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?

$ */etc/*passwd

*$ /etc/*group
group_name: раз sword:group_id:list

daemon Используется серверными процессами системы

bin Владеет исполняемыми файлами пользователя

sys Владеет системными файлами

adm Владеет файлами бюджета

uucp Используется UUCP

lp Используется подсистемами **1p** или **lpd**

nobody Используется NFS

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

UID - Identeficator of User
GID -Identificator of Group

To know uid of yourself:
$ id

```
test_user:x:1002:
amarocket@AmaRocket:~$ id
uid=1000(amarocket) gid=1000(amarocket) группы=1000(amarocket),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),120(lpadmin),131(lxd),132(sambashare)
amarocket@AmaRocket:~$
```

User uid:
$ id -u {username}

User Gid:

$ id -g {username}

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

$ id -G {Username}

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

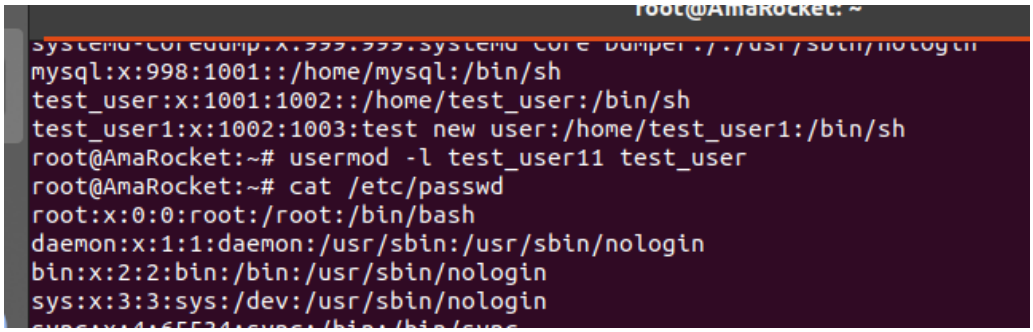
$ sudo useradd {options } {Username}
$ sudo passwd {Username}

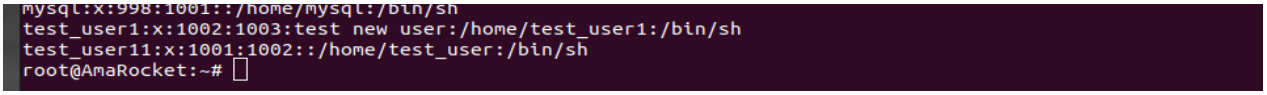
useradd {options } {Username}
passwd {Username}

We can add home direcory, group etc

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

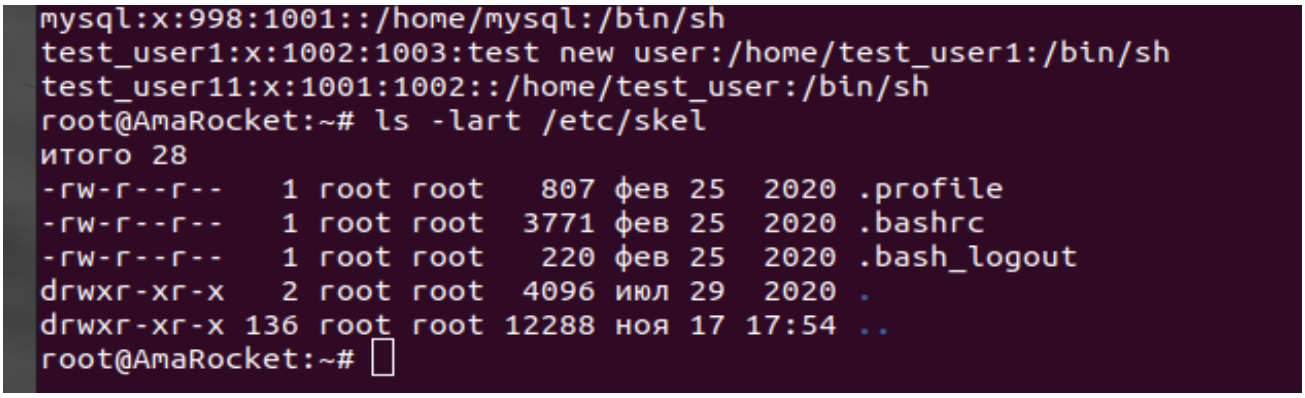
$ usermod -l {New_name} {Old-name}





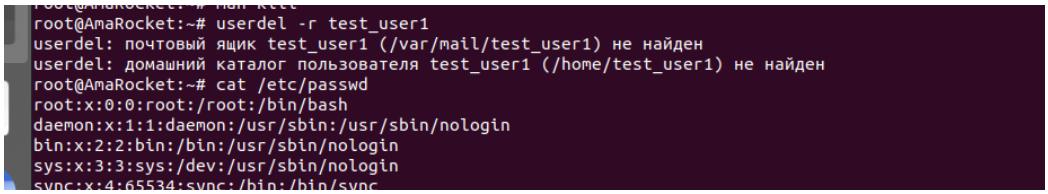
7) What is skell_dir? What is its structure?

$ ls -lart /etc/skel



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

userdel -r {Username}



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

passwd -l {Username} # Lock user
or:
usermod -l {Username}

passwd -u {Username} # Unlock user
or:
usermod -U username

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

passwd -d {Username}
usermod -s /usr/sbin/nologin {Username}

```
root@AmaRocket:~# passwd -d amarocket
passwd: информация об истечении срока действия пароля изменена.
root@AmaRocket:~# usermod -s /usr/sbin/nologin amarocket
```

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

$ ls -l

```
amarocket@AmaRocket:~$ htop
amarocket@AmaRocket:~$ ls -l
итого 88
drwxrwxr-x  3 amarocket amarocket  4096 ноя 13 20:19  CodeTrash
drwxrwxr-x  7 amarocket amarocket  4096 ноя 13 20:11  Console_app
drwxrwxr-x  6 amarocket amarocket  4096 окт 27 16:33  Django_pet
-rwxr-xr-x  1 amarocket amarocket    27 ноя  2 12:02  my-script.sh
drwxrwxr-x  2 amarocket amarocket  4096 ноя 17 02:20  NIX_Edu
drwx------ 12 amarocket amarocket  4096 ноя 17 10:46  snap
-rw-rw-r--  1 amarocket amarocket  3790 ноя  7 12:13  softserve_1
-rw-rw-r--  1 amarocket amarocket   866 ноя  7 12:12  softserve_2
drwxrwxr-x  4 amarocket amarocket  4096 ноя 17 13:56  SSA
drwxrwxr-x  4 amarocket amarocket  4096 ноя 15 20:01 'VirtualBox VMs'
drwxr-xr-x  2 amarocket amarocket  4096 июн 30  2020  Видео
drwxr-xr-x  2 amarocket amarocket  4096 июн 30  2020  Документы
drwx------  3 amarocket amarocket  4096 ноя 16 00:26  Загрузки
drwxr-xr-x  4 amarocket amarocket 20480 ноя 17 18:38  Изображения
drwxr-xr-x  2 amarocket amarocket  4096 июн 30  2020  Музыка
drwxr-xr-x  2 amarocket amarocket  4096 июн 30  2020  Общедоступные
drwxr-xr-x  5 amarocket amarocket  4096 ноя 17 15:52 'Рабочий стол'
drwxr-xr-x  2 amarocket amarocket  4096 июн 30  2020  Шаблоны
amarocket@AmaRocket:~$
```

- - Regular file.

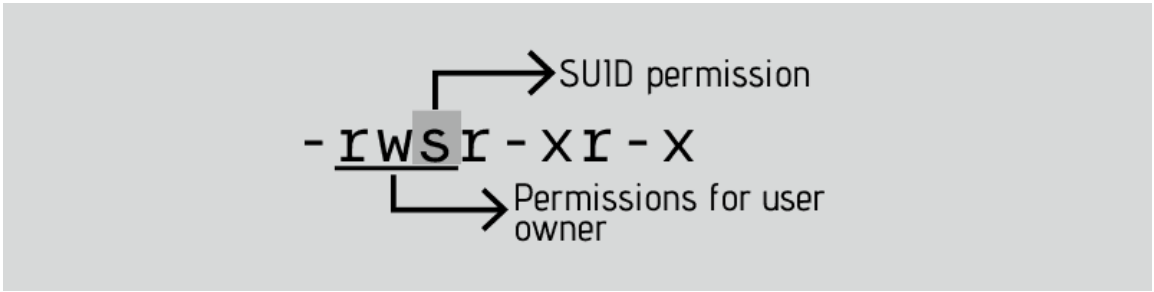
- b - Block special file.

- c - Character special file.
- d - Directory.
- l - Symbolic link.
- n - Network file.
- p - FIFO.
- s — Socket.
- r - Permission to read the file.
- w - Permission to write to the file.
- x - Permission to execute the file.
- s - setgid bit.
- t - sticky bit.

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

- **--- - нет прав, совсем;**
- **--x** - разрешено только выполнение файла, как программы но не изменение и не чтение;
- **-w-** - разрешена только запись и изменение файла;
- **-wx** - разрешено изменение и выполнение, но в случае с каталогом, вы не можете посмотреть его содержимое;
- **r--** - права только на чтение;
- **r-x** - только чтение и выполнение, без права на запись;
- **rw-** - права на чтение и запись, но без выполнения;
- **rwx** - все права;
- **--s** - установлен SUID или SGID бит, первый отображается в поле для владельца, второй для группы;
- **--t** - установлен sticky-bit, а значит пользователи не могут удалить этот файл.

SUID



- **SUID** - если этот бит установлен, то при выполнении программы, id пользователя, от которого она запущена заменяется на id владельца файла. Фактически, это позволяет обычным пользователям запускать программы от имени суперпользователя;
- **SGID** - этот флаг работает аналогичным образом, только разница в том, что пользователь считается членом группы, с которой связан файл, а не групп, к которым он действительно принадлежит. Если SGID флаг установлен на каталог, все файлы, созданные в нем, будут связаны с группой каталога, а не пользователя. Такое поведение используется для организации общих папок;
- **Sticky-bit** - этот бит тоже используется для создания общих папок. Если он установлен, то пользователи могут только создавать, читать и выполнять файлы, но не могут удалять файлы, принадлежащие другим пользователям.

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

Linux по отношению к любому файлу может выступать в трех ролях: как *хозяин* (user), как член *группы*, которой принадлежит файл (group), и как *посторонний* (other), никаких отношений собственности на этот файл не имеющий. Строка *атрибутов* – это три тройки "rwx", описывающие *права доступа* к файлу *хозяина* этого файла (первая тройка, "u"), *группы*, которой принадлежит файл (вторая тройка, "g") и *посторонних* (третья тройка, "o"). Если в какой-либо тройке не хватает буквы, а вместо нее стоит "-", значит, пользователю в соответствующей роли будет в соответствующем виде доступа отказано.

При выяснении отношений между файлом и пользователем, запустившим *процесс*, роль определяется так:

1. Если *UID* файла совпадает с *UID процесса*, пользователь – *хозяин файла*
2. Если *GID* файла совпадает с *GID* **любой** *группы*, в которую входит пользователь, он – член *группы*, которой принадлежит файл.
3. Если ни *UID*, ни *GID* файла не пересекаются с *UID процесса* и списком *групп*, в которые входит запустивший его пользователь, этот пользователь – *посторонний*.

Именно в роли *хозяина* пользователь (*процесс*) может **изменять** *ярлык файла*. Единственное, чего не может делать *хозяин* со своим файлом – менять ему *хозяина*.

+ *Права доступа* **изменяются** с помощью трех команд: chown (**ch**ange **own**er, сменить владельца), chgrp (**ch**ange **gr**oup, сменить *группу*) и chmod с расширенным форматом параметра: перед частью, определяющей **доступ** (перед знаком "+" или "-"), могут быть перечислены роли "u", "g", "o" и "a" (**a**ll, что соответствует "ugo"), доступ для которых изменяется. Кроме того, при задании доступа можно вместо "+" и "-" использовать "=", тогда для заданных ролей указанные способы доступа разрешаются, а **неуказанные** –

запрещаются. Вместо пары команд chown хозяин файл; chgrp группа файл можно применять одну: chown хозяин:группа файл, которая изменяет одновременно и *UID*, и *GID* файла (каталога, ссылки и т. п.).

Определение *прав доступа процесса* к объекту файловой системы (например, файла) происходит так. Используя *UID процесса*, список *групп*, в которые входит пользователь, запустивший этот *процесс*, *UID* файла и *GID* файла, система определяет **роль** *процесса* по отношению к файлу, а затем обращается к соответствующей тройке *атрибутов* файла. Процесс не может выступать сразу в **нескольких ролях**.

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.

$ chown

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

```
student@CsnKhai:~$ tree
.
└── 123456

0 directories, 1 file
student@CsnKhai:~$ pwd
/home/student
student@CsnKhai:~$ cat 123456
student@CsnKhai:~$ echo "Hop hey!" >> 123456
student@CsnKhai:~$ ls
123456
student@CsnKhai:~$ cat 123456
Hop hey!
student@CsnKhai:~$ mkdir NEW
student@CsnKhai:~$ ls
123456  NEW
student@CsnKhai:~$ sudo chown -R user:group /home/user
chown: invalid user: ‘user:group’
student@CsnKhai:~$ sudo chown -R student:student/home
chown: missing operand after ‘student:student/home’
Try 'chown --help' for more information.
student@CsnKhai:~$
student@CsnKhai:~$ umask
0002
student@CsnKhai:~$ umask -s
-bash: umask: -s: invalid option
umask: usage: umask [-p] [-S] [mode]
student@CsnKhai:~$ umask -S
u=rwx,g=rwx,o=rx
student@CsnKhai:~$
```

Umask цифра	разрешения по умолчанию для файлов	разрешения по умолчанию для каталога
	rw	rwx
1	rw	rw
2	r	rx
3	r	r
4	w	wx
5	w	w
6	x	x
7	не разрешено	не разрешено

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

```
student@CsnKhai: /opt/dump$ cd
student@CsnKhai:~$ ls -l
total 16
-rw-rw-r-- 1 student student    9 Nov 17 18:54 123456
drwxrwxr-x 2 student student 4096 Nov 17 19:26 dump
drwxrwxr-x 2 student student 4096 Nov 17 18:55 NEW
drwxrwxr-x 3 student student 4096 Nov 17 19:26 opt
student@CsnKhai:~$ ls
123456  dump  NEW  opt
student@CsnKhai:~$ chmod o+t 123456
student@CsnKhai:~$ ls -l
total 16
-rw-rw-r-T 1 student student    9 Nov 17 18:54 123456
drwxrwxr-x 2 student student 4096 Nov 17 19:26 dump
drwxrwxr-x 2 student student 4096 Nov 17 18:55 NEW
drwxrwxr-x 3 student student 4096 Nov 17 19:26 opt
student@CsnKhai:~$
```

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