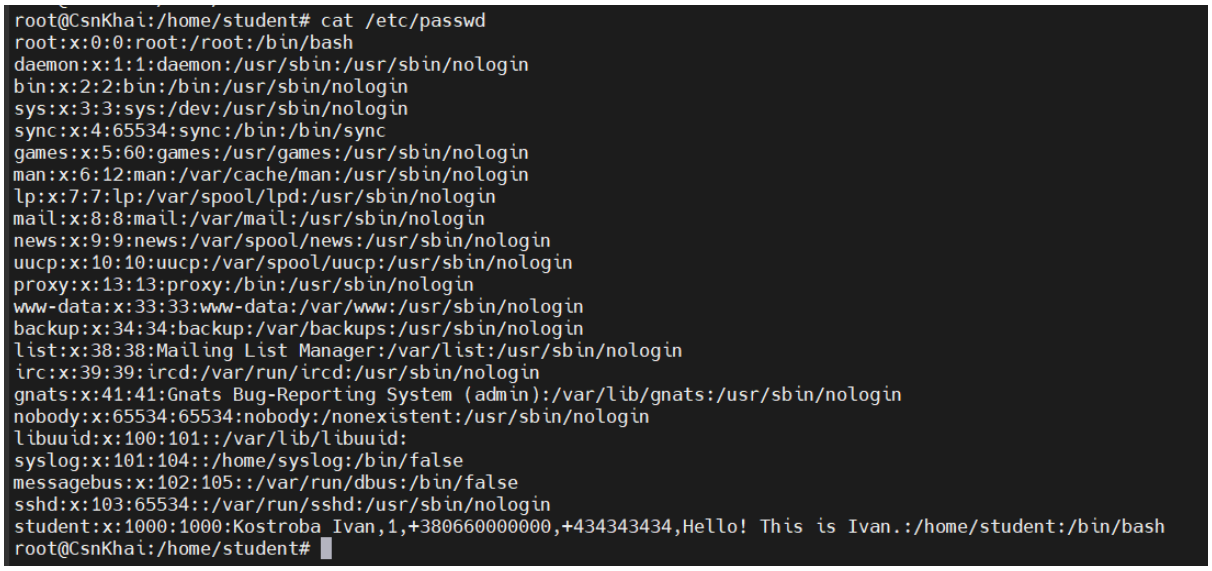
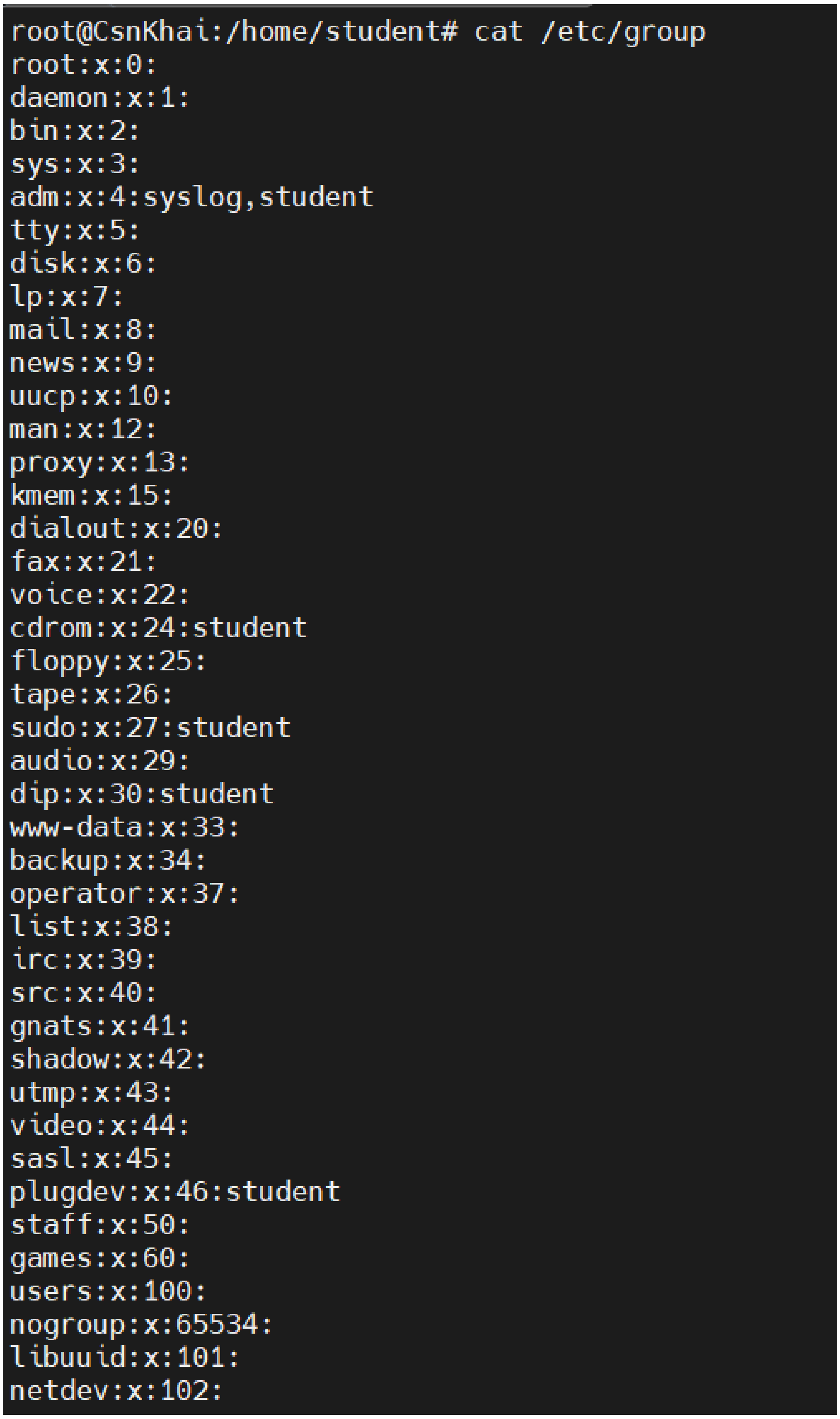
Task2
Task assignment.
1) Analyse the structure of the /etc/passwd and /etc/group file, what fields are present in it, what users exist on the system? Specify several pseudo-users, how to define them?



```
root@CsnKhai:/home/student# 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:Kostroba Ivan,1,+380660000000,+434343434,Hello! This is Ivan.:/home/student:/bin/bash
root@CsnKhai:/home/student# 
```

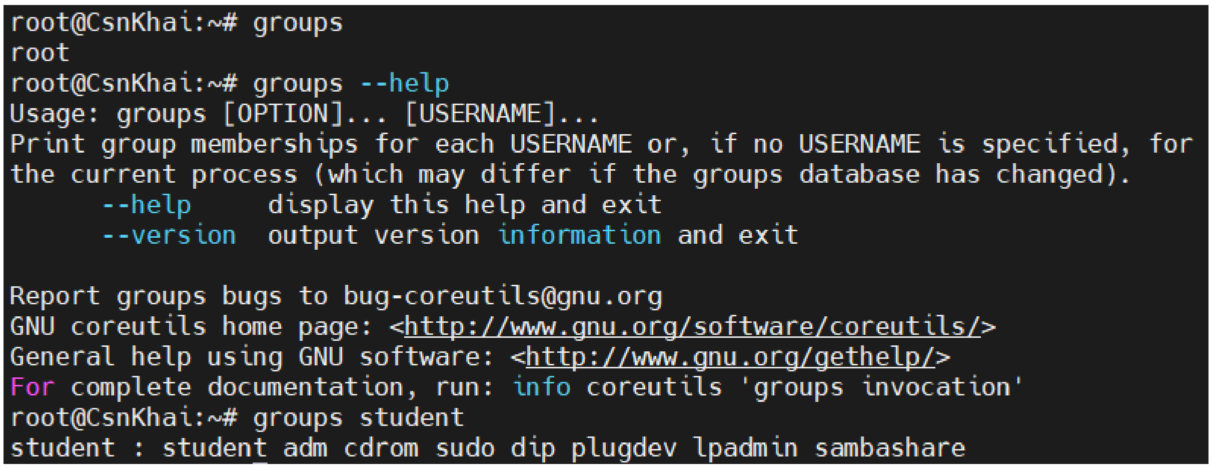
The structure of /etc/passwd is as follows:
Username: password (x because the actual password is hashed in /etc/shadow file) :user id : primary group id : comment : home directory : shell used. Pseudo-users are utility-users usually created by the system or additional applications that perform unseen tasks on the system. You cannot login as a pseudo-user, and it usually exists out of sight of a usual linux user. Pseudo-users can be determined by the following factors: A) they usually have an id from 1 to 999; B) they do not have a shell, having a /sbin/nologin instead (it is impossible to login as a pseudo-user). As can be seen from a system, aside from root, it has a bunch of pseudo-users (bin, sys, backup, nobody) and one actual user - a student.
The structure of /etc/group is:
Group name : password (it has x for the same reason users in etc/passwd have it) : group ID : list of users that belong to the group (except the users whose primary group is this one).
Contents of /etc/group:



```
root@CsnKhai:/home/student# cat /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:
```

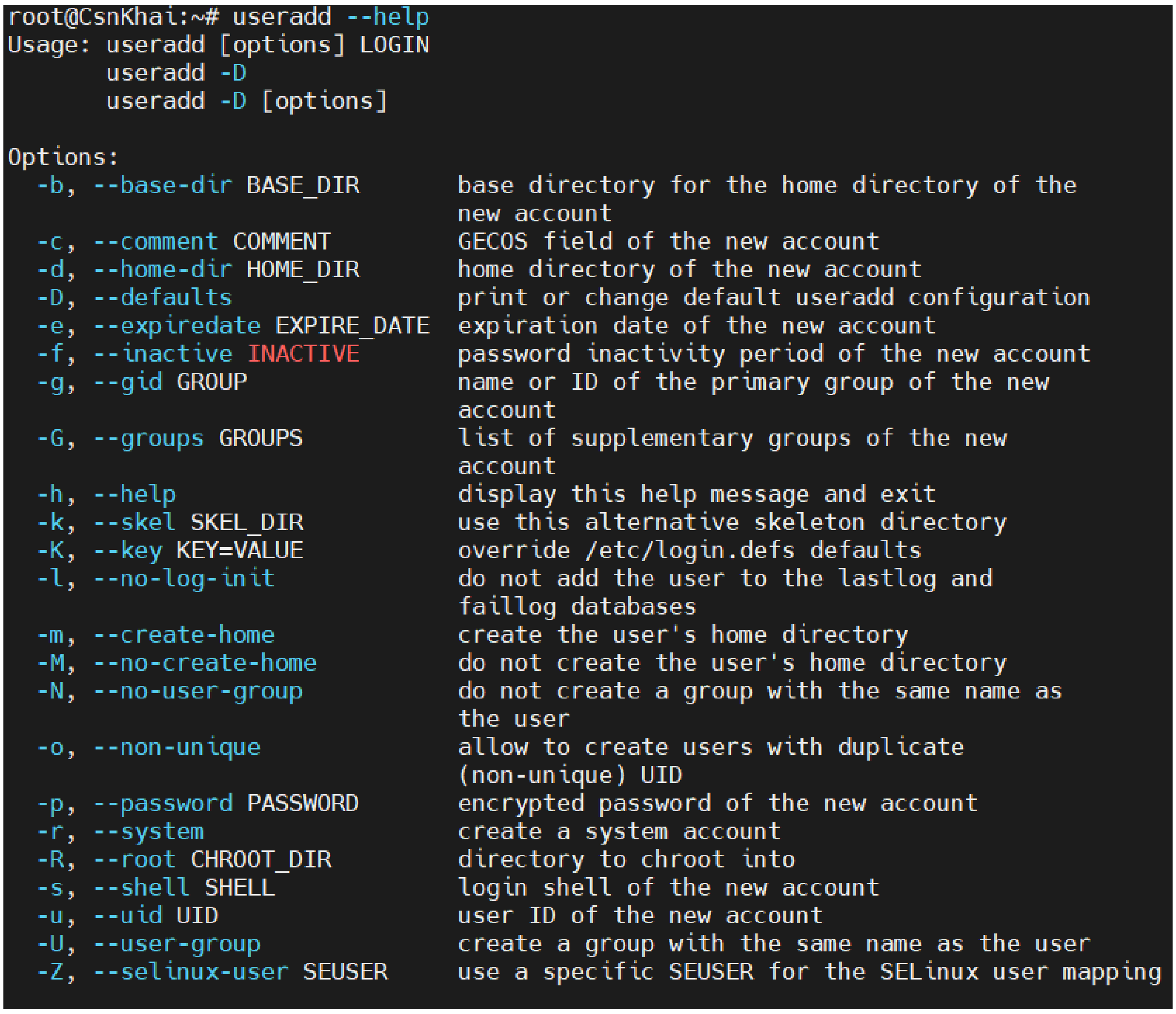
2) What are the uid ranges? What is UID? How to define it?

UID is a User Identifier in Linux. It is unique for each user, and it stands in a range from 0 to over 60 000 (who needs that many users, anyway?). Usual UID ranges are as follows: 0 for root, from 1 to 999 to pseudo-users, from 1000 and on for actual users.
UID can be changed via usermod command using the -u key. Also, it can be set by using the -u key with useradd command, that creates the user.
3) What is GID? How to define it?
GID is a Group Identifier in Linux. It is unique for each group, and also goes from 0 to more than 60000. It can be set up by using the -g key in groupadd command when creating a group or changed later by the groupmod command with -g key.
4) How to determine the belonging of a user to the specific group?
User always belongs to his primary group, which is specified in the /etc/passwd file. Also, in /etc/group we can see if any groups have this user, aside from a primary group. Also, the list of all groups that user is present in can be found out by the groups <username> command.



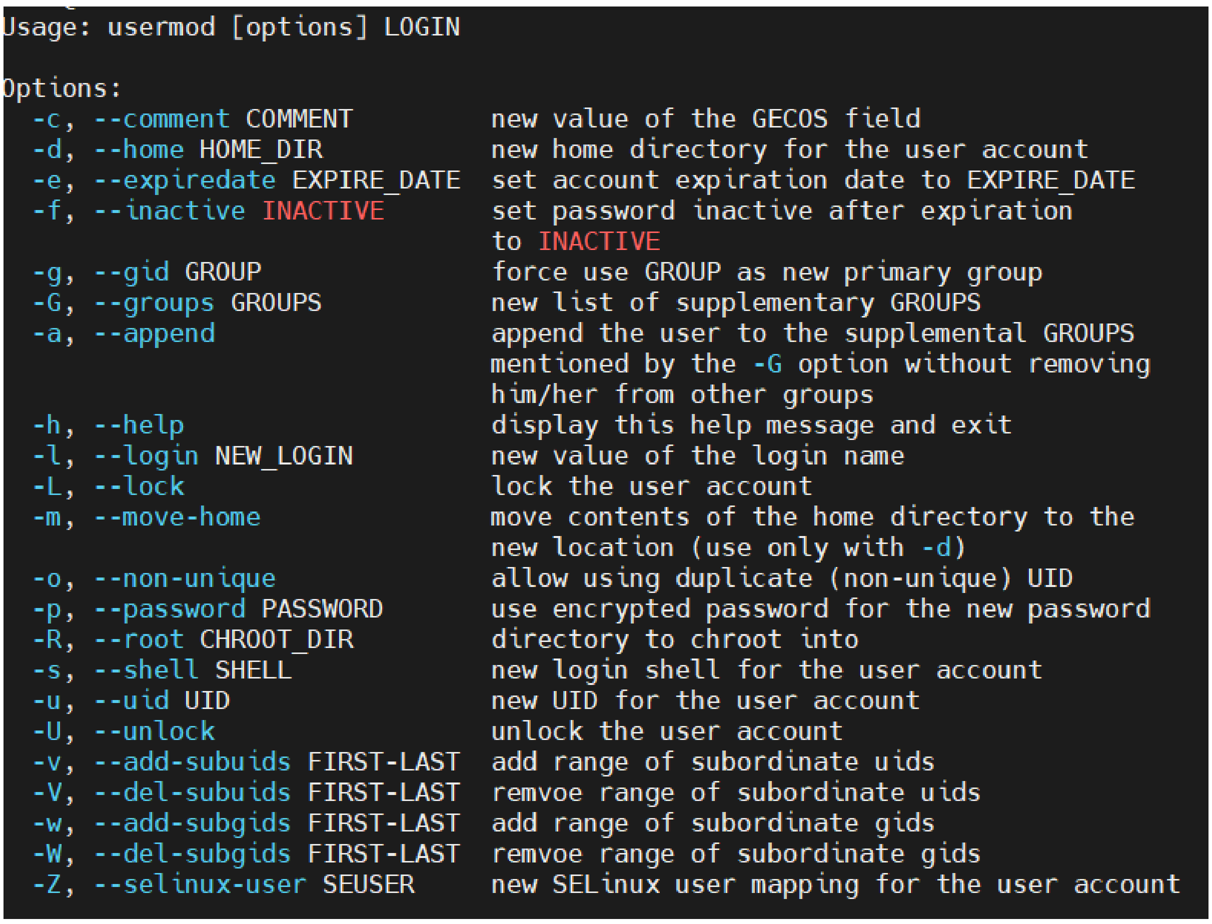
```
root@CsnKhai:~# groups
root
root@CsnKhai:~# groups --help
Usage: groups [OPTION]... [USERNAME]...
Print group memberships for each USERNAME or, if no USERNAME is specified, for
the current process (which may differ if the groups database has changed).
      --help     display this help and exit
      --version  output version information and exit

Report groups bugs to bug-coreutils@gnu.org
GNU coreutils home page: <http://www.gnu.org/software/coreutils/>
General help using GNU software: <http://www.gnu.org/gethelp/>
For complete documentation, run: info coreutils 'groups invocation'
root@CsnKhai:~# groups student
student : student adm cdrom sudo dip plugdev lpadmin sambashare
```

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



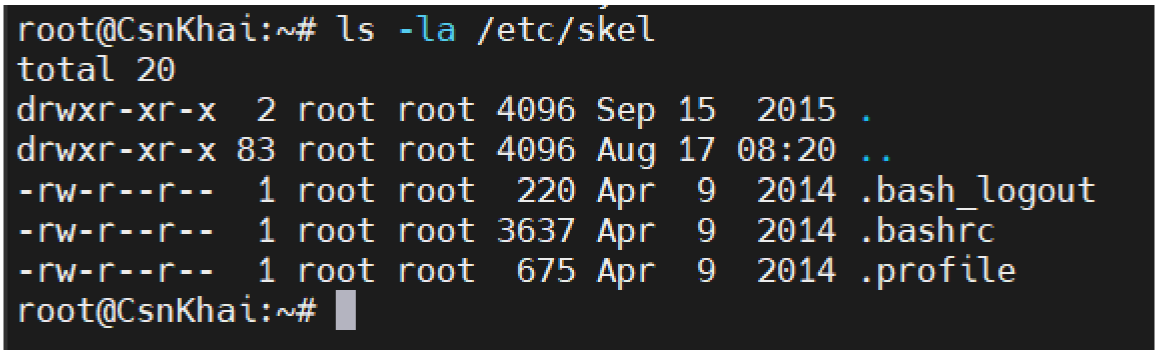
```
root@CsnKhai:~# useradd --help
Usage: useradd [options] LOGIN
       useradd -D
       useradd -D [options]

Options:
  -b, --base-dir BASE_DIR       base directory for the home directory of the
                                new account
  -c, --comment COMMENT         GECOS field of the new account
  -d, --home-dir HOME_DIR       home directory of the new account
  -D, --defaults                print or change default useradd configuration
  -e, --expiredate EXPIRE_DATE  expiration date of the new account
  -f, --inactive INACTIVE       password inactivity period of the new account
  -g, --gid GROUP               name or ID of the primary group of the new
                                account
  -G, --groups GROUPS           list of supplementary groups of the new
                                account
  -h, --help                    display this help message and exit
  -k, --skel SKEL_DIR           use this alternative skeleton directory
  -K, --key KEY=VALUE           override /etc/login.defs defaults
  -l, --no-log-init             do not add the user to the lastlog and
                                faillog databases
  -m, --create-home             create the user's home directory
  -M, --no-create-home          do not create the user's home directory
  -N, --no-user-group           do not create a group with the same name as
                                the user
  -o, --non-unique              allow to create users with duplicate
                                (non-unique) UID
  -p, --password PASSWORD       encrypted password of the new account
  -r, --system                  create a system account
  -R, --root CHROOT_DIR         directory to chroot into
  -s, --shell SHELL             login shell of the new account
  -u, --uid UID                 user ID of the new account
  -U, --user-group              create a group with the same name as the user
  -Z, --selinux-user SEUSER     use a specific SEUSER for the SELinux user mapping
```

6) How do I change the name (account name) of an existing user?
By using the usermod command with -l key.



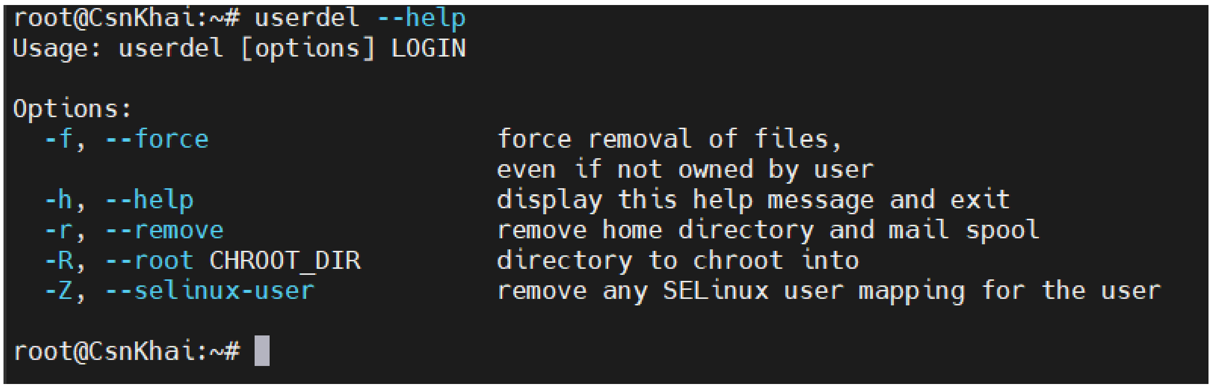
```
Usage: usermod [options] LOGIN

Options:
  -c, --comment COMMENT         new value of the GECOS field
  -d, --home HOME_DIR           new home directory for the user account
  -e, --expiredate EXPIRE_DATE  set account expiration date to EXPIRE_DATE
  -f, --inactive INACTIVE       set password inactive after expiration
                                to INACTIVE
  -g, --gid GROUP               force use GROUP as new primary group
  -G, --groups GROUPS           new list of supplementary GROUPS
  -a, --append                  append the user to the supplemental GROUPS
                                mentioned by the -G option without removing
                                him/her from other groups
  -h, --help                    display this help message and exit
  -l, --login NEW_LOGIN         new value of the login name
  -L, --lock                    lock the user account
  -m, --move-home               move contents of the home directory to the
                                new location (use only with -d)
  -o, --non-unique              allow using duplicate (non-unique) UID
  -p, --password PASSWORD       use encrypted password for the new password
  -R, --root CHROOT_DIR         directory to chroot into
  -s, --shell SHELL             new login shell for the user account
  -u, --uid UID                 new UID for the user account
  -U, --unlock                  unlock the user account
  -v, --add-subuids FIRST-LAST  add range of subordinate uids
  -V, --del-subuids FIRST-LAST  remvoe range of subordinate uids
  -w, --add-subgids FIRST-LAST  add range of subordinate gids
  -W, --del-subgids FIRST-LAST  remvoe range of subordinate gids
  -Z, --selinux-user SEUSER     new SELinux user mapping for the user account
```

7) What is skell_dir? What is its structure?
Directory skel, also - the skeleton directory - is used to initiate home directory when a new user is first created. Skeleton directory is defined by useradd file, usually /etc/skel. Sample /etc/skel:



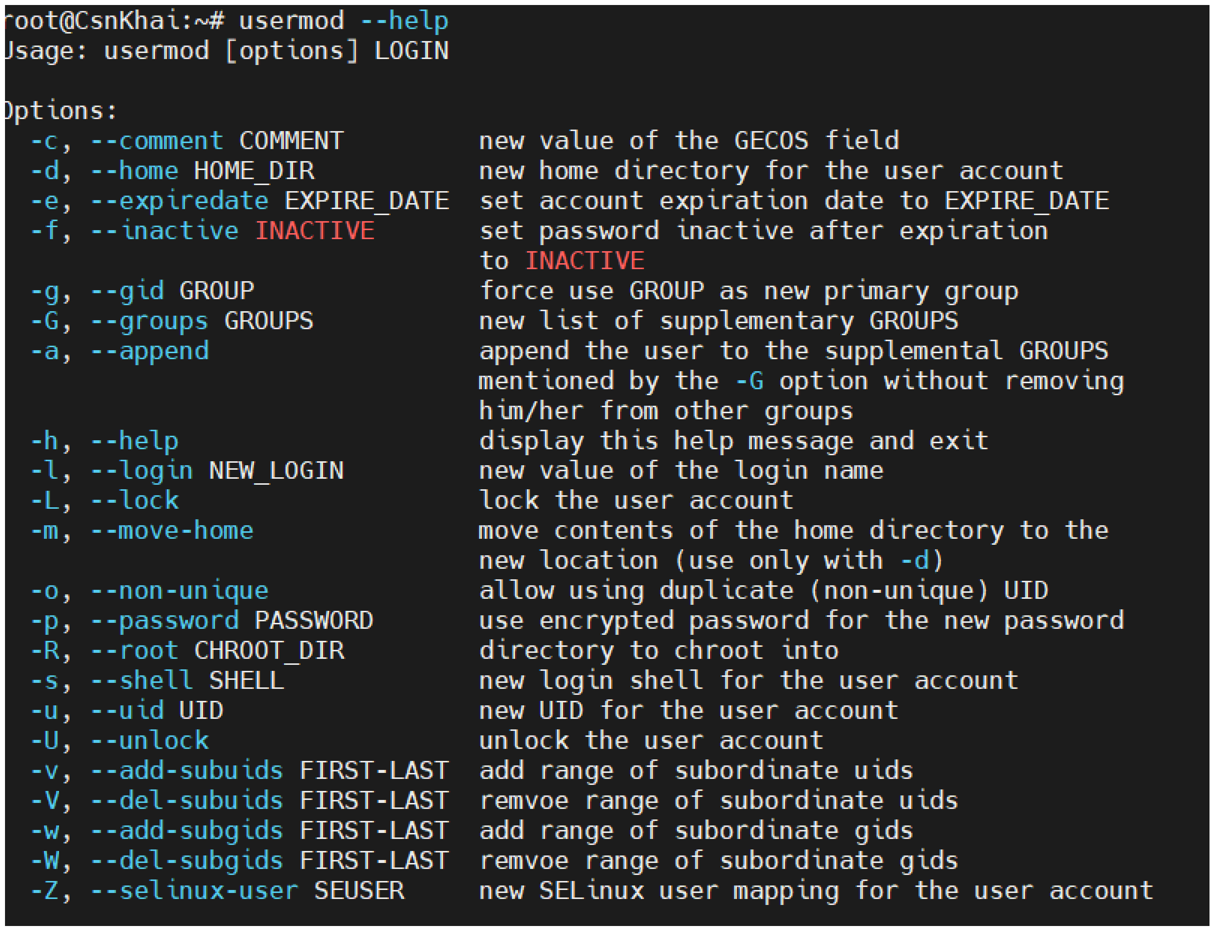
```
root@CsnKhai:~# ls -la /etc/skel
total 20
drwxr-xr-x  2 root root 4096 Sep 15  2015 .
drwxr-xr-x 83 root root 4096 Aug 17 08:20 ..
-rw-r--r--  1 root root  220 Apr  9  2014 .bash_logout
-rw-r--r--  1 root root 3637 Apr  9  2014 .bashrc
-rw-r--r--  1 root root  675 Apr  9  2014 .profile
root@CsnKhai:~#
```

So, for a new user, files such as .bashrc, .profile and .bash_logout will be created. If I add any text file to /etc/skel, it will be created in the home directory of all new users. Usually, the skel directory contains service files needed for a user.
8) How to remove a user from the system (including his mailbox)?
With a userdel command with -r key:



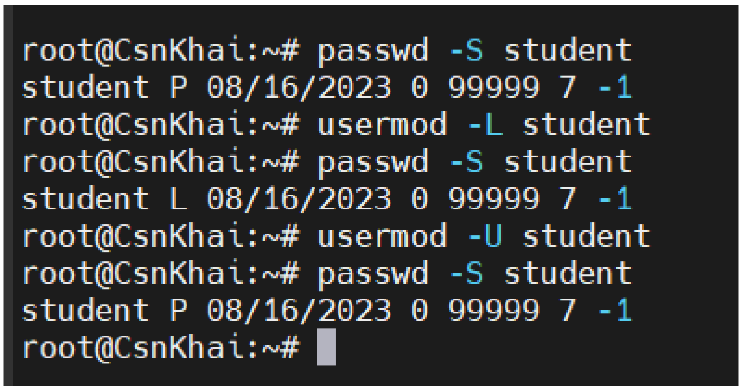
```
root@CsnKhai:~# userdel --help
Usage: userdel [options] LOGIN

Options:
  -f, --force                   force removal of files,
                                even if not owned by user
  -h, --help                    display this help message and exit
  -r, --remove                  remove home directory and mail spool
  -R, --root CHROOT_DIR         directory to chroot into
  -Z, --selinux-user            remove any SELinux user mapping for the user

root@CsnKhai:~# 
```

9) What commands and keys should be used to lock and unlock a user account?
Command usermod with keys -L (lock) and -U (unlock):



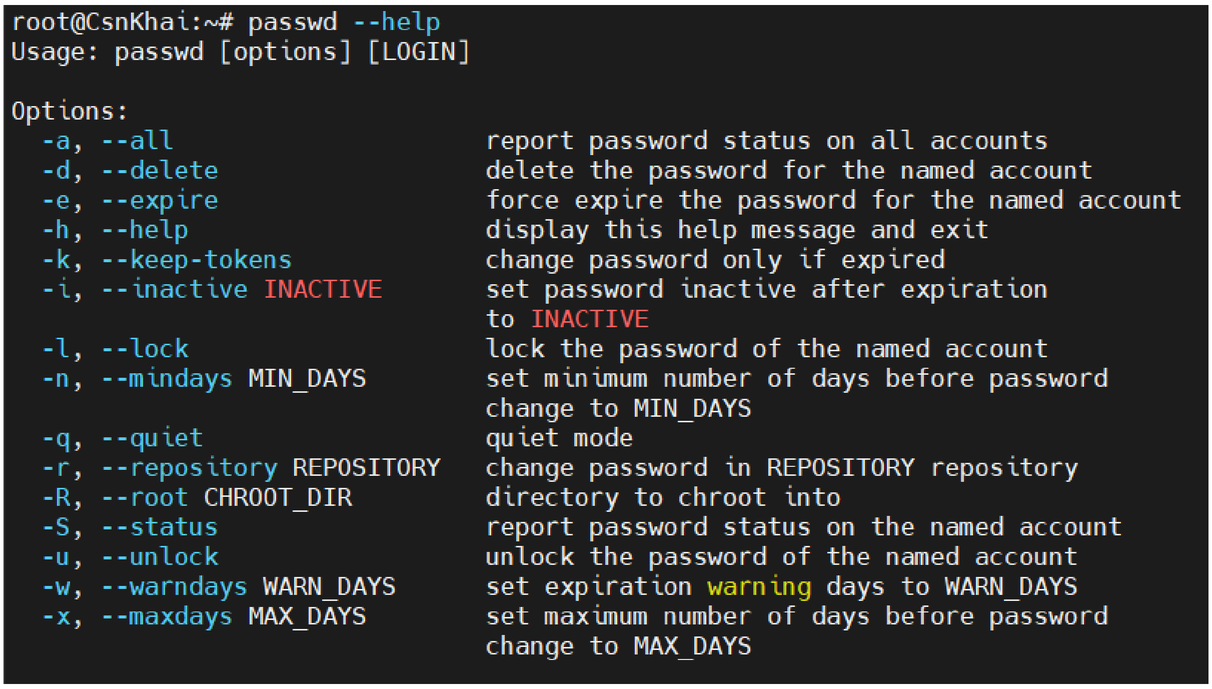
```
oot@CsnKhai:~# usermod --help
Jsage: usermod [options] LOGIN

ptions:
  -c, --comment COMMENT         new value of the GECOS field
  -d, --home HOME_DIR           new home directory for the user account
  -e, --expiredate EXPIRE_DATE  set account expiration date to EXPIRE_DATE
  -f, --inactive INACTIVE       set password inactive after expiration
                                to INACTIVE
  -g, --gid GROUP               force use GROUP as new primary group
  -G, --groups GROUPS           new list of supplementary GROUPS
  -a, --append                  append the user to the supplemental GROUPS
                                mentioned by the -G option without removing
                                him/her from other groups
  -h, --help                    display this help message and exit
  -l, --login NEW_LOGIN         new value of the login name
  -L, --lock                    lock the user account
  -m, --move-home               move contents of the home directory to the
                                new location (use only with -d)
  -o, --non-unique              allow using duplicate (non-unique) UID
  -p, --password PASSWORD       use encrypted password for the new password
  -R, --root CHROOT_DIR         directory to chroot into
  -s, --shell SHELL             new login shell for the user account
  -u, --uid UID                 new UID for the user account
  -U, --unlock                  unlock the user account
  -v, --add-subuids FIRST-LAST  add range of subordinate uids
  -V, --del-subuids FIRST-LAST  remvoe range of subordinate uids
  -w, --add-subgids FIRST-LAST  add range of subordinate gids
  -W, --del-subgids FIRST-LAST  remvoe range of subordinate gids
  -Z, --selinux-user SEUSER     new SELinux user mapping for the user account
```

Example usage:



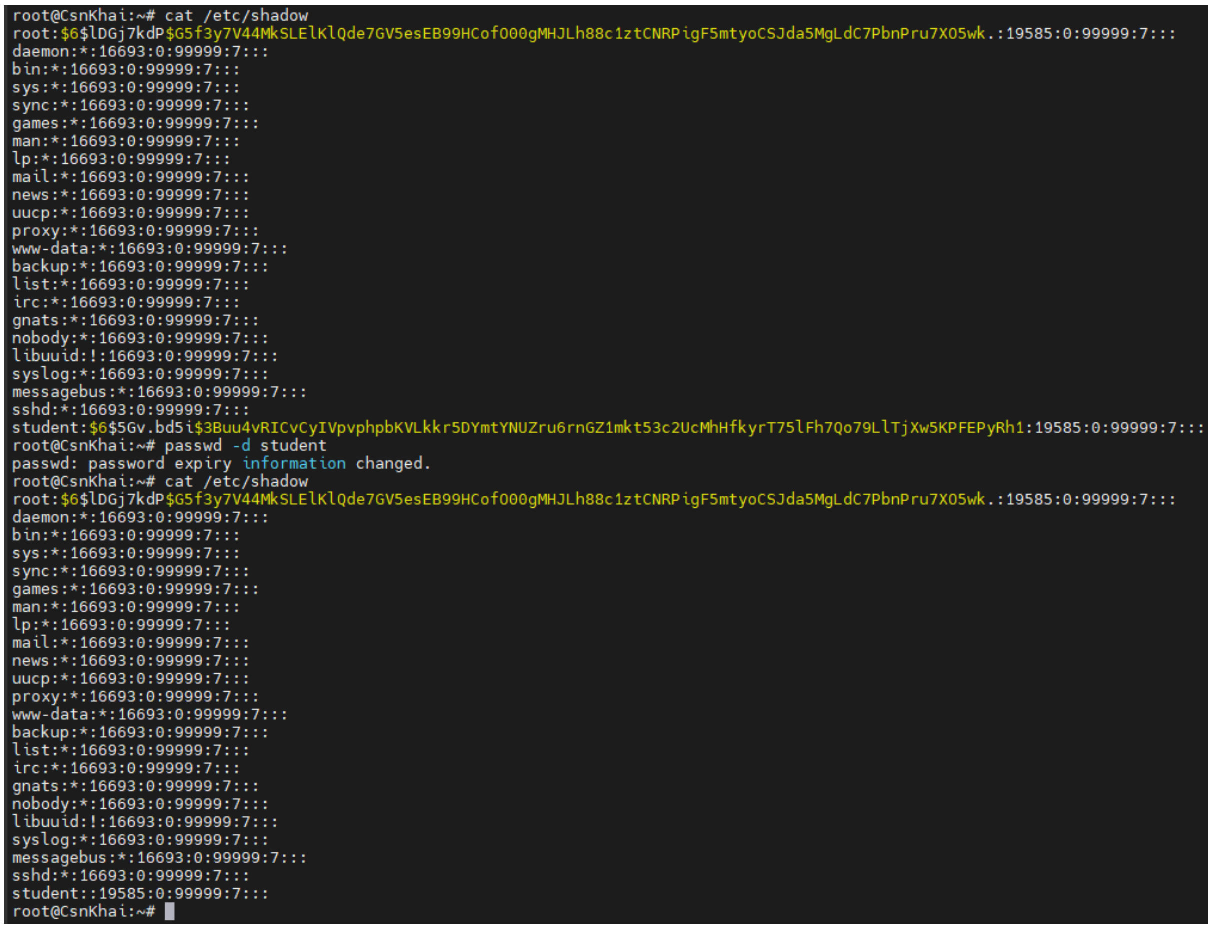
```
root@CsnKhai:~# passwd -S student
student P 08/16/2023 0 99999 7 -1
root@CsnKhai:~# usermod -L student
root@CsnKhai:~# passwd -S student
student L 08/16/2023 0 99999 7 -1
root@CsnKhai:~# usermod -U student
root@CsnKhai:~# passwd -S student
student P 08/16/2023 0 99999 7 -1
root@CsnKhai:~# 
```

10) How to remove a user's password and provide him with a password-free login for subsequent password change?
Via the key -d using the command passwd.



```
root@CsnKhai:~# passwd --help
Usage: passwd [options] [LOGIN]

Options:
  -a, --all                     report password status on all accounts
  -d, --delete                  delete the password for the named account
  -e, --expire                  force expire the password for the named account
  -h, --help                    display this help message and exit
  -k, --keep-tokens             change password only if expired
  -i, --inactive INACTIVE       set password inactive after expiration
                                to INACTIVE
  -l, --lock                    lock the password of the named account
  -n, --mindays MIN_DAYS        set minimum number of days before password
                                change to MIN_DAYS
  -q, --quiet                   quiet mode
  -r, --repository REPOSITORY   change password in REPOSITORY repository
  -R, --root CHROOT_DIR         directory to chroot into
  -S, --status                  report password status on the named account
  -u, --unlock                  unlock the password of the named account
  -w, --warndays WARN_DAYS      set expiration warning days to WARN_DAYS
  -x, --maxdays MAX_DAYS        set maximum number of days before password
                                change to MAX_DAYS
```

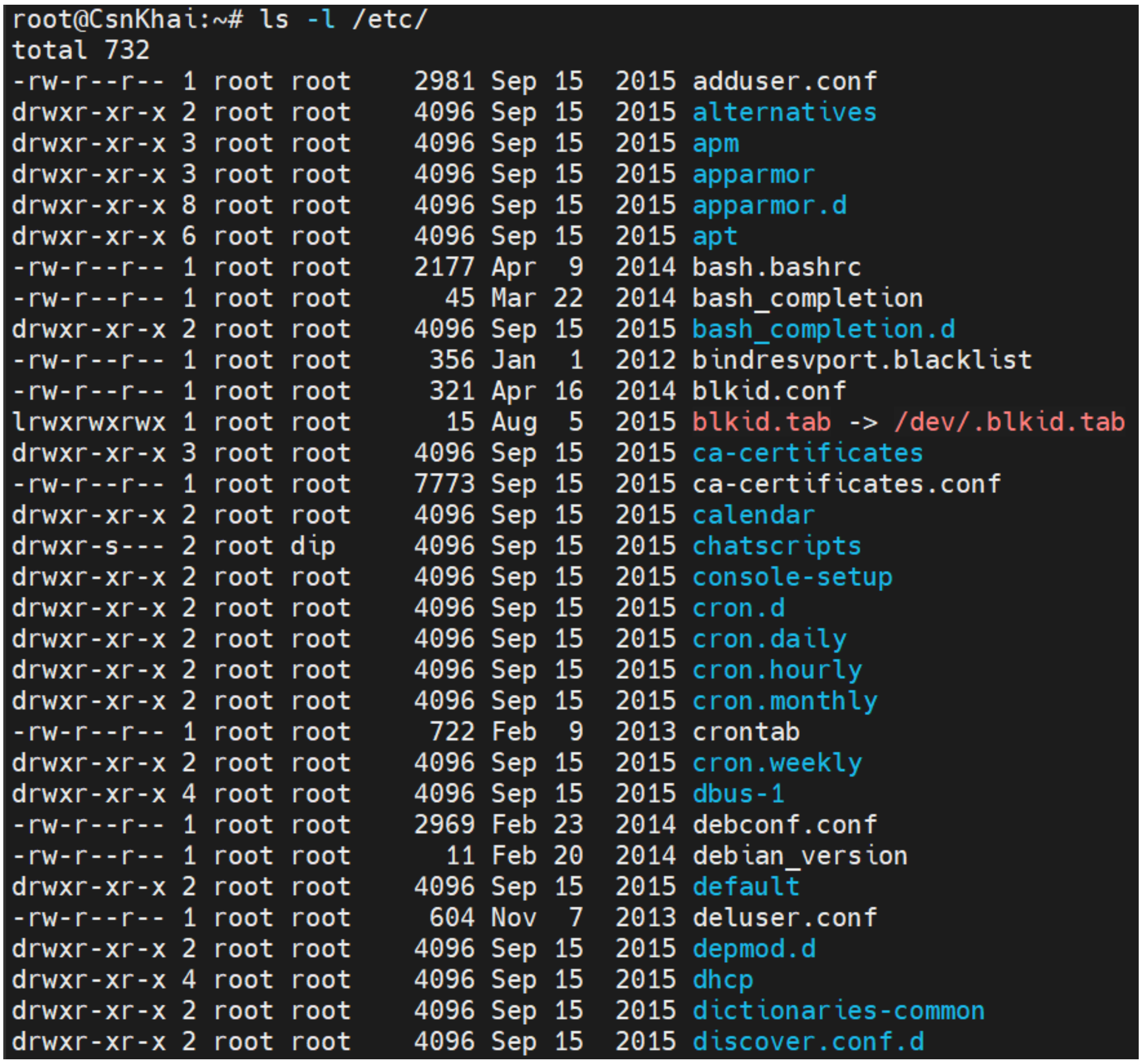
Usage example:



```
root@CsnKhai:~# cat /etc/shadow
root:$6$lDGj7kdP$G5f3y7V44MkSLElKlQde7GV5esEB99HCofO00gMHJLh88c1ztCNRPigF5mtyoCSJda5MgLdC7PbnPru7XO5wk.:19585:0:99999:7:::
daemon:*:16693:0:99999:7:::
bin:*:16693:0:99999:7:::
sys:*:16693:0:99999:7:::
sync:*:16693:0:99999:7:::
games:*:16693:0:99999:7:::
man:*:16693:0:99999:7:::
lp:*:16693:0:99999:7:::
mail:*:16693:0:99999:7:::
news:*:16693:0:99999:7:::
uucp:*:16693:0:99999:7:::
proxy:*:16693:0:99999:7:::
www-data:*:16693:0:99999:7:::
backup:*:16693:0:99999:7:::
list:*:16693:0:99999:7:::
irc:*:16693:0:99999:7:::
gnats:*:16693:0:99999:7:::
nobody:*:16693:0:99999:7:::
libuuid:!:16693:0:99999:7:::
syslog:*:16693:0:99999:7:::
messagebus:*:16693:0:99999:7:::
sshd:*:16693:0:99999:7:::
student:$6$5Gv.bd5i$3Buu4vRICvCyIVpvphpbKVLkkr5DYmtYNUZru6rnGZ1mkt53c2UcMhHfkyrT75lFh7Qo79LlTjXw5KPFEPyRh1:19585:0:99999:7:::
root@CsnKhai:~# passwd -d student
passwd: password expiry information changed.
root@CsnKhai:~# cat /etc/shadow
root:$6$lDGj7kdP$G5f3y7V44MkSLElKlQde7GV5esEB99HCofO00gMHJLh88c1ztCNRPigF5mtyoCSJda5MgLdC7PbnPru7XO5wk.:19585:0:99999:7:::
daemon:*:16693:0:99999:7:::
bin:*:16693:0:99999:7:::
sys:*:16693:0:99999:7:::
sync:*:16693:0:99999:7:::
games:*:16693:0:99999:7:::
man:*:16693:0:99999:7:::
lp:*:16693:0:99999:7:::
mail:*:16693:0:99999:7:::
news:*:16693:0:99999:7:::
uucp:*:16693:0:99999:7:::
proxy:*:16693:0:99999:7:::
www-data:*:16693:0:99999:7:::
backup:*:16693:0:99999:7:::
list:*:16693:0:99999:7:::
irc:*:16693:0:99999:7:::
gnats:*:16693:0:99999:7:::
nobody:*:16693:0:99999:7:::
libuuid:!:16693:0:99999:7:::
syslog:*:16693:0:99999:7:::
messagebus:*:16693:0:99999:7:::
sshd:*:16693:0:99999:7:::
student::19585:0:99999:7:::
root@CsnKhai:~# 
```

As can be seen, now password isn’t set up in /etc/shadow password.
11) Display the extended format of information about the directory, tell about

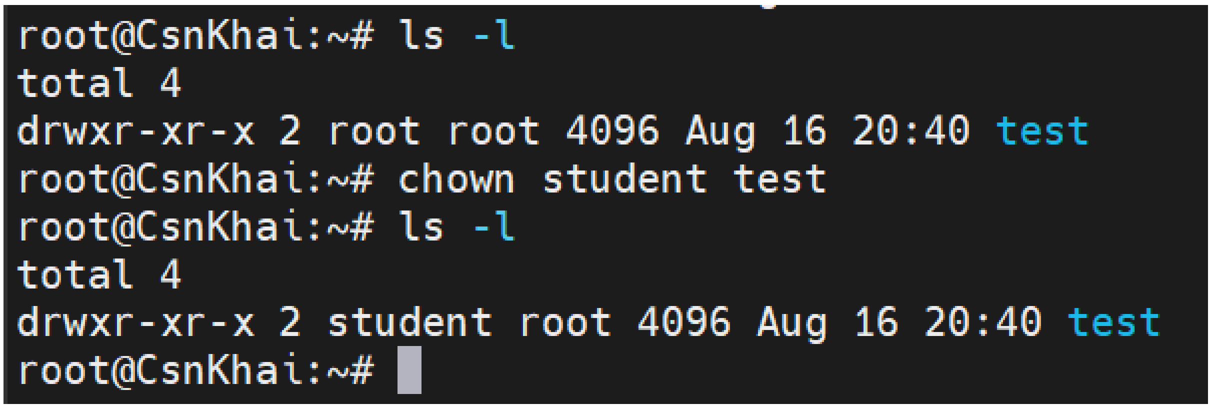
the information columns displayed on the terminal.
Ls with a key -l:



```
root@CsnKhai:~# ls -l /etc/
total 732
-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
```

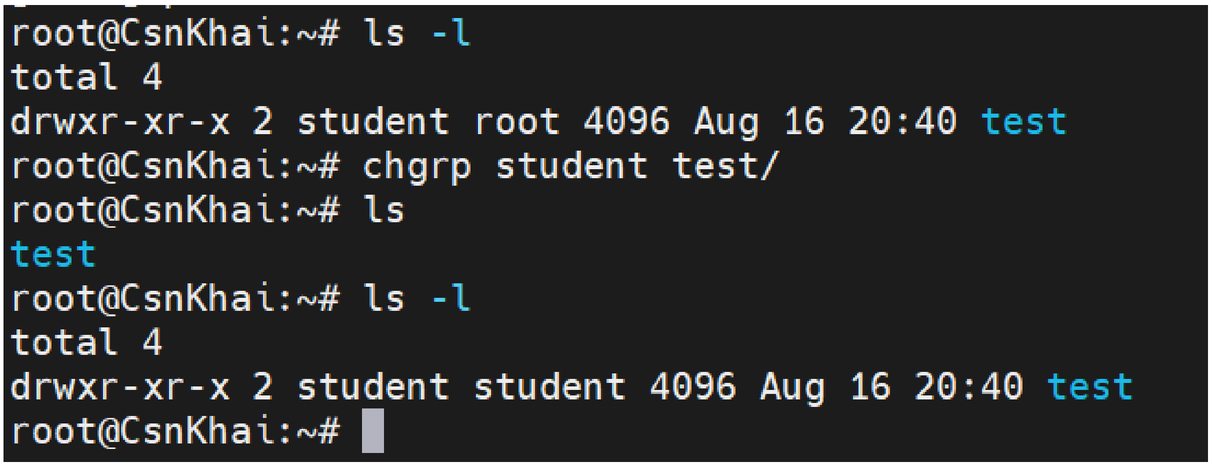
Columns meaning:
First column determines the type of file, and its permissions - first, for the owner of this file, the second for its group, and the third for all other users (so named guests). The second column determines the number of hard links pointing at this file. Third column determines owner of the file, fourth - group of the file. Fifth column determines the size of this file in bytes, and the sixth column shows the last time the file was changed (or created, if the file wasn't changed after the creation). Last column shows the file name. That's about it.
12) What access rights exist and for whom (i. e., describe the main roles)?
Briefly describe the acronym for access rights.
There are three types of access rights: Read, Write and eXecute - read for reading what the file consists of, write is a permission to edit the file, and execute for, unexpectedly, executing the file.
Roles, for which these permissions do exist, are so: First goes the owner of this file. Second goes the group of this file, and thus all the users that do exist in the same group as the file.
The third role is a guest user - any other than owner and same group user.

13) What is the sequence of defining the relationship between the file and the
User?
By default, the owner of the file is the same as the user that created it, and the group is pretty much the same - named for the user.
To setup any other user as an owner of the file, you can use a command “chown”:



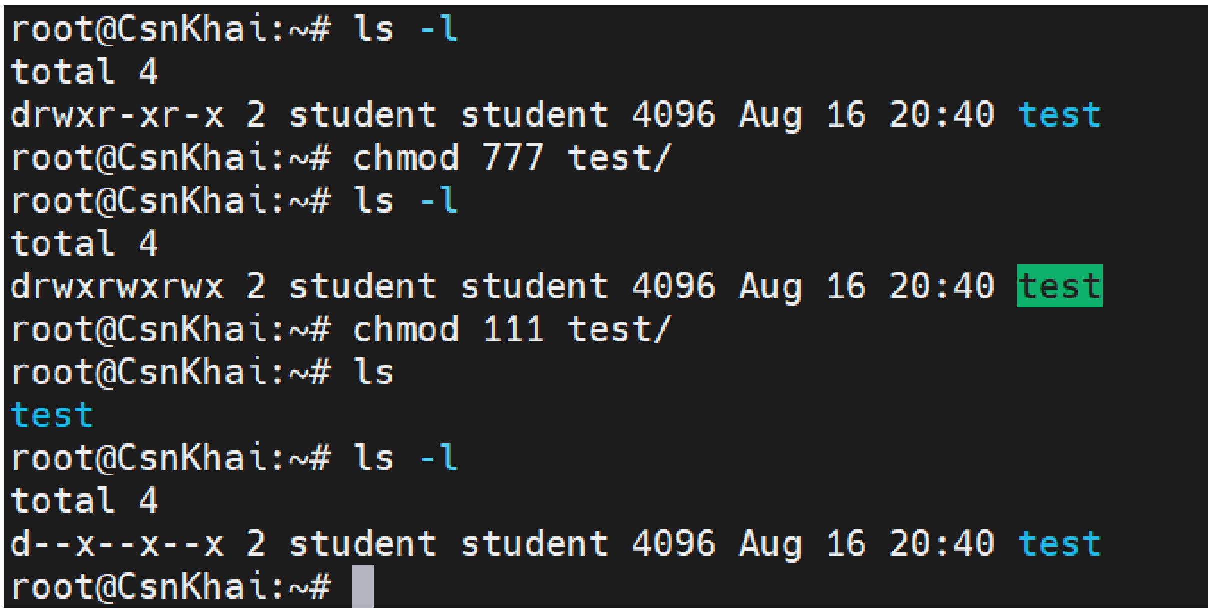
```
root@CsnKhai:~# ls -l
total 4
drwxr-xr-x 2 root root 4096 Aug 16 20:40 test
root@CsnKhai:~# chown student test
root@CsnKhai:~# ls -l
total 4
drwxr-xr-x 2 student root 4096 Aug 16 20:40 test
root@CsnKhai:~#
```

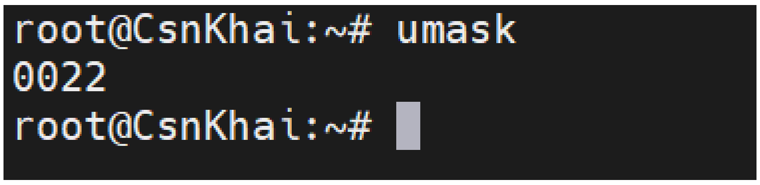
If the user that accesses the file is not the owner of the file, the group of the file needs to be checked - whether the user that tries to access the file is in this group. If not, the user trying to access the file is a guest user for this file and thus has the third sequence of permissions seen from ls -l command.
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 the owner you need to use the chown command - see screenshot above.
To change group of the file, you need to use the chgrp command:

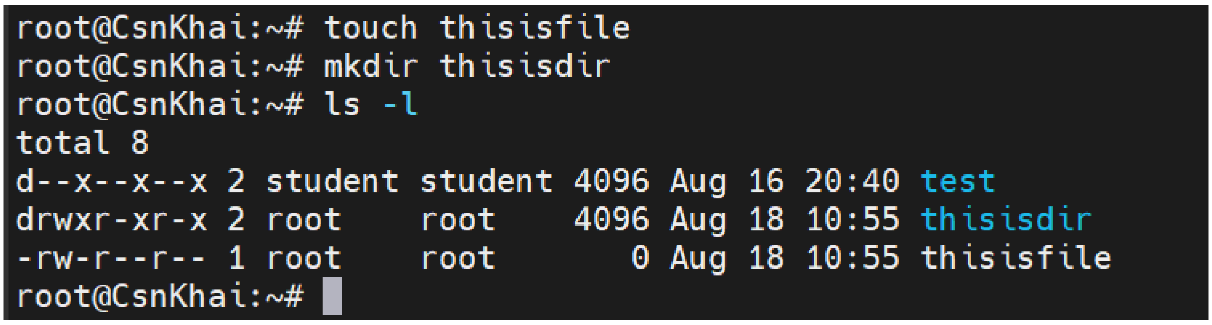


```
root@CsnKhai:~# ls -l
total 4
drwxr-xr-x 2 student root 4096 Aug 16 20:40 test
root@CsnKhai:~# chgrp student test/
root@CsnKhai:~# ls
test
root@CsnKhai:~# ls -l
total 4
drwxr-xr-x 2 student student 4096 Aug 16 20:40 test
root@CsnKhai:~#
```

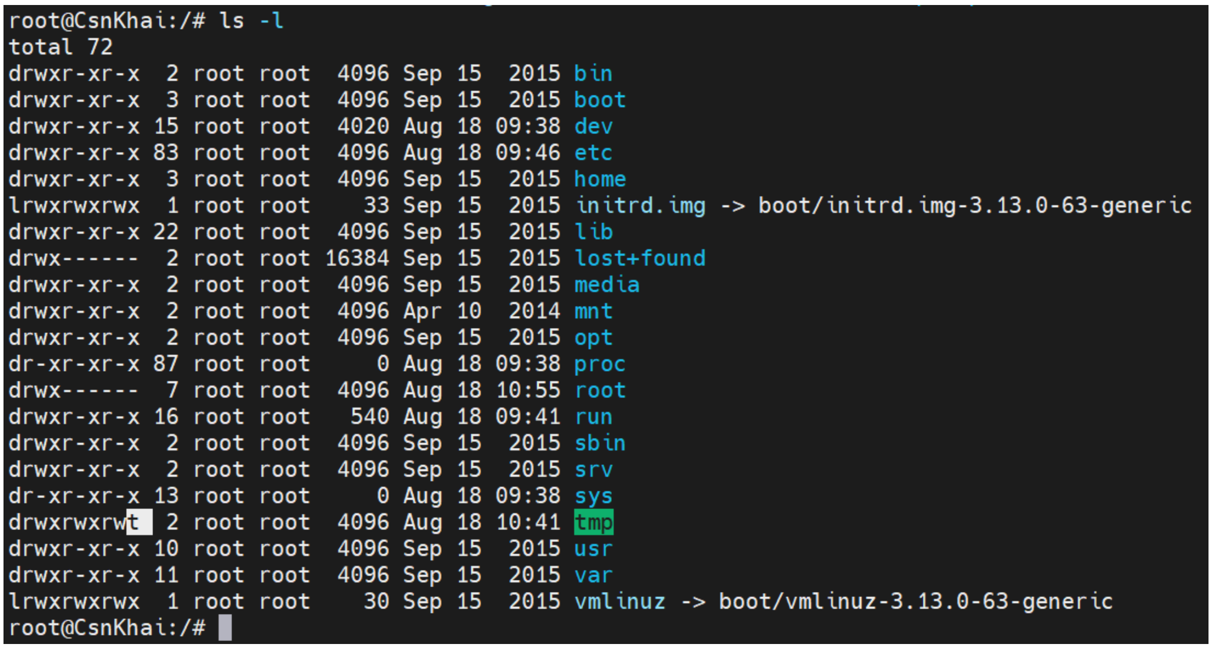
To change permissions of the file, you need to use command “chmod” and enter the new access rights for the file in octal format:


```
root@CsnKhai:~# ls -l
total 4
drwxr-xr-x 2 student student 4096 Aug 16 20:40 test
root@CsnKhai:~# chmod 777 test/
root@CsnKhai:~# ls -l
total 4
drwxrwxrwx 2 student student 4096 Aug 16 20:40 test
root@CsnKhai:~# chmod 111 test/
root@CsnKhai:~# ls
test
root@CsnKhai:~# ls -l
total 4
d--x--x--x 2 student student 4096 Aug 16 20:40 test
root@CsnKhai:~#
```
15) What is an example of octal representation of access rights? Describe the umask command.
Octal representation of access rights is a simple way to code the rights in one digit. This digit goes from 0 to 7 (so, it uses 3 bits) and is using the system of adding a particular number for different types of rights - 1 for eXecute, 2 for Write, 4 for Read. So, if you need to give rights to Write and eXecute, but not Read - it will be 1 + 2 = 3. If you need to give rights to Write and Read, but not eXecute - it will be 4 + 2 = 6. If all rights are needed, it will be 1 + 2 + 4 = 7. If no rights are given, the sum will be 0.
Umask is a command to set up the default permissions mask for new files. Mask means a number to be subtracted from a particular permission octal representation. To view current umask, simply type the command:


```
root@CsnKhai:~# umask
0022
root@CsnKhai:~#
```
It means 0 will be subtracted from owner rights, 2 from group rights and 2 from guest rights. (First 0 is to represent the absence of a sticky bit, more on that later).
The base values to subtract from are 6 for files and 7 for directories. Now, the permissions for a new file will be 6-0 = 6, 6-2 = 4, 6-2 = 4, or R+W, R, R.
For a new directory, it will be 7-0 = 7, 7-2 = 5, 7-2 = 5, or R+W+X, R+X, R+X. It can be seen by creating a new file and directory like that:

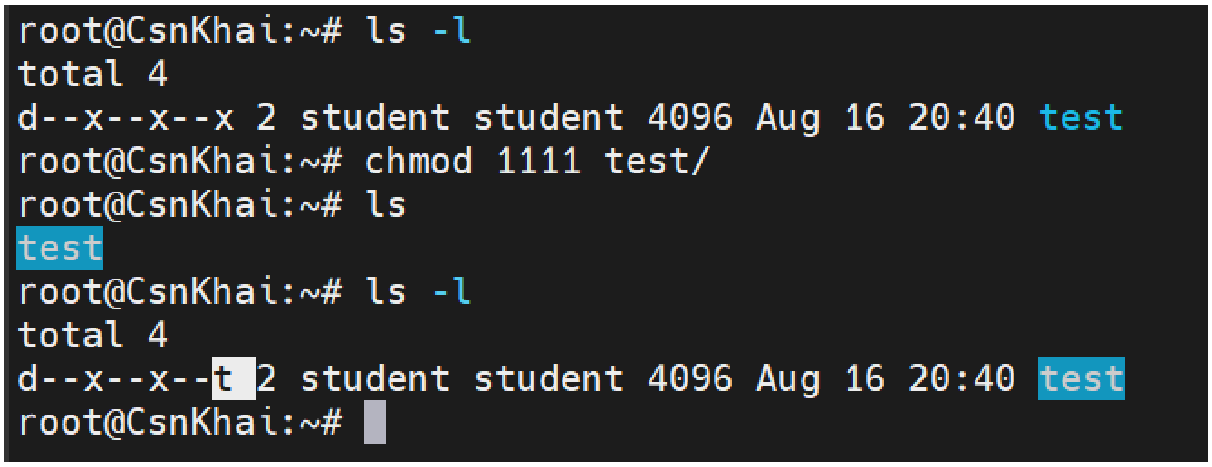

```
root@CsnKhai:~# touch thisisfile
root@CsnKhai:~# mkdir thisisdir
root@CsnKhai:~# ls -l
total 8
d--x--x--x 2 student student 4096 Aug 16 20:40 test
drwxr-xr-x 2 root    root    4096 Aug 18 10:55 thisisdir
-rw-r--r-- 1 root    root       0 Aug 18 10:55 thisisfile
root@CsnKhai:~#
```
16) Give definitions of sticky bits and mechanism of identifier substitution. Give

an example of files and directories with these attributes.
Sticky bit is an additional permissional parameter for files and directories, though, now it is used mainly on dirs to protect them and files in these directories from a deletion by users, who are not their owners. One of the examples of a sticky-bits on directories is one on tmp/ directory:



```
root@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 15 root root  4020 Aug 18 09:38 dev
drwxr-xr-x 83 root root  4096 Aug 18 09:46 etc
drwxr-xr-x  3 root root  4096 Sep 15  2015 home
lrwxrwxrwx  1 root root    33 Sep 15  2015 initrd.img -> boot/initrd.img-3.13.0-63-generic
drwxr-xr-x 22 root root  4096 Sep 15  2015 lib
drwx------  2 root root 16384 Sep 15  2015 lost+found
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 87 root root     0 Aug 18 09:38 proc
drwx------  7 root root  4096 Aug 18 10:55 root
drwxr-xr-x 16 root root   540 Aug 18 09:41 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 Aug 18 09:38 sys
drwxrwxrwt  2 root root  4096 Aug 18 10:41 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 -> boot/vmlinuz-3.13.0-63-generic
root@CsnKhai:/#
```

Here we can see an unexpected “t” letter in file permissions, while on all other files we do have standard R, W and Xs. “t” represents a sticky bit.
To set up a sticky bit, one need to use chmod command using “1” before usual permission values:



```
root@CsnKhai:~# ls -l
total 4
d--x--x--x 2 student student 4096 Aug 16 20:40 test
root@CsnKhai:~# chmod 1111 test/
root@CsnKhai:~# ls
test
root@CsnKhai:~# ls -l
total 4
d--x--x--t 2 student student 4096 Aug 16 20:40 test
root@CsnKhai:~#
```

Here, the usual 111 privileges were updated with a sticky bit before them.
17) What file attributes should be present in the command script?
Script needs to have a permission to eXecute (or X) for a user trying to execute them. Also, scripts should have a shebang on their first line to specify, using which interpreter will be used to execute the script.