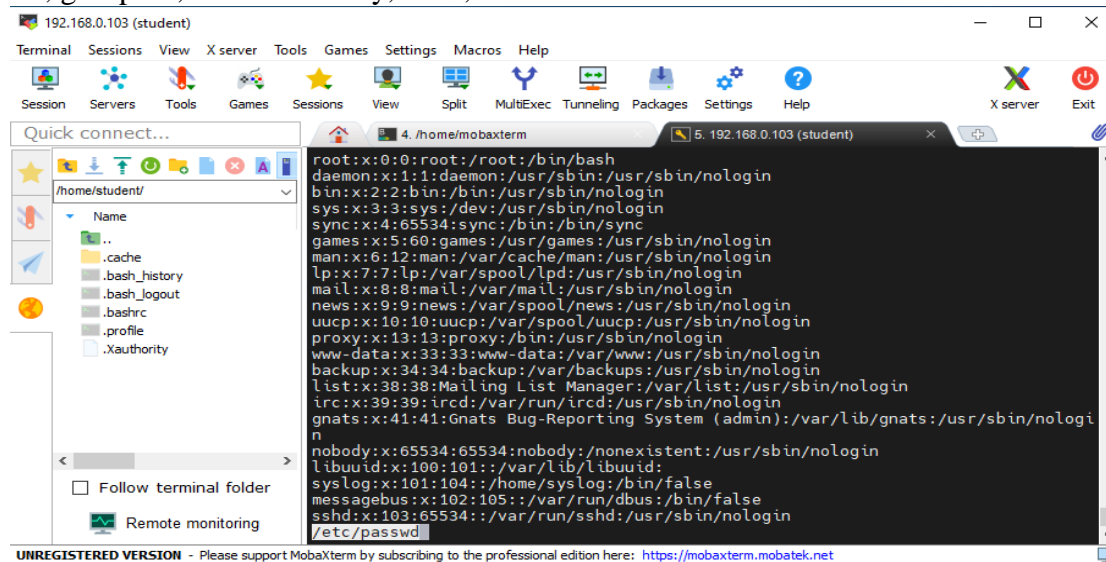


1) Analyze the structure of the `/etc/passwd` and `/etc/group` file, what fields are present in it, what users exist on the system? Specify several pseudo-users, how to define them?

`/etc/passwd` is a configuration file that contains information about user accounts. In this text file, we will find a list of system accounts, storing useful information from each account, such as user ID, group ID, home directory, shell, etc.



The screenshot shows a MobaXterm window with a terminal session. The terminal displays the contents of the `/etc/passwd` file. The output shows a list of system accounts, including `root`, `daemon`, `bin`, `sys`, `sync`, `games`, `man`, `lp`, `mail`, `news`, `uucp`, `proxy`, `www-data`, `backup`, `list`, `irc`, `gnats`, `nobody`, `libuid`, `syslog`, `messagebus`, `sshd`, and `/etc/passwd`. The fields in each line represent the username, UID, GID, and other account details.

User ID - login;

Password - the presence of a password;

UID - user identifier;

GID - default group ID;

User Info - additional information about the user (full name, contacts, etc.)

Home Dir - initial (home) directory;

Shell - registration shell, or shell

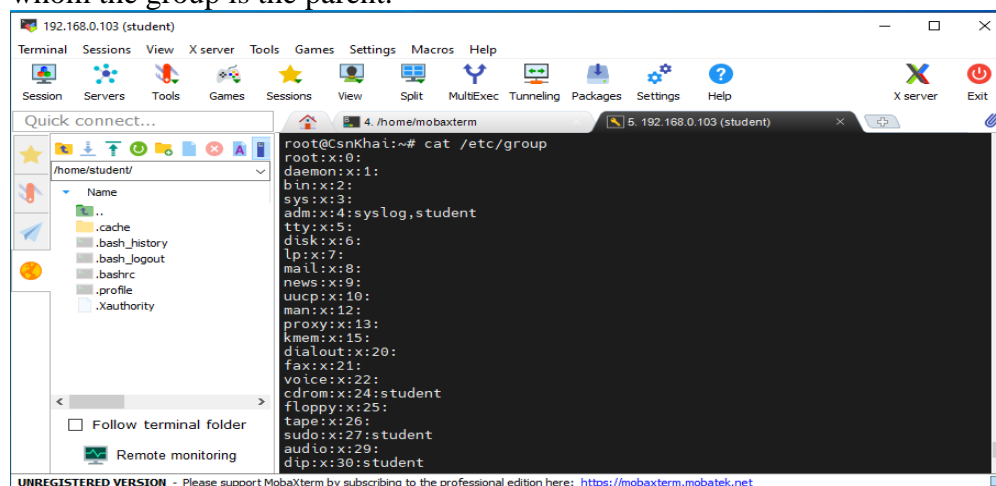
The **groups existing** on the system are listed in the `/etc/group` file, with the following fields:

The name of the group - name of the group.

Group password - encrypted group password (or x if shadow passwords are used).

Group ID (GID) - the identification number assigned to the group in the system.

List of members - a comma-separated list of users who belong to a group, excluding those for whom the group is the parent.



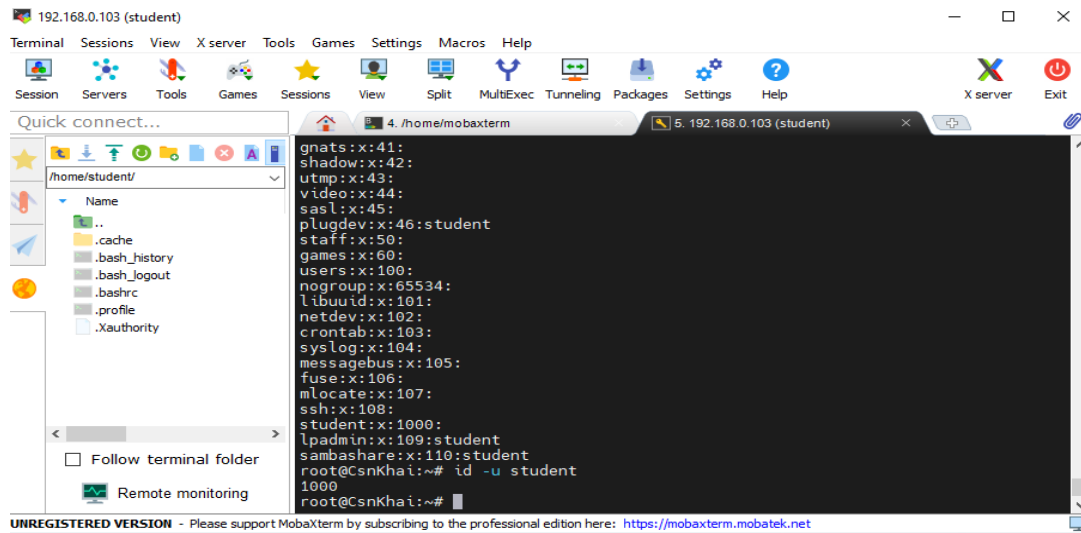
The screenshot shows a MobaXterm window with a terminal session. The terminal displays the contents of the `/etc/group` file. The output shows a list of system groups, including `root`, `daemon`, `bin`, `sys`, `adm`, `tty`, `disk`, `lp`, `mail`, `news`, `uucp`, `man`, `proxy`, `kmem`, `dialout`, `fax`, `voice`, `cdrom`, `floppy`, `tape`, `sudo`, `audio`, and `dip`. The fields in each line represent the group name, group password, GID, and a list of members.

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

UID — the user ID to which the process belongs.

On modern Linux systems, regular users have a UID with a four-digit number starting with 1000. Typically, 0 through 99 are reserved for system accounts.

\$ id -u username



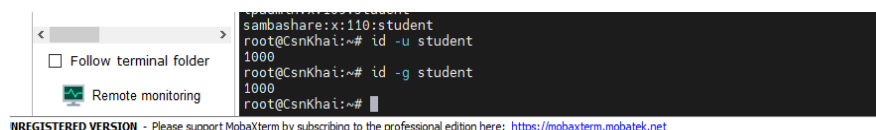
The screenshot shows a MobaXterm window with a terminal session. The terminal displays the output of the `id` command, listing system users and their UIDs. The output is as follows:

```
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:
libuid: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
root@CsnKhai:~# id -u student
1000
root@CsnKhai:~#
```

3) What is GID? How to define it?

GID (group ID) is a name that associates a system user with other users sharing something in common (perhaps a work project or a department name). A user can be a member of more than one group and thus have more than one GID.

\$ id -g student

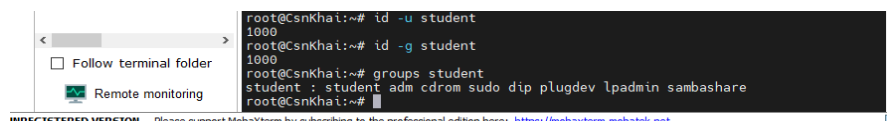


The screenshot shows a MobaXterm window with a terminal session. The terminal displays the output of the `id -g student` command, showing the group ID for the 'student' user.

```
sambashare:x:110:student
root@CsnKhai:~# id -u student
1000
root@CsnKhai:~# id -g student
1000
root@CsnKhai:~#
```

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

\$ groups student



The screenshot shows a MobaXterm window with a terminal session. The terminal displays the output of the `groups student` command, showing the groups the 'student' user belongs to.

```
root@CsnKhai:~# id -u student
1000
root@CsnKhai:~# id -g student
1000
root@CsnKhai:~# groups student
student : student adm cdrom sudo dip plugdev lpadmin sambashare
root@CsnKhai:~#
```

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

`useradd [options] user_name`

The most used parameters:

- b - create the user's home directory in the directory specified by default.
- e - account expiration date.
- g - specifies the primary group for the new user.
- G - specifies additional groups to which the user will belong.
- u - specifying the user ID.

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

\$ usermod -l

7) What is skell_dir? What is its structure?

The `/etc/skel` directory is the directory used by `useradd` to create the default settings in a new user's home directory. `Skell_dir` is a directory that contains files to be copied to the newly created directory.

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

First, you need to delete all user processes using the command:

```
$ pkill -KILL -u user_name
```

Then you need to delete the user account using the userdel command:

```
$ userdel -r user_name
```

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

How to lock users in Linux?

```
$ usermod -l user_name
```

How to unlock users in Linux?

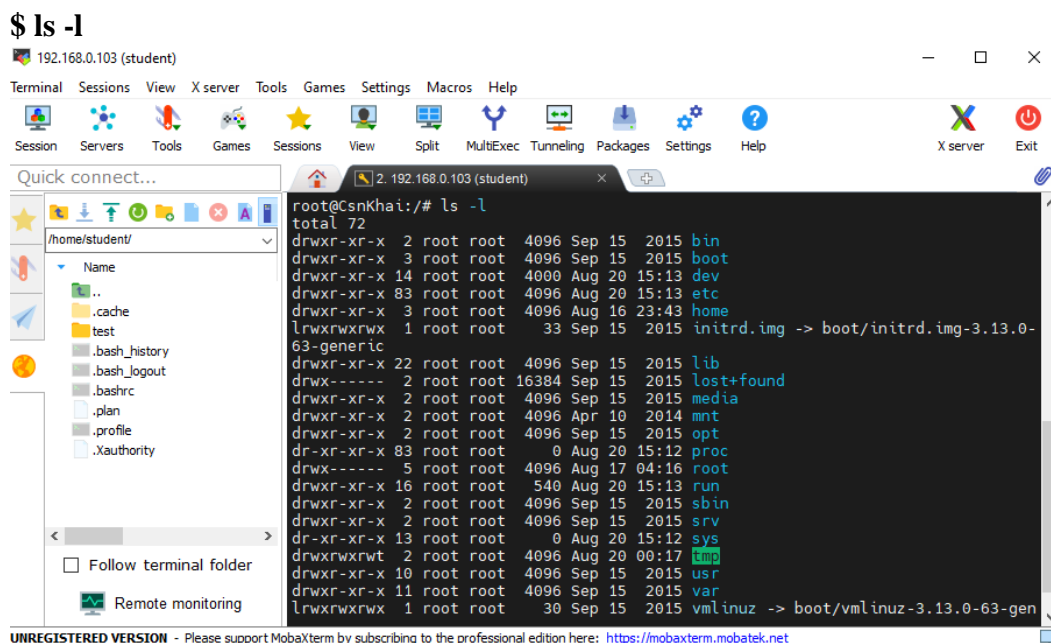
```
$ usermod -U user_name
```

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

```
$ passwd -d [user_name]
```

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

```
$ 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 Aug 20 15:13 dev
drwxr-xr-x 83 root root 4096 Aug 20 15:13 etc
drwxr-xr-x  3 root root 4096 Aug 16 23:43 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 83 root root   0 Aug 20 15:12 proc
drwx----- 5 root root 4096 Aug 17 04:16 root
drwxr-xr-x 16 root root 540 Aug 20 15:13 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 20 15:12 sys
drwxrwxrwt  2 root root 4096 Aug 20 00:17 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-gen
```

Information columns show: file type (- for a regular file, d for a directory, b for a block device, etc), file access rights, number of links to the file, owner name, group name, file size (in bytes), time stamp, and file name.

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

u - is short for "user", but means the owner of a file or directory.

g - owner group.

o - short for "others", but means everyone else.

a - short for "all". The combination of "u", "g" and "o".

r (read) - permission to read/view the file;

w (write) - permission to write/change a file;

x (execute) - permission to execute the file;

- - no permission set.

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.

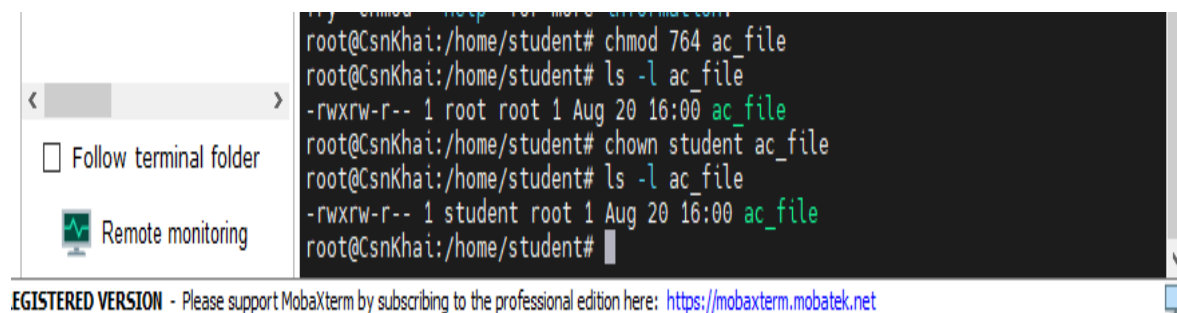
An example of changing file access rights:

\$ chmod 764 ac_file

The number 7 describes the permissions for the owner of the file, the number 6 indicates the access rights for the group and the 4 is the access rights for all other users. Accordingly, such an entry indicates that the owner of the file can read, modify, and run the file. All members of the group have access to read and make changes to the file, but cannot run it. Other users can only read the file.

An example of changing the owner of a file:

\$ chown student ac_file



The screenshot shows a terminal window with the following commands and output:

```
root@CsnKhai:/home/student# chmod 764 ac_file
root@CsnKhai:/home/student# ls -l ac_file
-rwxrw-r-- 1 root root 1 Aug 20 16:00 ac_file
root@CsnKhai:/home/student# chown student ac_file
root@CsnKhai:/home/student# ls -l ac_file
-rwxrw-r-- 1 student root 1 Aug 20 16:00 ac_file
root@CsnKhai:/home/student#
```

Below the terminal window, there is a banner for the MobaXterm software:

REGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

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

In numeric mode, a three-digit value represents specific file permissions (for example, 744.) These are called octal values. The first digit is for owner permissions, the second digit is for group permissions, and the third is for other users. Each permission has a numeric value assigned to it:

r (read): 4

w (write): 2

x (execute): 1

In the permission value 744, the first digit corresponds to the user, the second digit to the group, and the third digit to others. By adding up the value of each user classification, you can find the file permissions.

For example, a file might have read, write, and execute permissions for its owner, and only read permission for all other users. That looks like this:

Owner: $rw\text{x} = 4+2+1 = 7$

Group: $r\text{--} = 4+0+0 = 4$

Others: $r\text{--} = 4+0+0 = 4$

The results produce the three-digit value 744.

The **umask command** specifies the permissions that the user does not want to be given out to the newly created file or directory.

When creating any file, the operating system requests a permissions mask and calculates the mask based on it. By default, the mask is 0002. The first digit does not affect anything and is a relic of the syntax of the C language. Further numbers are similar to the access rights in Linux: the first is the owner, the second is the group and the third is everyone else. This mask is used to calculate file permissions. Without going into details, everything is calculated quite simply, the mask is taken away from the maximum rights and the rights for the file are obtained. In fact, it turns out that the mask contains permissions that will not be set for the file. Therefore, the default permissions for a file will be $666 - 002 = 664$, and for a directory - $777 - 002 = 775$.

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

When a directory's sticky bit is set, the file system treats the files in that directory as such that only the owner of the file, the owner of the directory, or root can rename or delete the file. It is usually installed in the temporary files folder (/tmp) to prevent normal users from deleting or moving other users' files. The sticky bit can be installed using the chmod command using its octal representation of 1000 or its symbol t.

Command substitution allows you to capture the output of any command as an argument to another command.

Command substitution is the mechanism by which the shell performs a given set of commands and then substitutes their output in the place of the commands.

The command substitution is performed when a command is given as:

`command`

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

The command script must have execute permission on the file (x):