions means that sticky bit is set.

```
user@user-vm:~/Example$ ls -l
total 4
-r-Sr--r-- 1 test_man user    0 rpy 12 18:06 1.txt
-rw-rwSr-- 1 user     user    0 rpy 12 19:42 2.txt
-rw-rw-r-- 1 user     user    0 rpy 12 20:37 sticky_file.txt
drwxrwxr-x 2 test_man user 4096 rpy 12 18:07 testdir
user@user-vm:~/Example$ chmod +t sticky_file.txt
user@user-vm:~/Example$ ls -l
total 4
-r-Sr--r-- 1 test_man user    0 rpy 12 18:06 1.txt
-rw-rwSr-- 1 user     user    0 rpy 12 19:42 2.txt
-rw-rw-r-T 1 user     user    0 rpy 12 20:37 sticky_file.txt
drwxrwxr-x 2 test_man user 4096 rpy 12 18:07 testdir
```

Setuid is a permission bit that allows a user to run an executable as the owner of that file

```
user@user-vm:~/Example$ ls -l
total 4
-r--r--r-- 1 test_man user    0 rpy 12 18:06 1.txt
drwxrwxr-x 2 test_man user 4096 rpy 12 18:07 testdir
user@user-vm:~/Example$ chmod u+s 1.txt
chmod: changing permissions of '1.txt': Operation not permitted
user@user-vm:~/Example$ sudo chmod u+s 1.txt
[sudo] password for user:
user@user-vm:~/Example$ ls -l
total 4
-r-Sr--r-- 1 test_man user    0 rpy 12 18:06 1.txt
drwxrwxr-x 2 test_man user 4096 rpy 12 18:07 testdir
```

Setgid is a permission bit that allows a user to run an executable as the group that owns the file.

```
user@user-vm:~/Example$ ls
1.txt  2.txt  testdir
user@user-vm:~/Example$ chmod g+s 2.txt
user@user-vm:~/Example$ ls -l
total 4
-r-Sr--r-- 1 test_man user    0 rpy 12 18:06 1.txt
-rw-rwSr-- 1 user     user    0 rpy 12 19:42 2.txt
drwxrwxr-x 2 test_man user 4096 rpy 12 18:07 testdir
user@user-vm:~/Example$
```

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

To run command line script permission of it should be set for execution.\

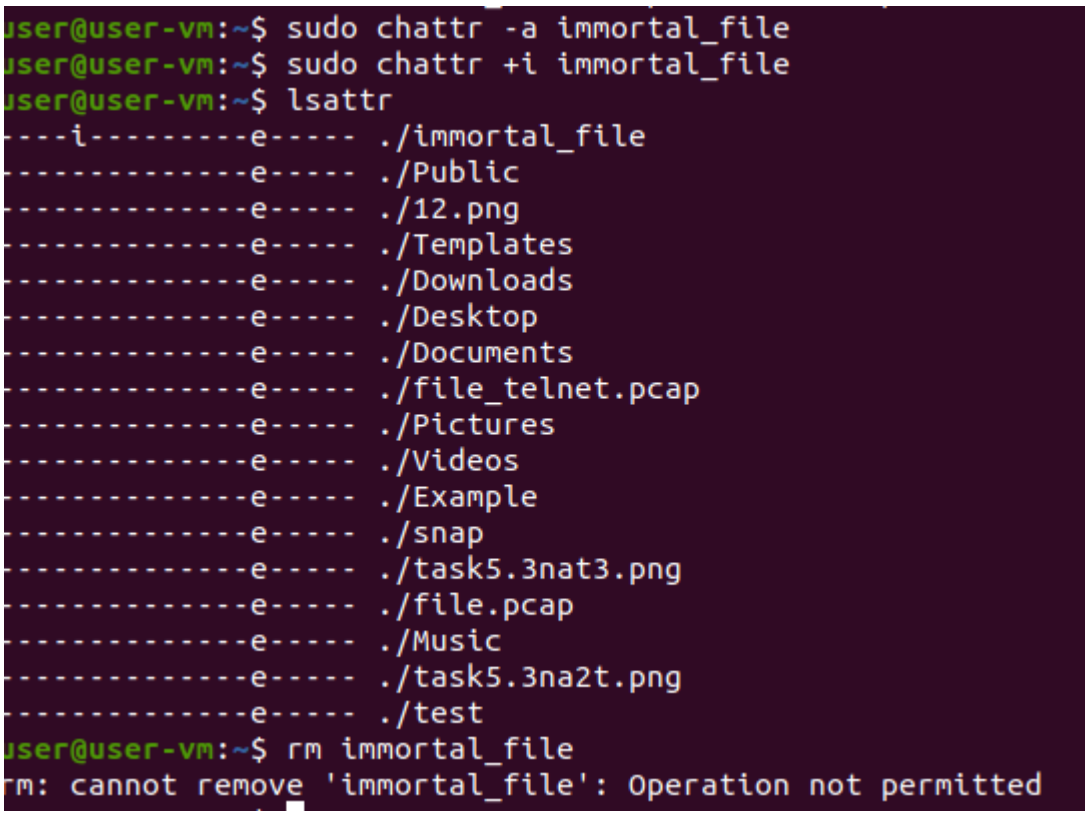
```
drwxr-xr-x  2 user     user      4096 вер  2 16:49 Public
-rwxrwxr-x  1 user     user         0 rpy 12 21:11 run.sh
-rw-rw-r--  1 user     user        72 вер  3 21:18 .selected_editor
drwx------  5 user     user      4096 rpy 11 12:04 snap
drwx------  2 user     user      4096 rpy 12 15:19 .ssh
-rw-r--r--  1 user     user         0 вер  2 16:52 .sudo_as_admin_successful
-rwxrw-rw-  1 user     user    114127 rpy 12 16:28 task5.3na2t.png
-rwxrw-rw-  1 user     user     43659 rpy 12 16:29 task5.3nat3.png
drwxr-xr-x  2 user     user      4096 вер  2 16:49 Templates
drwxr-xr-x  6 root     root      4096 вер  2 17:53 test
drwxr-xr-x  2 user     user      4096 вер  2 16:49 Videos
-rw-rw-r--  1 user     user       175 вер  3 20:51 .wget-hsts
user@user-vm:~$ sudo chmod +x run.sh
user@user-vm:~$ sudo chmod +x run.sh
user@user-vm:~/Example$ touch run.sh
user@user-vm:~/Example$ sudo chmod +x run.sh
user@user-vm:~/Example$ ls -l
total 4
-r-Sr--r-- 1 test_man user    0 rpy 12 18:06 1.txt
-rw-rwSr-- 1 user     user    0 rpy 12 19:42 2.txt
-rwxrwxr-x 1 user     user    0 rpy 12 21:12 run.sh
-rw-rw-r-T 1 user     user    0 rpy 12 20:37 sticky_file.txt
drwxrwxr-x 2 test_man user 4096 rpy 12 18:07 testdir
user@user-vm:~/Example$
```

There is also special attributes for files in Linux at the file system level, regardless of the standard (read, write, execute).
chattr (Change Attribute) is a Linux command line utility that is used to set / unset special file attributes to prevent accidental modification and deletion of files or directories, even if you are logged in as root. For initial stage only two attributes are important; append and immutable.

- **a - append only:** this attribute allows a file to be added to, but not to be removed. It prevents accidental or malicious changes to files that record data, such as log files.

```
user@user-vm:~$ sudo chattr +a immortal_file
user@user-vm:~$ sudo lsattr
-----a--------e----- ./immortal_file
--------------e----- ./Public
--------------e----- ./12.png
--------------e----- ./Templates
--------------e----- ./Downloads
--------------e----- ./Desktop
--------------e----- ./Documents
--------------e----- ./file_telnet.pcap
--------------e----- ./Pictures
--------------e----- ./Videos
--------------e----- ./Example
--------------e----- ./snap
--------------e----- ./task5.3nat3.png
--------------e----- ./file.pcap
--------------e----- ./Music
--------------e----- ./task5.3na2t.png
--------------e----- ./test
user@user-vm:~$ rm immortal_file
rm: cannot remove 'immortal_file': Operation not permitted
```

- **i - immutable:** it makes a file immutable, which goes a step beyond simply disabling write access to the file. The file can’t be deleted, links to it can’t be created, and the file can’t be renamed.



- **c - compressed:** it causes the kernel to compress data written to the file automatically and uncompress it when it's read back.
- **d - no dump:** it makes sure the file is not backed up in backups where the dump utility is used
- **e - extent format:** it indicates that the file is using extents for mapping the blocks on disk.
- **j - data journaling:** it ensures that on an Ext3 file system the file is first written to the journal and only after that to the data blocks on the hard disk.
- **s - secure deletion:** it makes sure that recovery of a file is not possible after it has been deleted.
- **t - no tail-merging:** Tail-merging is a process in which small data pieces at a file's end that don't fill a complete block are merged with similar pieces of data from other files.
- **u - undeletable:** When a file is deleted, its contents are saved which allows a utility to be developed that works with that information to salvage deleted files.
- **A - no atime updates:** Linux won't update the access time stamp when you access a file.
- **D - synchronous directory updates:** it makes sure that changes to files are written to disk immediately, and not to cache first.
- **S - synchronous updates:** the changes on a file are written synchronously on the disk.
- **T - and top of directory hierarchy:** A directory will be deemed to be the top of directory hierarchies for the purposes of the Orlov block allocator.