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

   In /etc/passwd and /etc/group we can find information about users in system that looks like group_name:password:group_id:list in /etc/group or user_name:password:user_id:group_id:info:directory:shell in /etc/passwd pseudo-users the last needed to confirm process ownership they haven't password

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

   Users that have id from 1 to 999 daemons, system, reserved users and pseudo-users the last needed to confirm process ownership. User with id 0 – root. And users with id from 1000 to 9999 it's regular users.

3) What is GID? How to define it?

   GID it's id of the group that user belongs to. We can check /etc/passwd or /etc/group file to find user's GID

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

   By his GID

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

   For add user to the system using commad adduser or useradd with require parameter is username

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

   Using command usermod with key -l + new_username

7) What is skell_dir? What is its structure?

   Skell dir it's directory that contains files for copying to the new user directory

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

   To remove user from the system including gis mailbox and home directory we should use userdel -r username

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

For lock and unlock user account we use usermod with -L flag for lock or with -U flag for unlock

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

passwd -d username

```
root@CsnKhai:/home/student# passwd -d anoly
passwd: password expiry information changed.
```

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

ls -l. first column: type of the file and permissions for it (read,write,execute, second colum means count of subderictories , 3rd – owner, 4th – group owner, 5th – occupied space, 6th – date of creation, 7th – name

```
student@CsnKhai:~$ ls -l
total 564
-rw-rw-r-- 1 student student    109 Nov 15 14:59 aboutroot
-rw-rw-r-- 1 student student    109 Nov 16 15:25 dirinfo
-rw-rw-r-- 1 student student 560105 Nov 16 14:54 -L
drwxrwxr-x 2 student student   4096 Nov 16 15:53 newdir
-rw-rw-r-- 1 student student      0 Nov 17 20:16 ssh
drwxrwxr-x 2 student student   4096 Nov 17 20:16 test
```

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

rwx – Read, Write, Execute. Access rights exists for owner, for group and for others

13) 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.

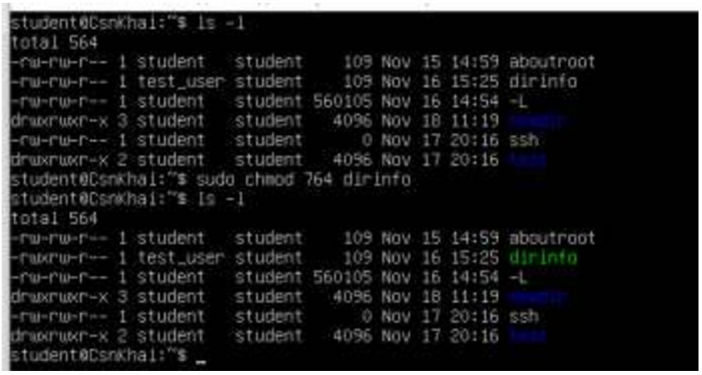
Chown to change owner and chmode to change mode of access. Examples:

```
student@CsnKhai:~$ ls -l
total 564
-rw-rw-r-- 1 student student    109 Nov 15 14:59 aboutroot
-rw-rw-r-- 1 anoly   student    109 Nov 16 15:25 dirinfo
-rw-rw-r-- 1 student student 560105 Nov 16 14:54 -L
drwxrwxr-x 3 student student   4096 Nov 18 11:19 newdir
-rw-rw-r-- 1 student student      0 Nov 17 20:16 ssh
drwxrwxr-x 2 student student   4096 Nov 17 20:16 test
student@CsnKhai:~$ sudo chown test_user  dirinfo
```

```
student@CsnKhai:~$ sudo chmod -w dirinfo
chmod: dirinfo: new permissions are r--rw-r--, not r--r--r--
student@CsnKhai:~$ ls -l
total 564
-rw-rw-r-- 1 student   student    109 Nov 15 14:59 aboutroot
-r--rw-r-- 1 test_user student    109 Nov 16 15:25 dirinfo
-rw-rw-r-- 1 student   student 560105 Nov 16 14:54 -L
drwxrwxr-x 3 student   student   4096 Nov 18 11:19 newdir
-rw-rw-r-- 1 student   student      0 Nov 17 20:16 ssh
drwxrwxr-x 2 student   student   4096 Nov 17 20:16 test
student@CsnKhai:~$
```

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

We can use octal representation of access rights, where: 1 – x, 2 – w, 3 – wx, 4 – r, 5 – rx, 6 – rw, 7 – rwx.



Umask command sets default access rights for files in filesystem

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

Sticky bit protects file from mistaken deletion. We can manually append it to file using symbolic way: chmod +t filename, or numerical way, for example 1754 (file with sticky bit must have executable rights at least for owner). By default in system directory /tmp have sticky bit.

```
student@CsnKhai:~$ ls -l /
total 72
drwxr-xr-x  2 root root  4096 Sep 15  2015 bin
drwxr-xr-x  3 root root  4096 Sep 15  2015 boot
drwxr-xr-x 14 root root  4000 Nov 18 11:07 dev
drwxr-xr-x 83 root root  4096 Nov 18 11:09 etc
drwxr-xr-x  5 root root  4096 Nov 17 18:33 home
lrwxrwxrwx  1 root root    33 Sep 15  2015 initrd.im
63-generic
drwxr-xr-x 22 root root  4096 Sep 15  2015 lib
drwx------  2 root root 16384 Sep 15  2015 lost+foun
drwxr-xr-x  2 root root  4096 Sep 15  2015 media
drwxr-xr-x  2 root root  4096 Apr 10  2014 mnt
drwxr-xr-x  2 root root  4096 Sep 15  2015 opt
dr-xr-xr-x 72 root root     0 Nov 18 11:07 proc
drwx------  5 root root  4096 Sep 15  2015 root
drwxr-xr-x 16 root root   540 Nov 18 11:07 run
drwxr-xr-x  2 root root  4096 Sep 15  2015 sbin
drwxr-xr-x  2 root root  4096 Sep 15  2015 srv
dr-xr-xr-x 13 root root     0 Nov 18 11:07 sys
drwxrwxrwt  2 root root  4096 Nov 18 11:46 tmp
drwxr-xr-x 10 root root  4096 Sep 15  2015 usr
drwxr-xr-x 11 root root  4096 Sep 15  2015 var
lrwxrwxrwx  1 root root    30 Sep 15  2015 vmlinuz -
eric
```

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

-rwx-rwx-r--