Linux Lab 1

Amgad Mohammed Mohammed Mohammed Shalaby

October 18, 2023

1. What is the difference between **cat** and **more** command?

- **cat** (short for concatenate) is used to display the entire content of a file, typically all at once. It simply outputs the contents of the file to the terminal, which may scroll quickly if the file is large.
- **more** is a pager program that allows you to view text one screen at a time. It's useful for reading through large files or long text output.

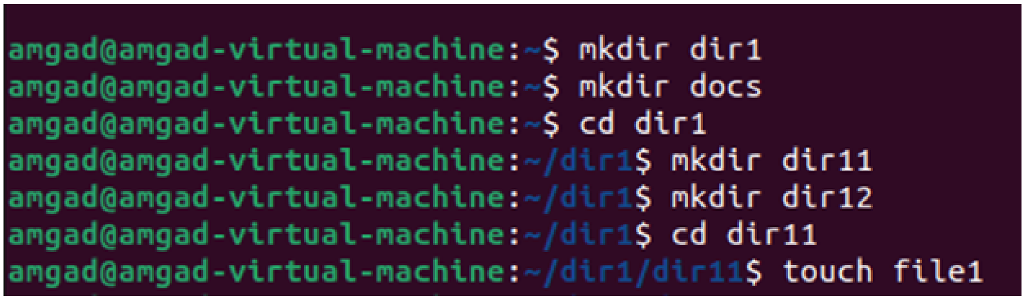
> - The primary difference is that **cat** displays the entire content at once, while **more** displays it in a paginated manner, allowing you to navigate through the content screen by screen using the Enter key or the spacebar.

2. What is the difference between **rm** and **rmdir** using man?

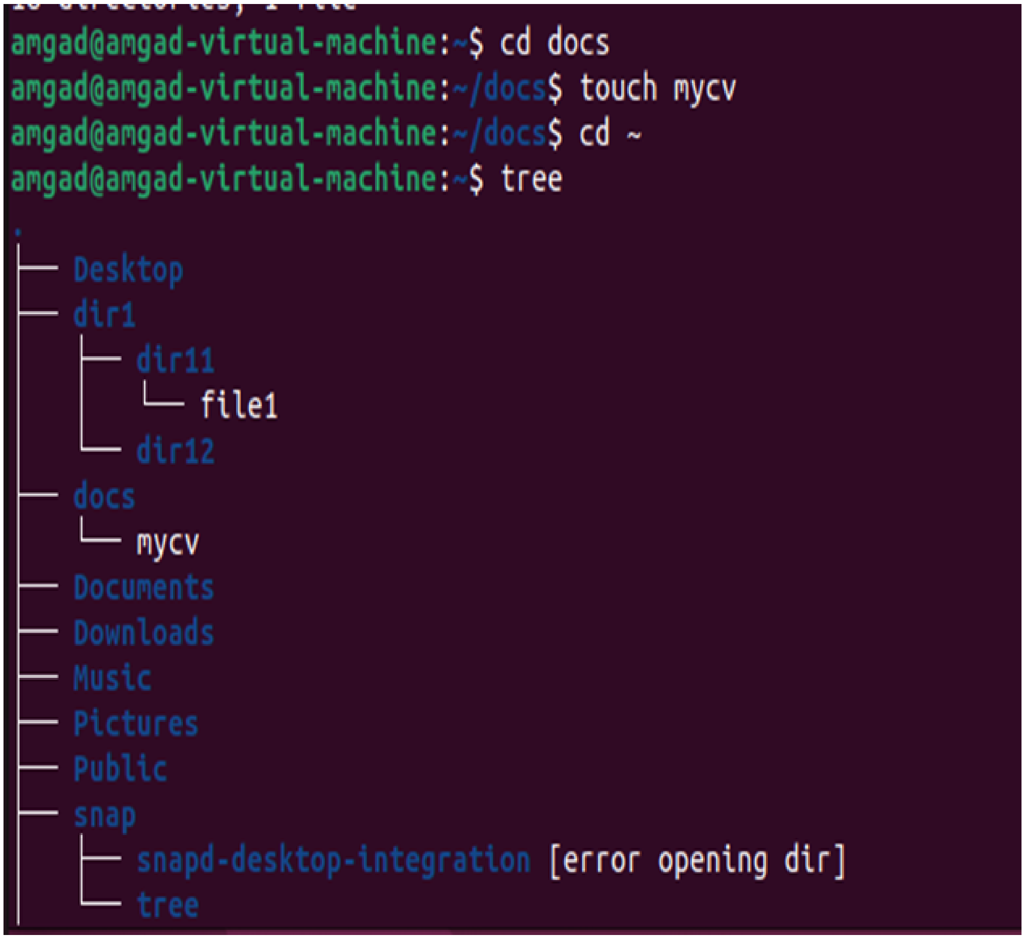
- **rm** (remove) is a command used to delete files and directories. When used with the **-r** or **-rf** option, it can delete directories and their contents recursively.
- **rmdir** is a command specifically used to remove empty directories. It won't remove directories with files or subdirectories in them.

> - The main difference is that **rm** is a more versatile command for removing files and directories, including non-empty directories, while **rmdir** is limited to removing only empty directories.

3. Create the following hierarchy under your home directory:

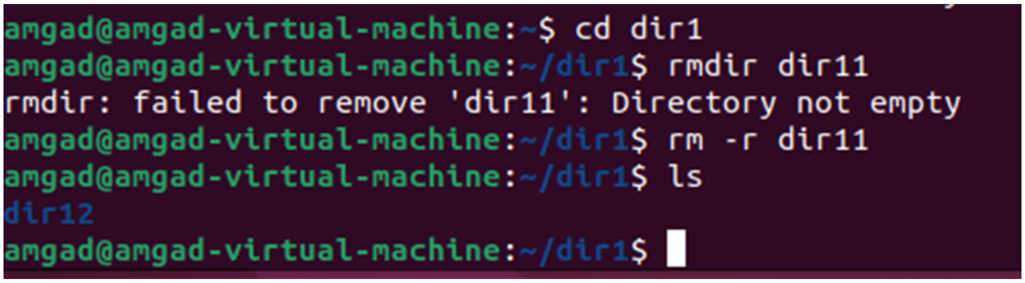


```
amgad@amgad-virtual-machine:~$ mkdir dir1
amgad@amgad-virtual-machine:~$ mkdir docs
amgad@amgad-virtual-machine:~$ cd dir1
amgad@amgad-virtual-machine:~/dir1$ mkdir dir11
amgad@amgad-virtual-machine:~/dir1$ mkdir dir12
amgad@amgad-virtual-machine:~/dir1$ cd dir11
amgad@amgad-virtual-machine:~/dir1/dir11$ touch file1
```



```
amgad@amgad-virtual-machine:~$ cd docs
amgad@amgad-virtual-machine:~/docs$ touch mycv
amgad@amgad-virtual-machine:~/docs$ cd ~
amgad@amgad-virtual-machine:~$ tree
.
├── Desktop
├── dir1
│   ├── dir11
│   │   └── file1
│   └── dir12
├── docs
│   └── mycv
├── Documents
├── Downloads
├── Music
├── Pictures
├── Public
├── snap
│   ├── snapd-desktop-integration [error opening dir]
│   └── tree
```

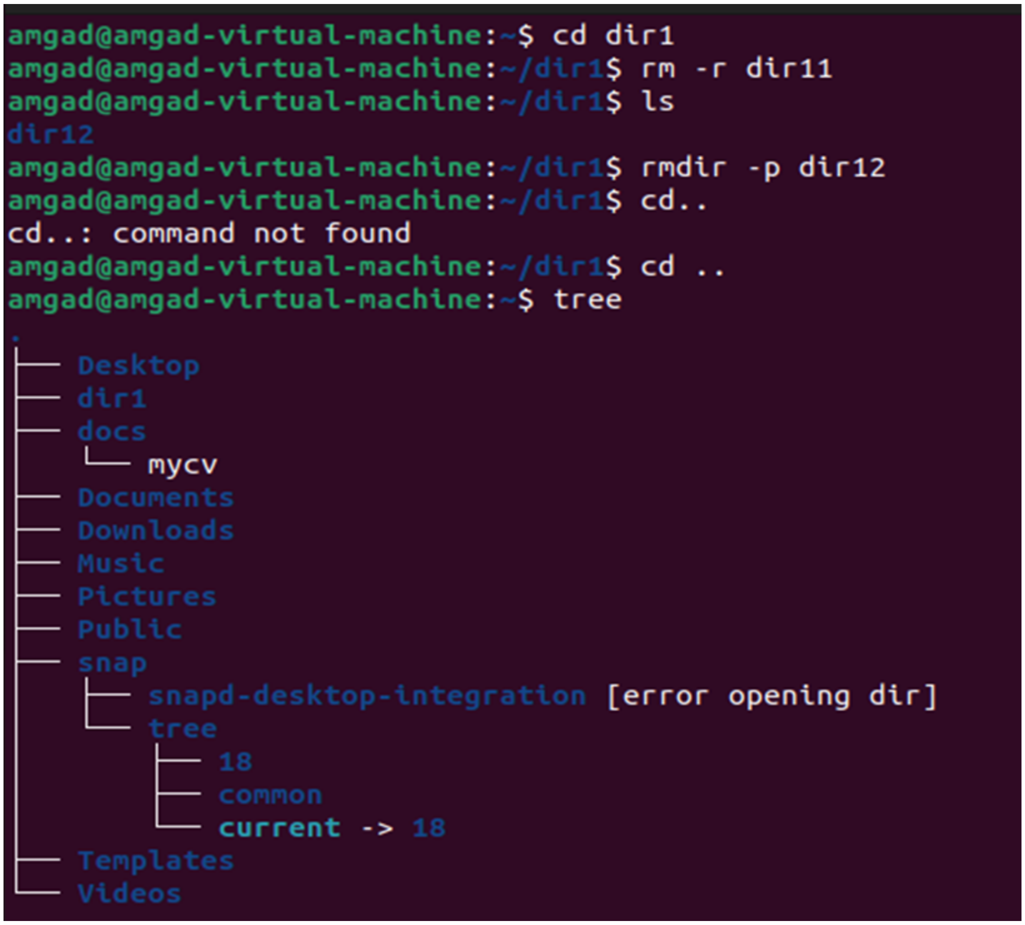
a) Remove **dir11** in one-step. What did you notice? And how did you overcome that?



```
amgad@amgad-virtual-machine:~$ cd dir1
amgad@amgad-virtual-machine:~/dir1$ rmdir dir11
rmdir: failed to remove 'dir11': Directory not empty
amgad@amgad-virtual-machine:~/dir1$ rm -r dir11
amgad@amgad-virtual-machine:~/dir1$ ls
dir12
amgad@amgad-virtual-machine:~/dir1$ 
```

Observation: If I use **rmdir** will encounter an error because it designed to remove empty directors and **dir11** contain the file in it so it is not empty to overcome that I use **rm -r dir11** ,this command will recursively remove the directory and all contents.

b) Then remove **dir12** using **rmdir –p** command. State what happened to the hierarchy (Note: you are in your home directory).



```
amgad@amgad-virtual-machine:~$ cd dir1
amgad@amgad-virtual-machine:~/dir1$ rm -r dir11
amgad@amgad-virtual-machine:~/dir1$ ls
dir12
amgad@amgad-virtual-machine:~/dir1$ rmdir -p dir12
amgad@amgad-virtual-machine:~/dir1$ cd..
cd..: command not found
amgad@amgad-virtual-machine:~/dir1$ cd ..
amgad@amgad-virtual-machine:~$ tree
.
├── Desktop
├── dir1
├── docs
│   └── mycv
├── Documents
├── Downloads
├── Music
├── Pictures
├── Public
├── snap
│   ├── snapd-desktop-integration [error opening dir]
│   └── tree
│       ├── 18
│       ├── common
│       └── current -> 18
├── Templates
└── Videos
```

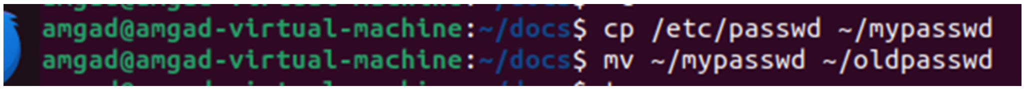
Because **dir12** is empty the directory is removed without any errors and this is the hierarchy after this command.

c) The output of the command **pwd** was **/home/user**. Write the absolute and relative path for the file **mycv**

> **Absolute path:** /home/amgad/docs/mycv
> **Relative path:** amgad/docs/mycv

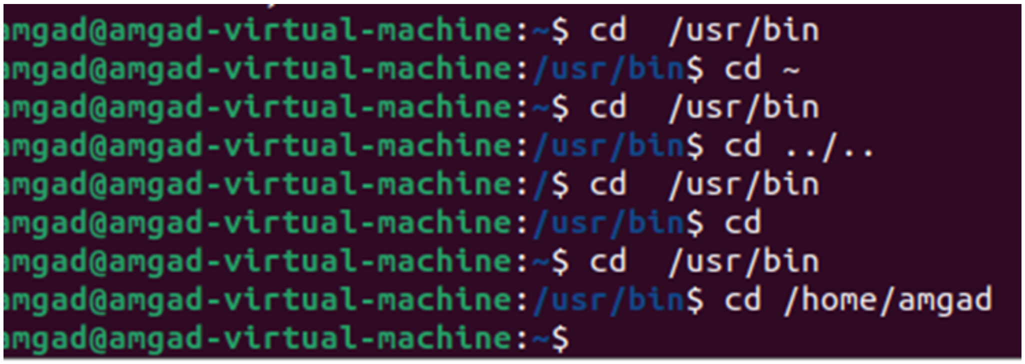
4. Copy the **/etc/passwd** file to your home directory making its name is **mypasswd**.
5. Rename this new file to be **oldpasswd**.

- **First command** will create a copy of the file in my home.
- **Second command** in the image will rename it.



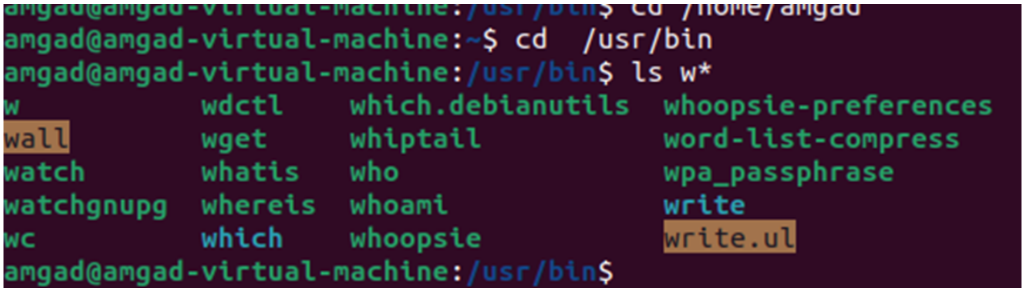
```
amgad@amgad-virtual-machine:~/docs$ cp /etc/passwd ~/mypasswd
amgad@amgad-virtual-machine:~/docs$ mv ~/mypasswd ~/oldpasswd
```

6. You are in **/usr/bin**, list four ways to go to your home directory.



```
mgad@amgad-virtual-machine:~$ cd  /usr/bin
mgad@amgad-virtual-machine:/usr/bin$ cd ~
mgad@amgad-virtual-machine:~$ cd  /usr/bin
mgad@amgad-virtual-machine:/usr/bin$ cd ../..
mgad@amgad-virtual-machine:/$ cd  /usr/bin
mgad@amgad-virtual-machine:/usr/bin$ cd
mgad@amgad-virtual-machine:~$ cd  /usr/bin
mgad@amgad-virtual-machine:/usr/bin$ cd /home/amgad
mgad@amgad-virtual-machine:~$
```

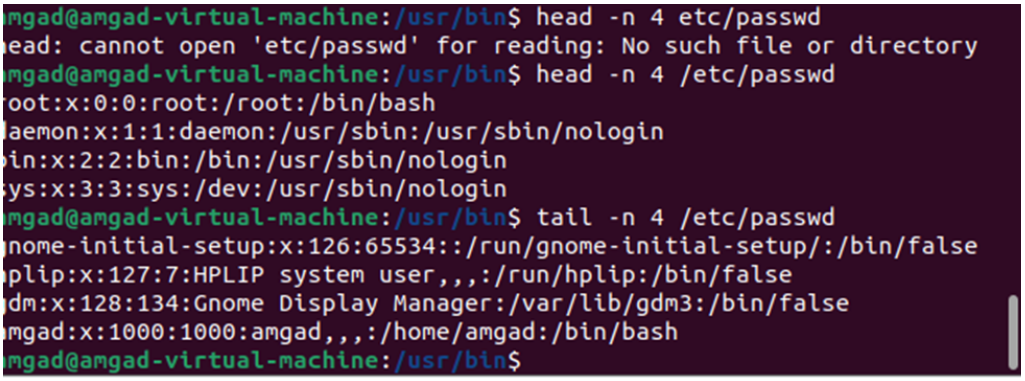
7. List Linux commands in **/usr/bin** that start with letter **w**.



```
amgad@amgad-virtual-machine:/usr/bin$ cd /home/amgad
amgad@amgad-virtual-machine:~$ cd  /usr/bin
amgad@amgad-virtual-machine:/usr/bin$ ls w*
w           wdctl    which.debianutils  whoopsie-preferences
wall        wget     whiptail           word-list-compress
watch       whatis   who                wpa_passphrase
watchgnupg  whereis  whoami             write
wc          which    whoopsie           write.ul
amgad@amgad-virtual-machine:/usr/bin$
```

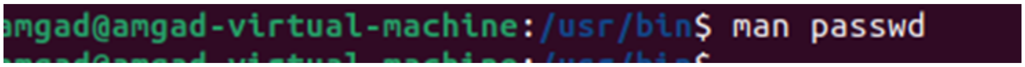
8. Display the first 4 lines of **/etc/passwd**.

9. Display the last 7 lines of **/etc/passwd**.

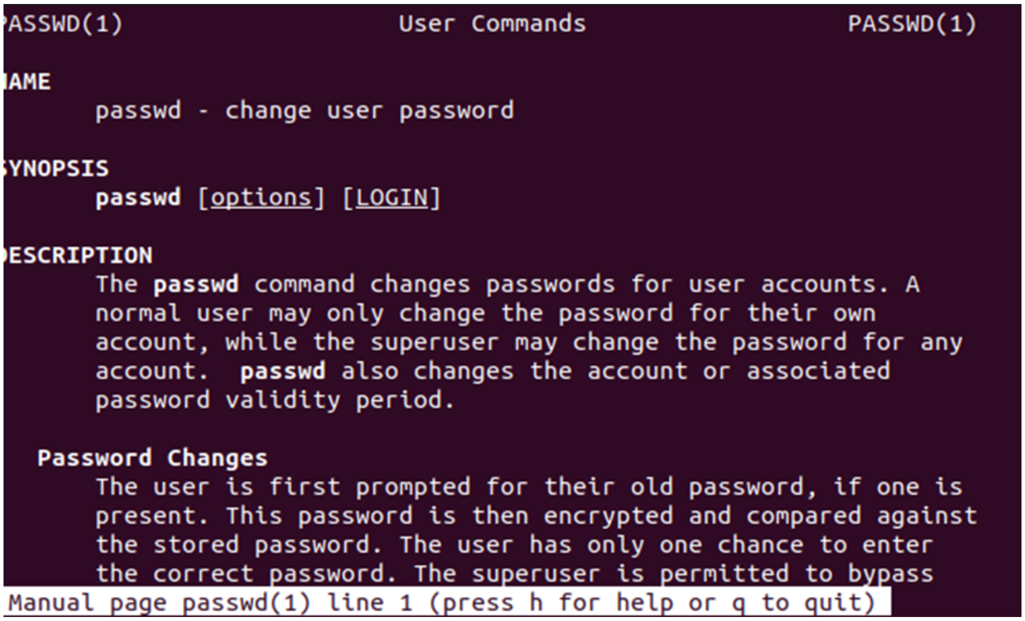


```
mgad@amgad-virtual-machine:/usr/bin$ head -n 4 etc/passwd
ead: cannot open 'etc/passwd' for reading: No such file or directory
mgad@amgad-virtual-machine:/usr/bin$ head -n 4 /etc/passwd
oot:x:0:0:root:/root:/bin/bash
aemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
in:x:2:2:bin:/bin:/usr/sbin/nologin
ys:x:3:3:sys:/dev:/usr/sbin/nologin
mgad@amgad-virtual-machine:/usr/bin$ tail -n 4 /etc/passwd
nome-initial-setup:x:126:65534::/run/gnome-initial-setup/:/bin/false
plip:x:127:7:HPLIP system user,,,:/run/hplip:/bin/false
dm:x:128:134:Gnome Display Manager:/var/lib/gdm3:/bin/false
mgad:x:1000:1000:amgad,,,:/home/amgad:/bin/bash
mgad@amgad-virtual-machine:/usr/bin$
```

10. Display the man pages of **passwd** the command and the file sequentially in one command.

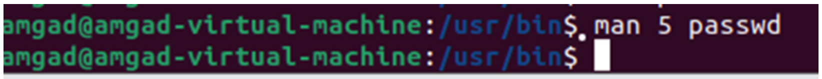


```
mgad@amgad-virtual-machine:/usr/bin$ man passwd
```

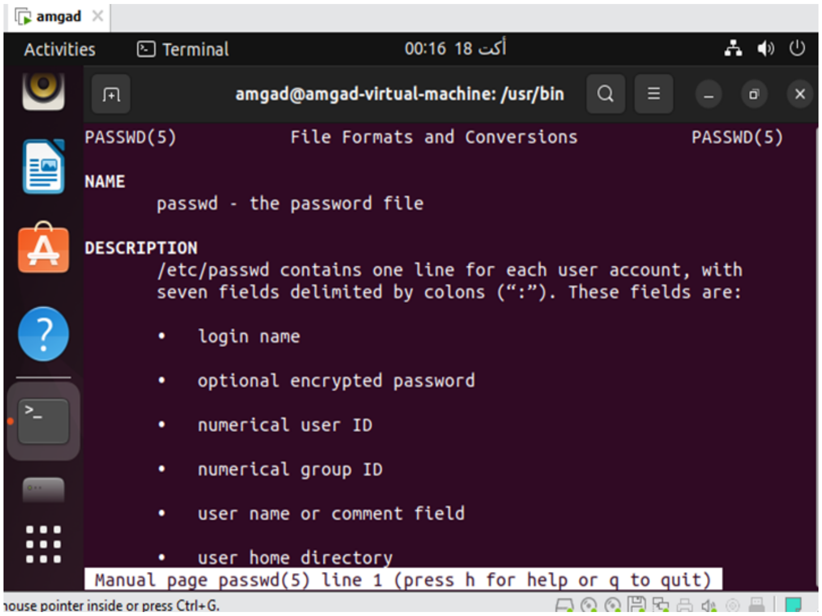


```
ASSWD(1)                    User Commands                   PASSWD(1)

AME
       passwd - change user password

YNOPSIS
       passwd [options] [LOGIN]

ESCRIPTION
       The passwd command changes passwords for user accounts. A
       normal user may only change the password for their own
       account, while the superuser may change the password for any
       account.  passwd also changes the account or associated
       password validity period.

  Password Changes
       The user is first prompted for their old password, if one is
       present. This password is then encrypted and compared against
       the stored password. The user has only one chance to enter
       the correct password. The superuser is permitted to bypass
Manual page passwd(1) line 1 (press h for help or q to quit)
```

11. Display the man page of the **passwd** file.



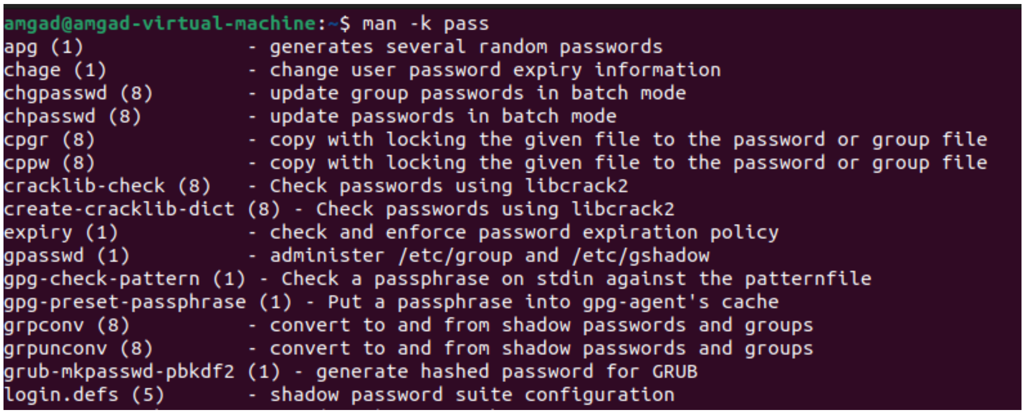
```
amgad@amgad-virtual-machine:/usr/bin$ man 5 passwd
amgad@amgad-virtual-machine:/usr/bin$ 
```



amgad
Activities
Terminal
00:16 18 أكت
amgad@amgad-virtual-machine: /usr/bin
ıouse pointer inside or press Ctrl+G.

```
PASSWD(5)              File Formats and Conversions              PASSWD(5)

NAME
       passwd - the password file

DESCRIPTION
       /etc/passwd contains one line for each user account, with
       seven fields delimited by colons (“:”). These fields are:

       •   login name

       •   optional encrypted password

       •   numerical user ID

       •   numerical group ID

       •   user name or comment field

       •   user home directory
 Manual page passwd(5) line 1 (press h for help or q to quit)
```

12. Display a list of all the commands that contain the keyword **passwd** in their man page.



```
amgad@amgad-virtual-machine:~$ man -k pass
apg (1)              - generates several random passwords
chage (1)            - change user password expiry information
chgpasswd (8)        - update group passwords in batch mode
chpasswd (8)         - update passwords in batch mode
cpgr (8)             - copy with locking the given file to the password or group file
cppw (8)             - copy with locking the given file to the password or group file
cracklib-check (8)   - Check passwords using libcrack2
create-cracklib-dict (8) - Check passwords using libcrack2
expiry (1)           - check and enforce password expiration policy
gpasswd (1)          - administer /etc/group and /etc/gshadow
gpg-check-pattern (1) - Check a passphrase on stdin against the patternfile
gpg-preset-passphrase (1) - Put a passphrase into gpg-agent's cache
grpconv (8)          - convert to and from shadow passwords and groups
grpunconv (8)        - convert to and from shadow passwords and groups
grub-mkpasswd-pbkdf2 (1) - generate hashed password for GRUB
login.defs (5)       - shadow password suite configuration
```