**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?**

**For the /etc/passwd**

The / etc / passwd file contains a list of users known to the system. IN

During user registration, the system looks at this file to find the user ID and home directory.

Each line of the file describes one user and 7 fields separated by colons:

Registration name. Login names must be unique and contain no more than 32 characters.

They can contain any characters other than colon and newline character. They don't have to start with a number.

User ID is a 32-bit integer that uniquely identifies a user on the system.

By default, IDs less than 500 are reserved for service accounts. Regular users have IDs starting at 500.

Default group ID.

Like the User ID, the Group ID (GID) is a 32-bit integer.

Personal data field. The GECOS field is mainly used to store personal information about each user. It has no well-defined syntax.

home directory. Once logged in, the user is taken to their home directory. If this directory does not exist at the time of registration, a message about its absence is displayed.

Command interpreter.

- As the login shell, as a rule, the command interpreter is specified

**For the /etc/group**

The / etc / group file contains the names of the groups present in Linux and lists the members of each group.

structure / etc / group

Each entry in the / etc / group file represents one group and contains 4 fields:

Group name. By default, when a new user is created, his group is also created with the same name as the user's login name.

An encrypted password or an x ​​to indicate the use of the / etc / gshadow file;

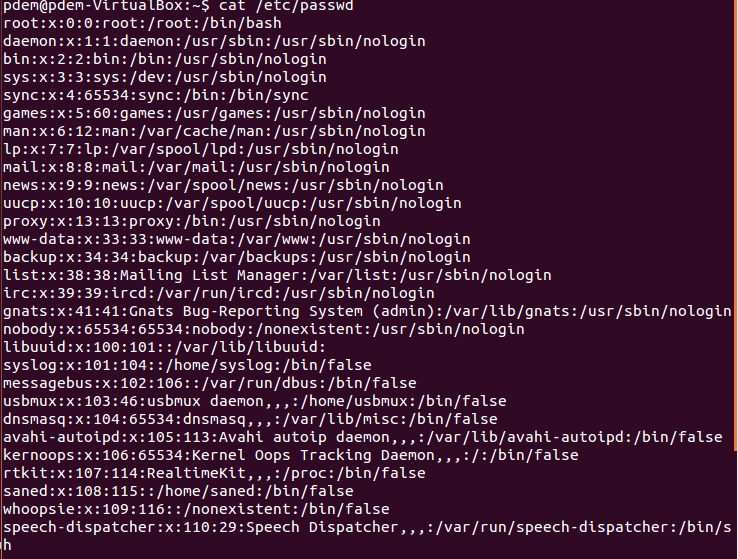
Group ID. (number)

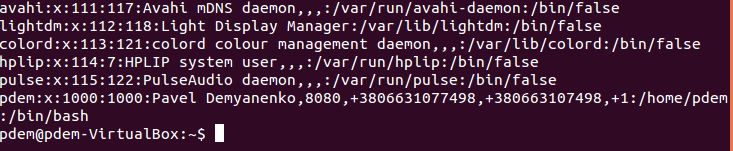
A comma-separated list of members without spaces.

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

Uid ranks are access rights and what a particular user can do. zero rank available to root

In Unix-like systems, users interact with users .

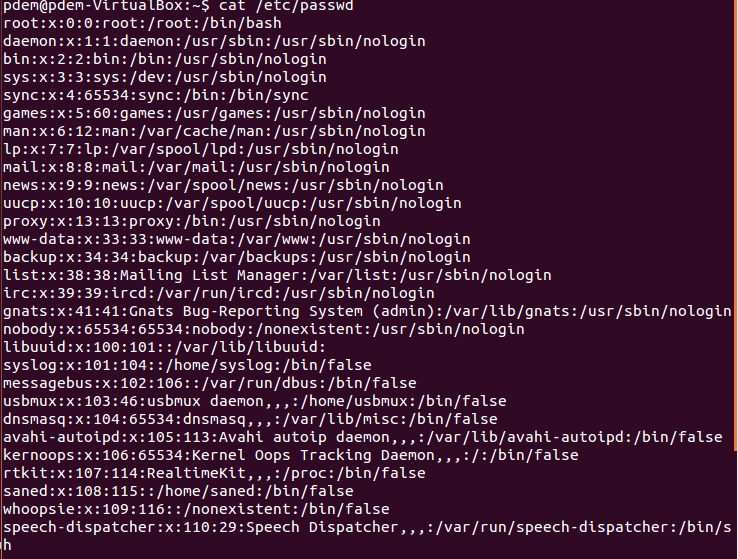


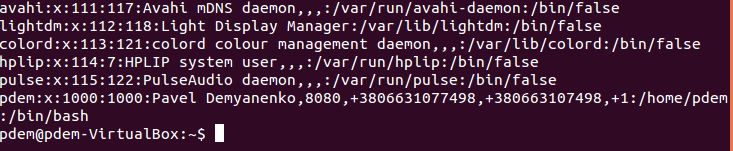


The third field here represents the user ID or UID.

**3) What is GID? How to define it?**

GID-group id. determines which resources a particular group can access





The 4th field here represents the user GID.

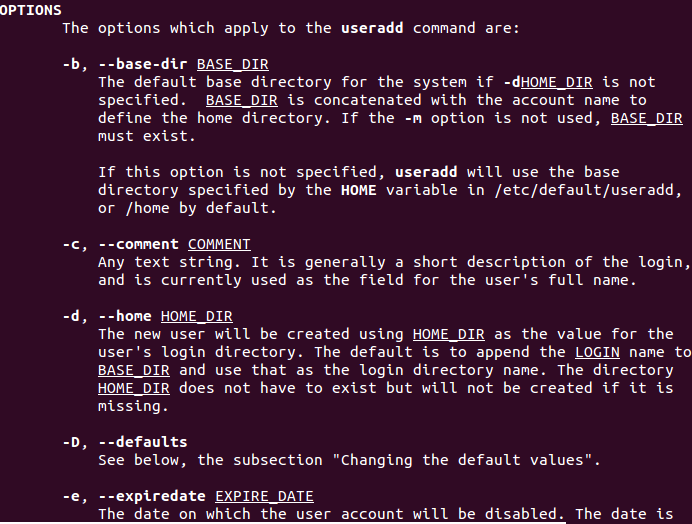
**4) How to determine belonging of user to the specific group?**

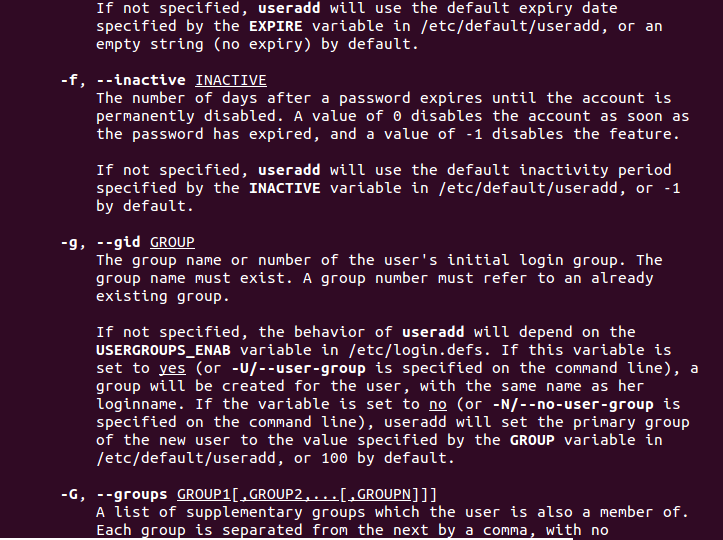
id -Gn username

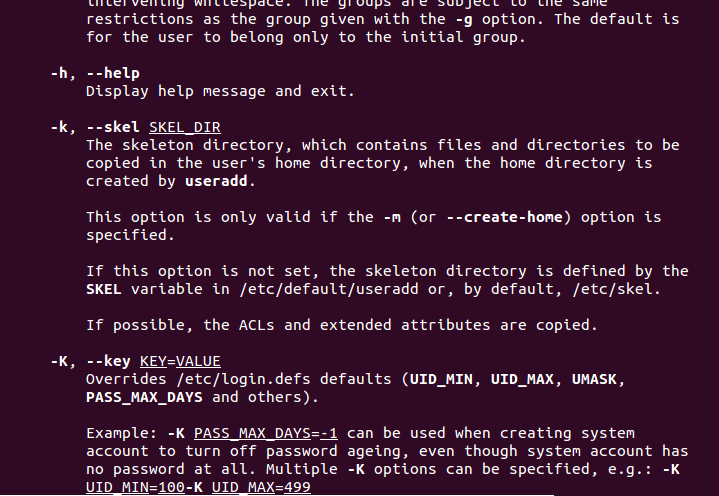
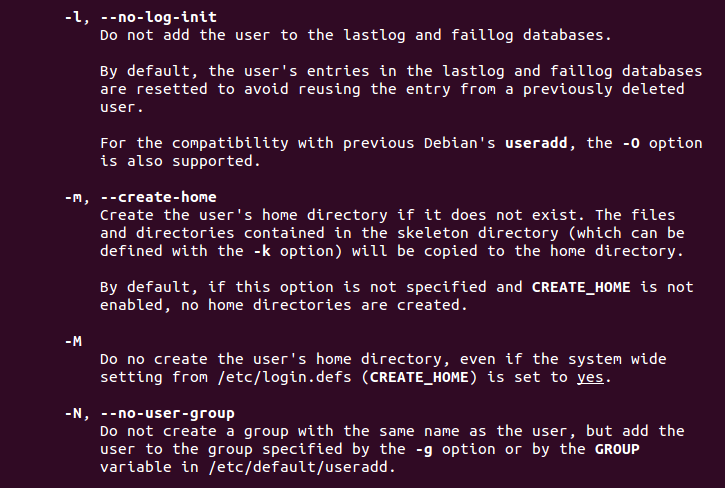
**5) What are the commands for adding a user to the system? What are the basic**

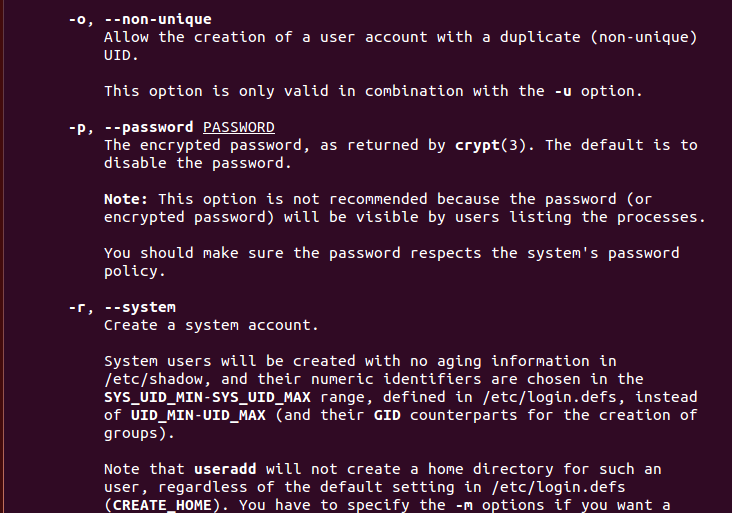
**parameters required to create a user?**

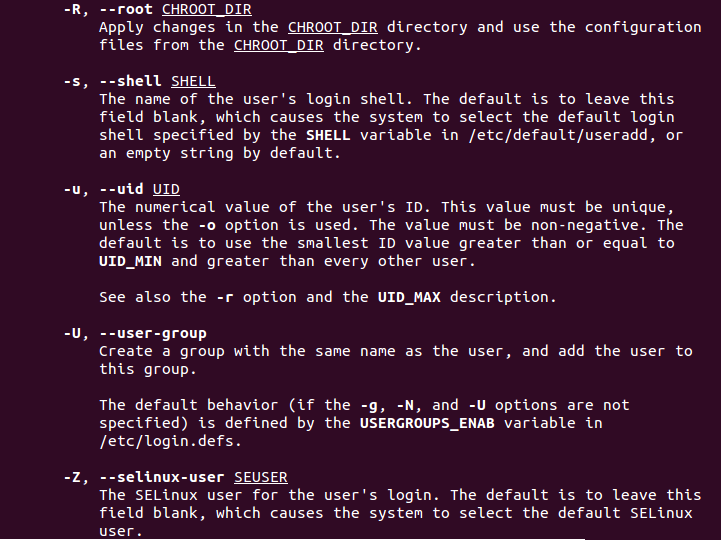
Sudo useradd [options] username





\ 





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

usermod -d /home/lastusername -m -g lastusername -l lastusername newusername

А без костылей, насколько понял, никак.

**7) What is skell\_dir? What is its structure?**

skell\_dir is a directory containing files to copy to a newly created custom directory.

-uid is a unique identifier for the user.

-m - Create a source directory (add) or move the files in the current home directory to a new home directory (mod).

-o - Allows the command to run successfully, even if the user ID is not unique.

-g - select the primary group for the user

-G - Selects secondary groups for the user.

-r - removes the user's home directory.

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

userdel -Z -r -f username

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

passwd -l user\_name

passwd -u user\_name

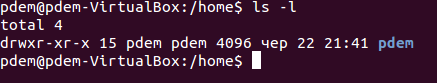
**10) How to remove a user's password and provide him with a password-free**

**login for subsequent password change?**

Sudo Passwd --expire username

**11) Display the extended format of information about the directory, tell about**

**the information columns displayed on the terminal.**



In the description of each file, the first character indicates the type of the file object, followed by the rights

In the second column, the number indicates the number of hard links for the files.

Next are the names of the owner, group, size, date of last modification, and file name.

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

**Briefly describe the acronym for access rights.**

Initially, each file had three access parameters. Here they are: Read - allows you to get the contents of the file, but not for writing. For a directory, allows you to get a list of files and directories located in it; Write - allows you to write new data to a file or modify existing ones, and also allows you to create and modify files and directories; Execution - You cannot execute a program if it does not have an execution flag. This attribute is set for all programs and scripts, it is with the help of it that the system can understand that this file needs to be run as a program. But all these rights would be meaningless if applied to all users at once. Therefore, each file has three categories of users, for which you can set various combinations of access rights: Owner - a set of rights for the owner of the file, the user who created it or is now set by its owner. Usually the owner has all the rights, read, write and execute. Group - any user group that exists in the system and is associated with a file. However, this can only be one group, and it is usually the owner's group, although a different group can be assigned to the file. Others - all users, except for the owner and users included in the file group. It is with the help of these sets of permissions that the permissions of files in linux are established.-rwxr-xr-x

The owner of the file can read, modify, and execute the file. Group users and all other users can read and execute this file.

-rw -------

The owner of the file can read and modify the file. No other users have access to the file.

-rwxrwxrwx

All users have the right to read, write and execute the file.

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

**user?**

chmod [права доступа] имя файла

For example, if we want to change the permissions on file1.txt to these:

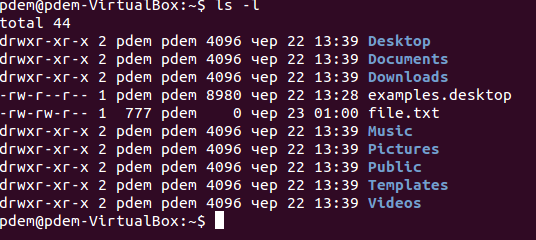
-rwxr - rw- 1 user user 0 Jun 22 11:53 pm file1.txt

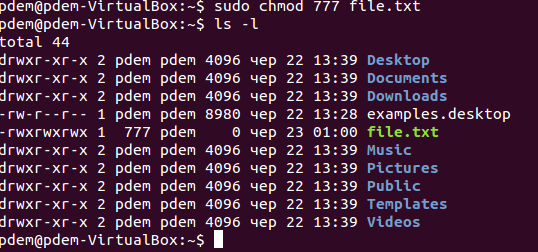
chmod 746 file1.txt

**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.**



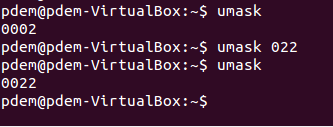




**What is an example of octal representation of access rights? Describe the**

**umask command.**





Насколько понял , меняет права на те что попросил, и когда создаешь новый файл/директорию, то она уже идет с такими же правами

**16) Give definitions of sticky bits and mechanism of identifier substitution. Give**

**an example of files and directories with these attributes.**

For the first time, the sticky nozzle is being used to reduce the time of the most frequently used software. After closing, the code and data remained in memory, and the next run was faster than the program.

Usually, when a user launches a program for execution, this program gets the same permissions to files and directories that the user who launched the program has. If the "user ID change bit" is set, then the program will receive access rights to files and directories that the owner of the program file has



В конкретном случае, буква t указывает, что у данной директории есть sticky bit , такие директории, может далять только owner.

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

All these objects use the same name system and have identical attributes that characterize access rights.

The file name and its attributes (type of object, access rights, information about its location on the carrier, etc.) are stored independently of each other. Names are stored in directories, and attributes are stored in special structures - inodes.