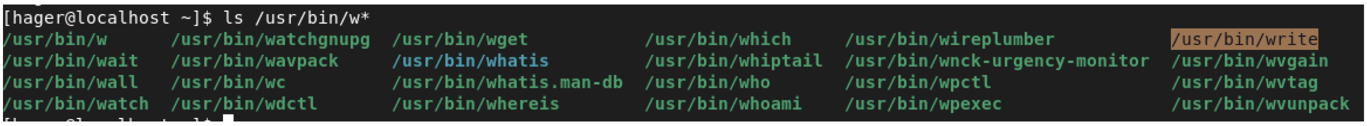
complete lab1

📌8-List Linux commands in /usr/bin that start with letter w

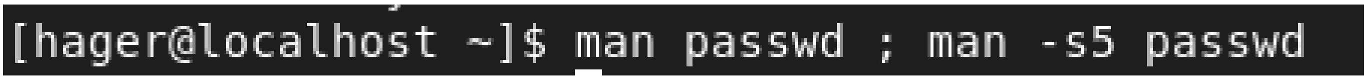
📸



```
[hager@localhost ~]$ ls /usr/bin/w*
/usr/bin/w      /usr/bin/watchgnupg  /usr/bin/wget          /usr/bin/which     /usr/bin/wireplumber          /usr/bin/write
/usr/bin/wait   /usr/bin/wavpack     /usr/bin/whatis        /usr/bin/whiptail  /usr/bin/wnck-urgency-monitor  /usr/bin/wvgain
/usr/bin/wall   /usr/bin/wc          /usr/bin/whatis.man-db /usr/bin/who       /usr/bin/wpctl                 /usr/bin/wvtag
/usr/bin/watch  /usr/bin/wdctl       /usr/bin/whereis       /usr/bin/whoami    /usr/bin/wpexec                /usr/bin/wvunpack
```

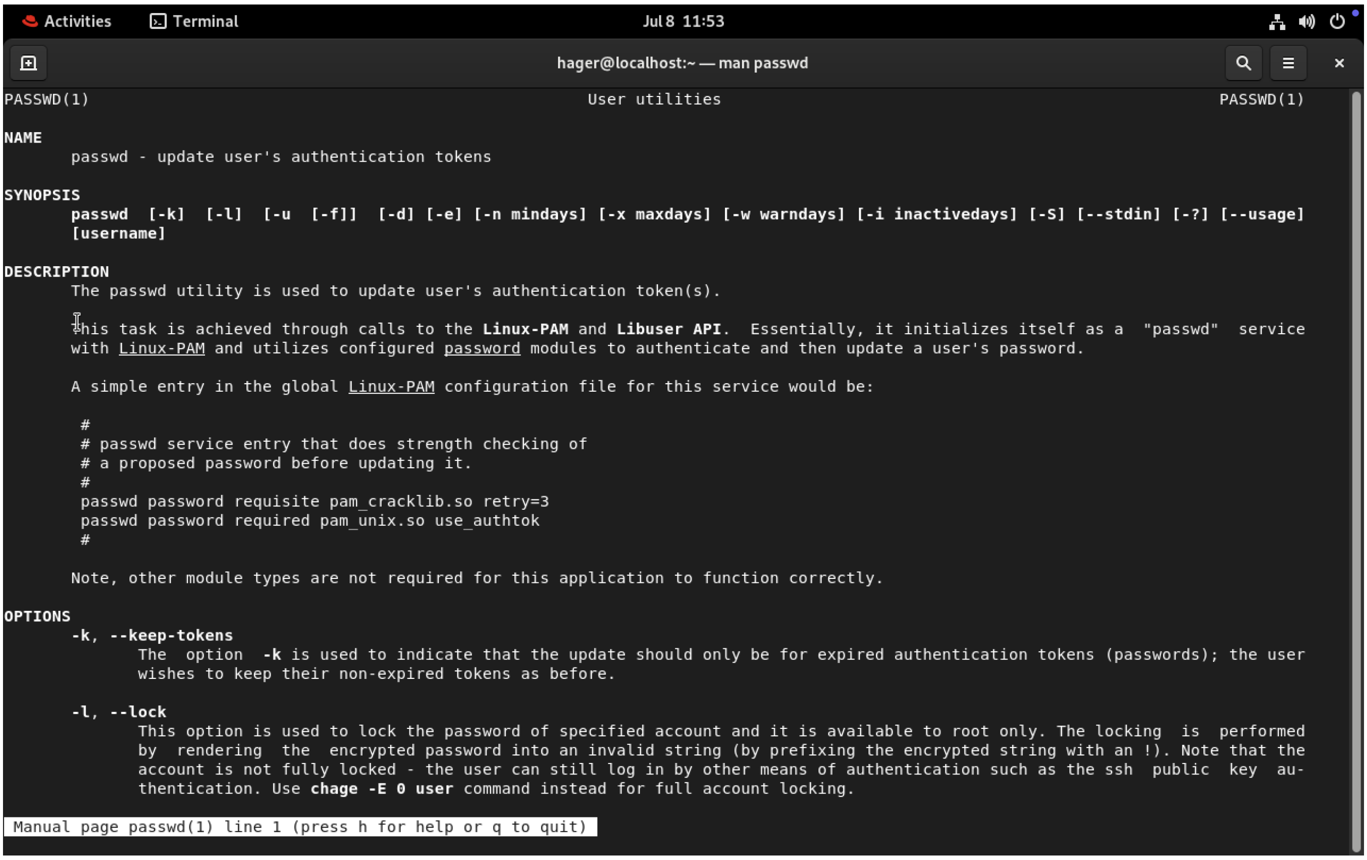
📌11-Display the man pages of passwd the command and the file sequentially in one command.

📸



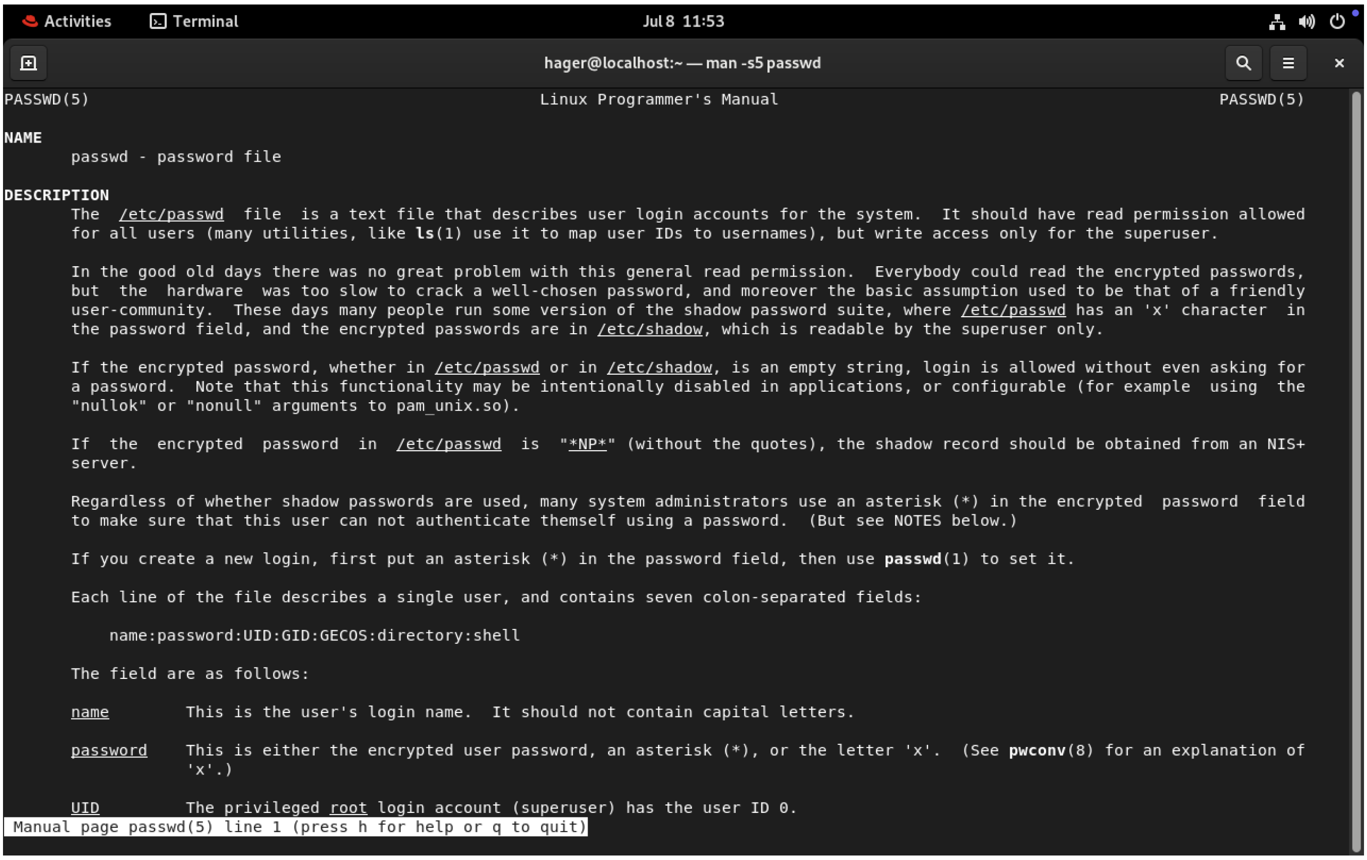
```
[hager@localhost ~]$ man passwd ; man -s5 passwd
```

📸



```
PASSWD(1)                                  User utilities                                  PASSWD(1)

NAME
       passwd - update user's authentication tokens

SYNOPSIS
       passwd  [-k]  [-l]  [-u  [-f]]  [-d] [-e] [-n mindays] [-x maxdays] [-w warndays] [-i inactivedays] [-S] [--stdin] [-?] [--usage]
       [username]

DESCRIPTION
       The passwd utility is used to update user's authentication token(s).

       This task is achieved through calls to the Linux-PAM and Libuser API.  Essentially, it initializes itself as a  "passwd"  service
       with Linux-PAM and utilizes configured password modules to authenticate and then update a user's password.

       A simple entry in the global Linux-PAM configuration file for this service would be:

        #
        # passwd service entry that does strength checking of
        # a proposed password before updating it.
        #
        passwd password requisite pam_cracklib.so retry=3
        passwd password required pam_unix.so use_authtok
        #

       Note, other module types are not required for this application to function correctly.

OPTIONS
       -k, --keep-tokens
              The  option  -k is used to indicate that the update should only be for expired authentication tokens (passwords); the user
              wishes to keep their non-expired tokens as before.

       -l, --lock
              This option is used to lock the password of specified account and it is available to root only. The locking  is  performed
              by  rendering  the  encrypted password into an invalid string (by prefixing the encrypted string with an !). Note that the
              account is not fully locked - the user can still log in by other means of authentication such as the ssh  public  key  au-
              thentication. Use chage -E 0 user command instead for full account locking.

 Manual page passwd(1) line 1 (press h for help or q to quit)
```

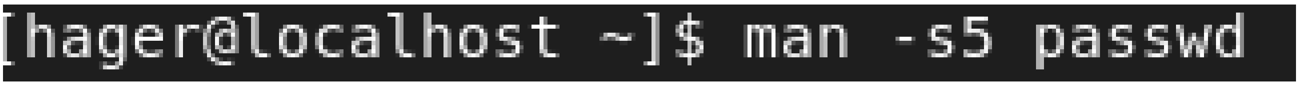
📸



```
PASSWD(5)                         Linux Programmer's Manual                         PASSWD(5)

NAME
       passwd - password file

DESCRIPTION
       The  /etc/passwd  file  is a text file that describes user login accounts for the system.  It should have read permission allowed
       for all users (many utilities, like ls(1) use it to map user IDs to usernames), but write access only for the superuser.

       In the good old days there was no great problem with this general read permission.  Everybody could read the encrypted passwords,
       but  the  hardware  was too slow to crack a well-chosen password, and moreover the basic assumption used to be that of a friendly
       user-community.  These days many people run some version of the shadow password suite, where /etc/passwd has an 'x' character  in
       the password field, and the encrypted passwords are in /etc/shadow, which is readable by the superuser only.

       If the encrypted password, whether in /etc/passwd or in /etc/shadow, is an empty string, login is allowed without even asking for
       a password.  Note that this functionality may be intentionally disabled in applications, or configurable (for example  using  the
       "nullok" or "nonull" arguments to pam_unix.so).

       If  the  encrypted  password  in  /etc/passwd  is  "*NP*" (without the quotes), the shadow record should be obtained from an NIS+
       server.

       Regardless of whether shadow passwords are used, many system administrators use an asterisk (*) in the encrypted  password  field
       to make sure that this user can not authenticate themself using a password.  (But see NOTES below.)

       If you create a new login, first put an asterisk (*) in the password field, then use passwd(1) to set it.

       Each line of the file describes a single user, and contains seven colon-separated fields:

           name:password:UID:GID:GECOS:directory:shell

       The field are as follows:

       name        This is the user's login name.  It should not contain capital letters.

       password    This is either the encrypted user password, an asterisk (*), or the letter 'x'.  (See pwconv(8) for an explanation of
                   'x'.)

       UID         The privileged root login account (superuser) has the user ID 0.
 Manual page passwd(5) line 1 (press h for help or q to quit)
```

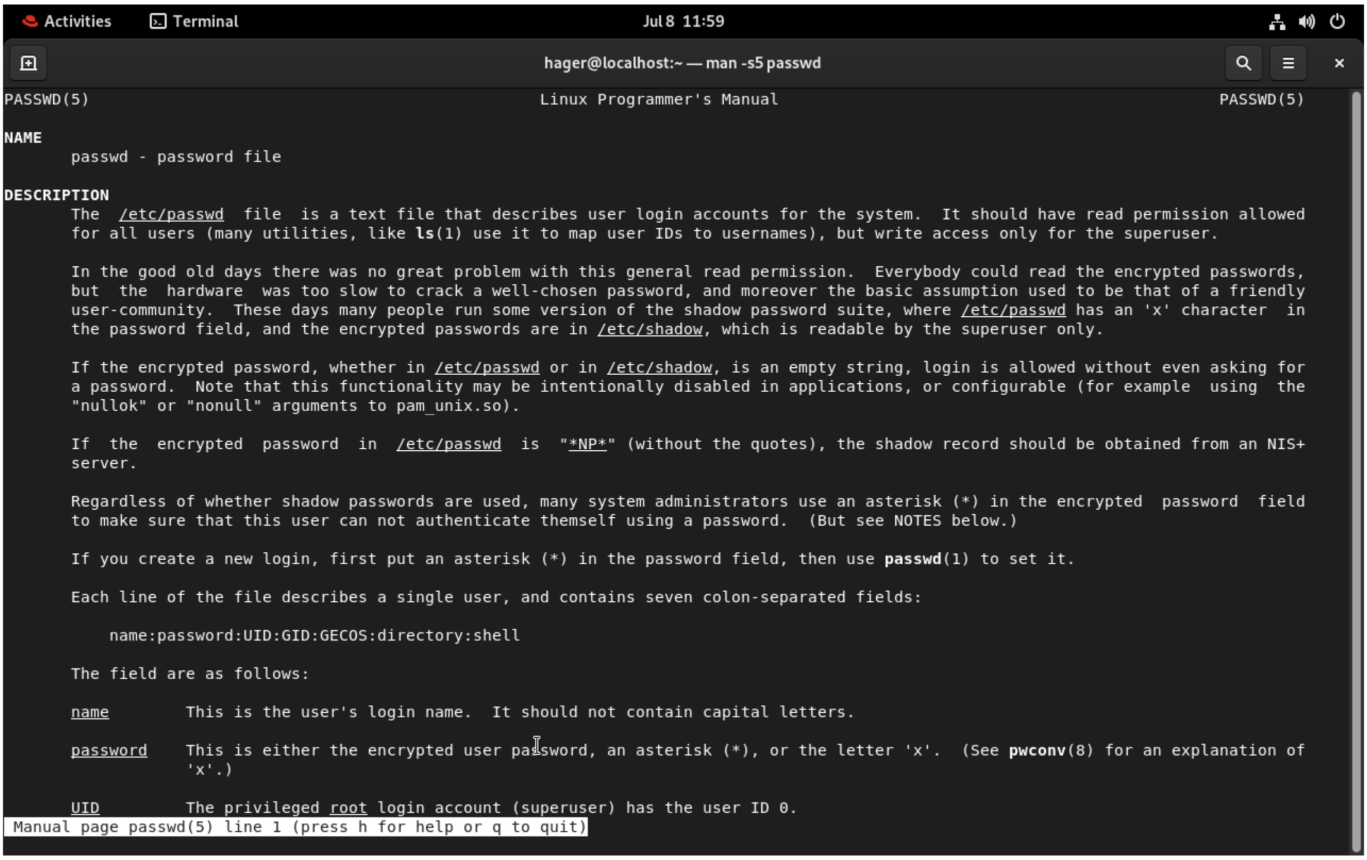
📌12- Display the man page of the passwd file.

📸



```
[hager@localhost ~]$ man -s5 passwd
```

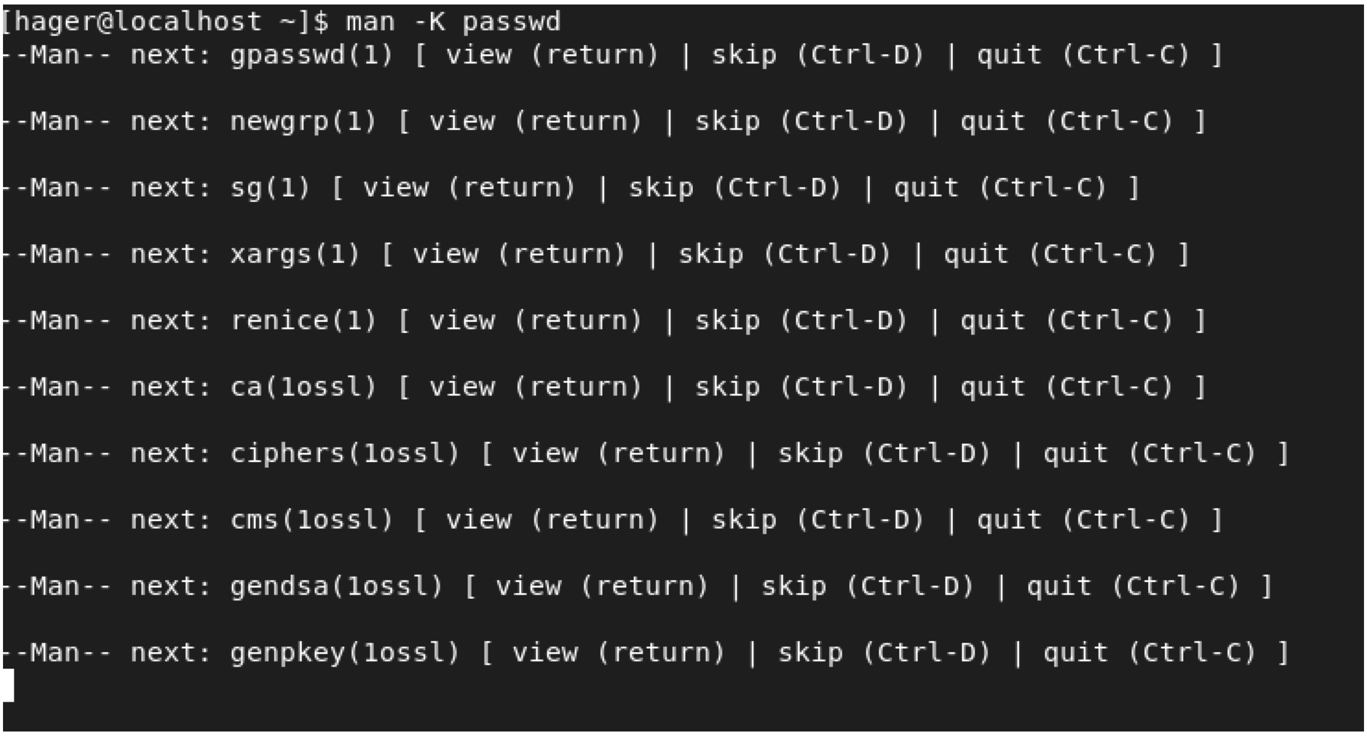
📸



```
PASSWD(5)                         Linux Programmer's Manual                         PASSWD(5)

NAME
       passwd - password file

DESCRIPTION
       The  /etc/passwd  file  is a text file that describes user login accounts for the system.  It should have read permission allowed
       for all users (many utilities, like ls(1) use it to map user IDs to usernames), but write access only for the superuser.

       In the good old days there was no great problem with this general read permission.  Everybody could read the encrypted passwords,
       but  the  hardware  was too slow to crack a well-chosen password, and moreover the basic assumption used to be that of a friendly
       user-community.  These days many people run some version of the shadow password suite, where /etc/passwd has an 'x' character  in
       the password field, and the encrypted passwords are in /etc/shadow, which is readable by the superuser only.

       If the encrypted password, whether in /etc/passwd or in /etc/shadow, is an empty string, login is allowed without even asking for
       a password.  Note that this functionality may be intentionally disabled in applications, or configurable (for example  using  the
       "nullok" or "nonull" arguments to pam_unix.so).

       If  the  encrypted  password  in  /etc/passwd  is  "*NP*" (without the quotes), the shadow record should be obtained from an NIS+
       server.

       Regardless of whether shadow passwords are used, many system administrators use an asterisk (*) in the encrypted  password  field
       to make sure that this user can not authenticate themself using a password.  (But see NOTES below.)

       If you create a new login, first put an asterisk (*) in the password field, then use passwd(1) to set it.

       Each line of the file describes a single user, and contains seven colon-separated fields:

           name:password:UID:GID:GECOS:directory:shell

       The field are as follows:

       name        This is the user's login name.  It should not contain capital letters.

       password    This is either the encrypted user password, an asterisk (*), or the letter 'x'.  (See pwconv(8) for an explanation of
                   'x'.)

       UID         The privileged root login account (superuser) has the user ID 0.
 Manual page passwd(5) line 1 (press h for help or q to quit)
```

📌13- Display a list of all the commands that contain the keyword passwd in their man page.

📸

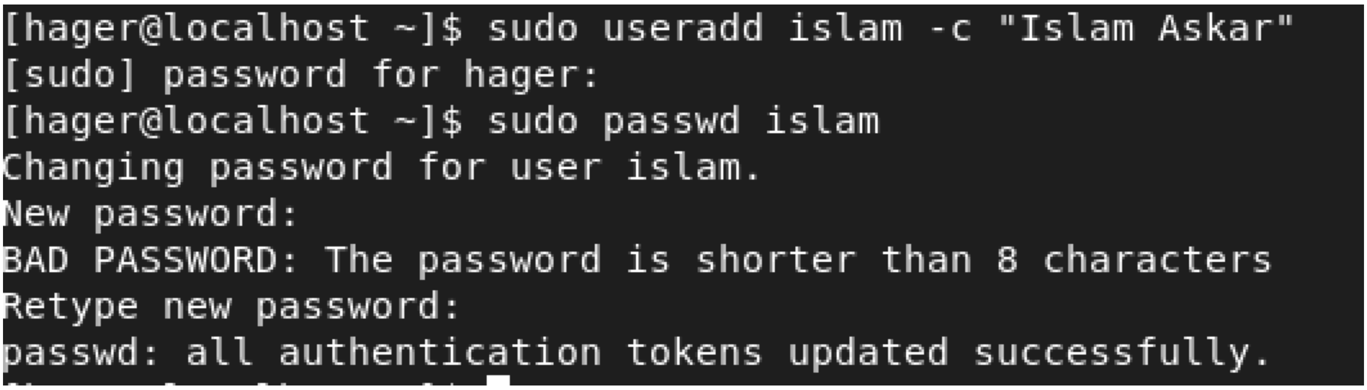


```
[hager@localhost ~]$ man -K passwd
--Man-- next: gpasswd(1) [ view (return) | skip (Ctrl-D) | quit (Ctrl-C) ]

--Man-- next: newgrp(1) [ view (return) | skip (Ctrl-D) | quit (Ctrl-C) ]

--Man-- next: sg(1) [ view (return) | skip (Ctrl-D) | quit (Ctrl-C) ]

--Man-- next: xargs(1) [ view (return) | skip (Ctrl-D) | quit (Ctrl-C) ]

--Man-- next: renice(1) [ view (return) | skip (Ctrl-D) | quit (Ctrl-C) ]

--Man-- next: ca(1ossl) [ view (return) | skip (Ctrl-D) | quit (Ctrl-C) ]

--Man-- next: ciphers(1ossl) [ view (return) | skip (Ctrl-D) | quit (Ctrl-C) ]

--Man-- next: cms(1ossl) [ view (return) | skip (Ctrl-D) | quit (Ctrl-C) ]

--Man-- next: gendsa(1ossl) [ view (return) | skip (Ctrl-D) | quit (Ctrl-C) ]

--Man-- next: genpkey(1ossl) [ view (return) | skip (Ctrl-D) | quit (Ctrl-C) ]
```

lab2

** 📌1-Create a user account with the following attribute username: islam Fullname/comment: Islam Askar Password: islam **

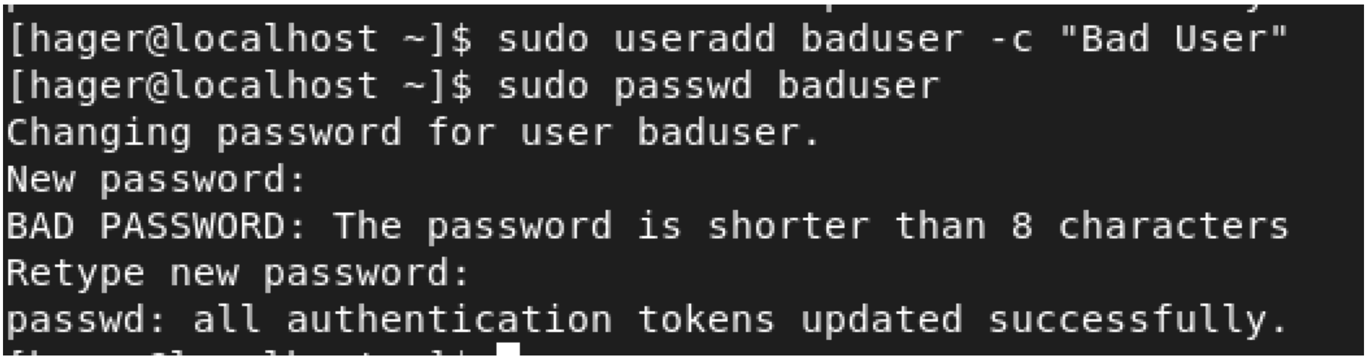
📸



```
[hager@localhost ~]$ sudo useradd islam -c "Islam Askar"
[sudo] password for hager:
[hager@localhost ~]$ sudo passwd islam
Changing password for user islam.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
```

📌2-Create a user account with the following attribute Username: baduser Full name/comment: Bad User Password: baduser

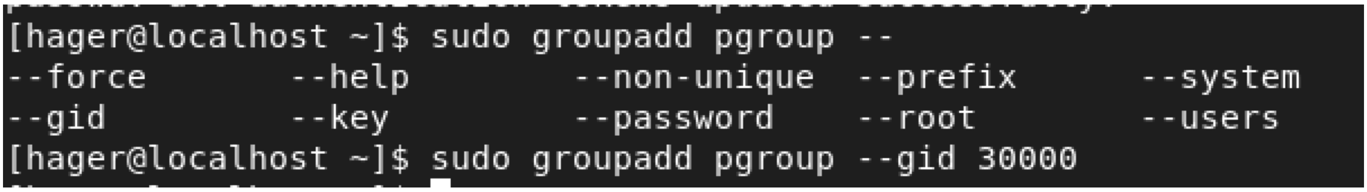
📸



```
[hager@localhost ~]$ sudo useradd baduser -c "Bad User"
[hager@localhost ~]$ sudo passwd baduser
Changing password for user baduser.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
```

📌3-Create a supplementary (Secondary) group called pgroup with group ID of 30000

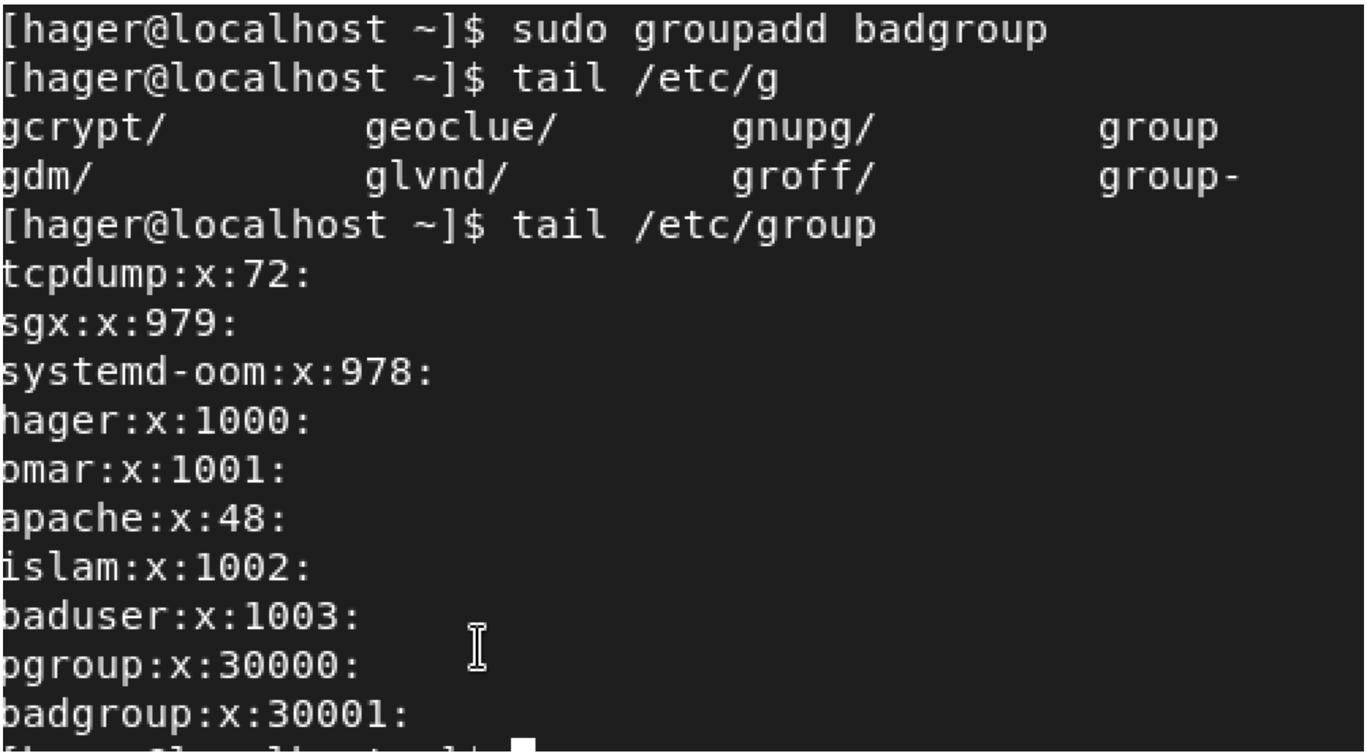
📸



```
[hager@localhost ~]$ sudo groupadd pgroup --
--force       --help        --non-unique  --prefix      --system
--gid         --key         --password    --root        --users
[hager@localhost ~]$ sudo groupadd pgroup --gid 30000
```

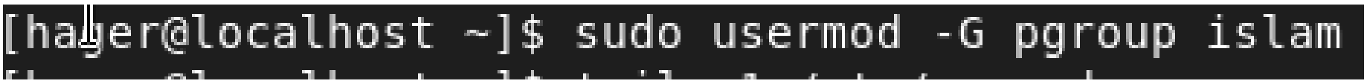
📌4-Create a supplementary group called badgroup

📸

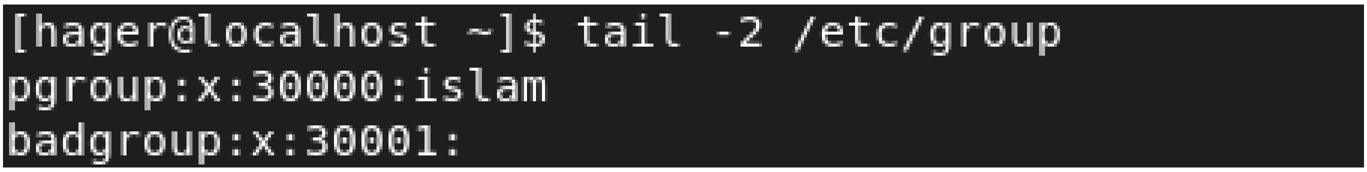


```
[hager@localhost ~]$ sudo groupadd badgroup
[hager@localhost ~]$ tail /etc/g
gcrypt/         geoclue/        gnupg/          group
gdm/            glvnd/          groff/          group-
[hager@localhost ~]$ tail /etc/group
tcpdump:x:72:
sgx:x:979:
systemd-oom:x:978:
hager:x:1000:
omar:x:1001:
apache:x:48:
islam:x:1002:
baduser:x:1003:
pgroup:x:30000:
badgroup:x:30001:
```

📌5-Add islam user to the pgroup group as a supplementary group



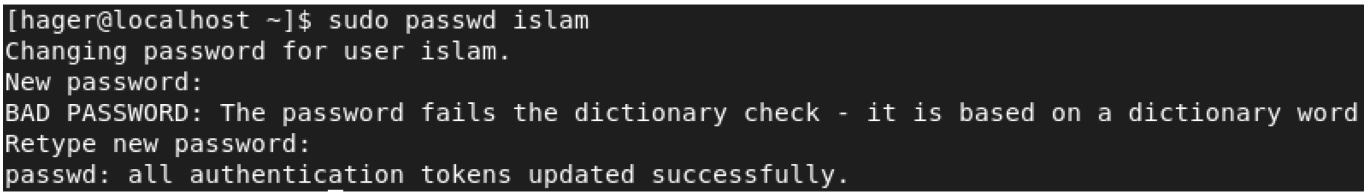
```
[hager@localhost ~]$ sudo usermod -G pgroup islam
```



```
[hager@localhost ~]$ tail -2 /etc/group
pgroup:x:30000:islam
badgroup:x:30001:
```

📌6-Modify the password of islam's account to password

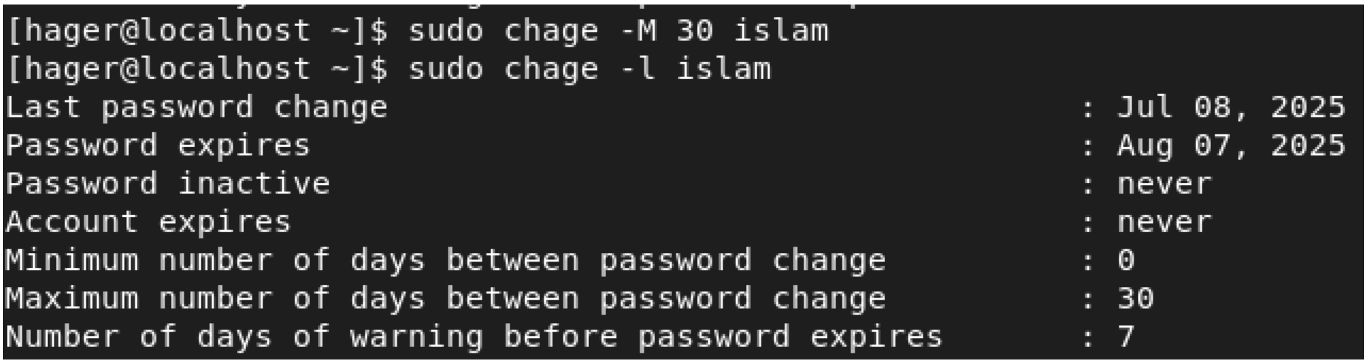
📸



```
[hager@localhost ~]$ sudo passwd islam
Changing password for user islam.
New password:
BAD PASSWORD: The password fails the dictionary check - it is based on a dictionary word
Retype new password:
passwd: all authentication tokens updated successfully.
```

📌7-Modify islam's account so the password expires after 30 days

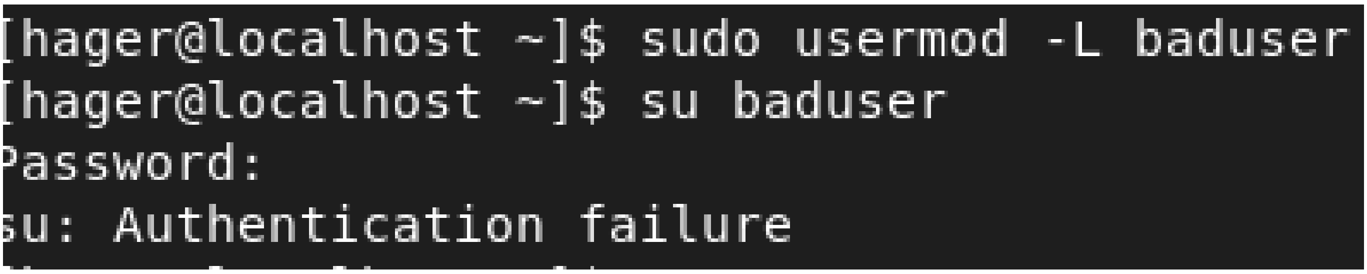
📸



```
[hager@localhost ~]$ sudo chage -M 30 islam
[hager@localhost ~]$ sudo chage -l islam
Last password change                                    : Jul 08, 2025
Password expires                                        : Aug 07, 2025
Password inactive                                       : never
Account expires                                         : never
Minimum number of days between password change          : 0
Maximum number of days between password change          : 30
Number of days of warning before password expires       : 7
```

📌8-Lock bad user account so he can't log in

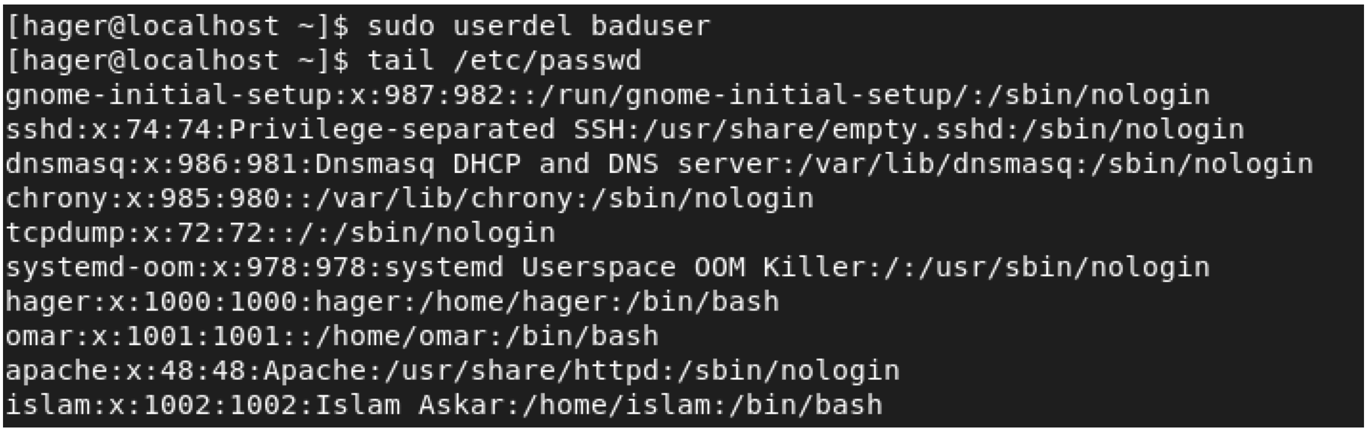
📸



```
[hager@localhost ~]$ sudo usermod -L baduser
[hager@localhost ~]$ su baduser
Password:
su: Authentication failure
```

📌9-Delete bad user account

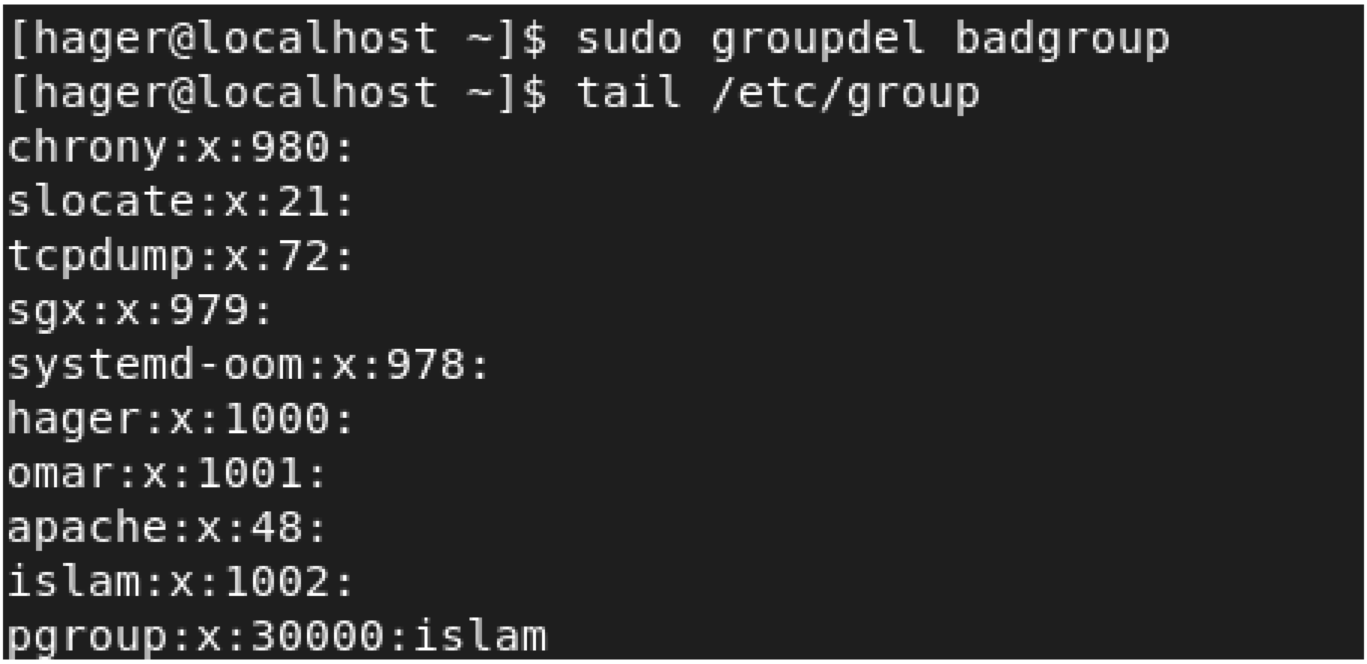
📸



```
[hager@localhost ~]$ sudo userdel baduser
[hager@localhost ~]$ tail /etc/passwd
gnome-initial-setup:x:987:982::/run/gnome-initial-setup/:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/usr/share/empty.sshd:/sbin/nologin
dnsmasq:x:986:981:Dnsmasq DHCP and DNS server:/var/lib/dnsmasq:/sbin/nologin
chrony:x:985:980::/var/lib/chrony:/sbin/nologin
tcpdump:x:72:72::/:/sbin/nologin
systemd-oom:x:978:978:systemd Userspace OOM Killer:/:/usr/sbin/nologin
hager:x:1000:1000:hager:/home/hager:/bin/bash
omar:x:1001:1001::/home/omar:/bin/bash
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
islam:x:1002:1002:Islam Askar:/home/islam:/bin/bash
```

📌10-Delete the supplementary group called badgroup.

📸



```
[hager@localhost ~]$ sudo groupdel badgroup
[hager@localhost ~]$ tail /etc/group
chrony:x:980:
slocate:x:21:
tcpdump:x:72:
sgx:x:979:
systemd-oom:x:978:
hager:x:1000:
omar:x:1001:
apache:x:48:
islam:x:1002:
pgroup:x:30000:islam
```