Lab 1

1. Install CentOS /RHEL

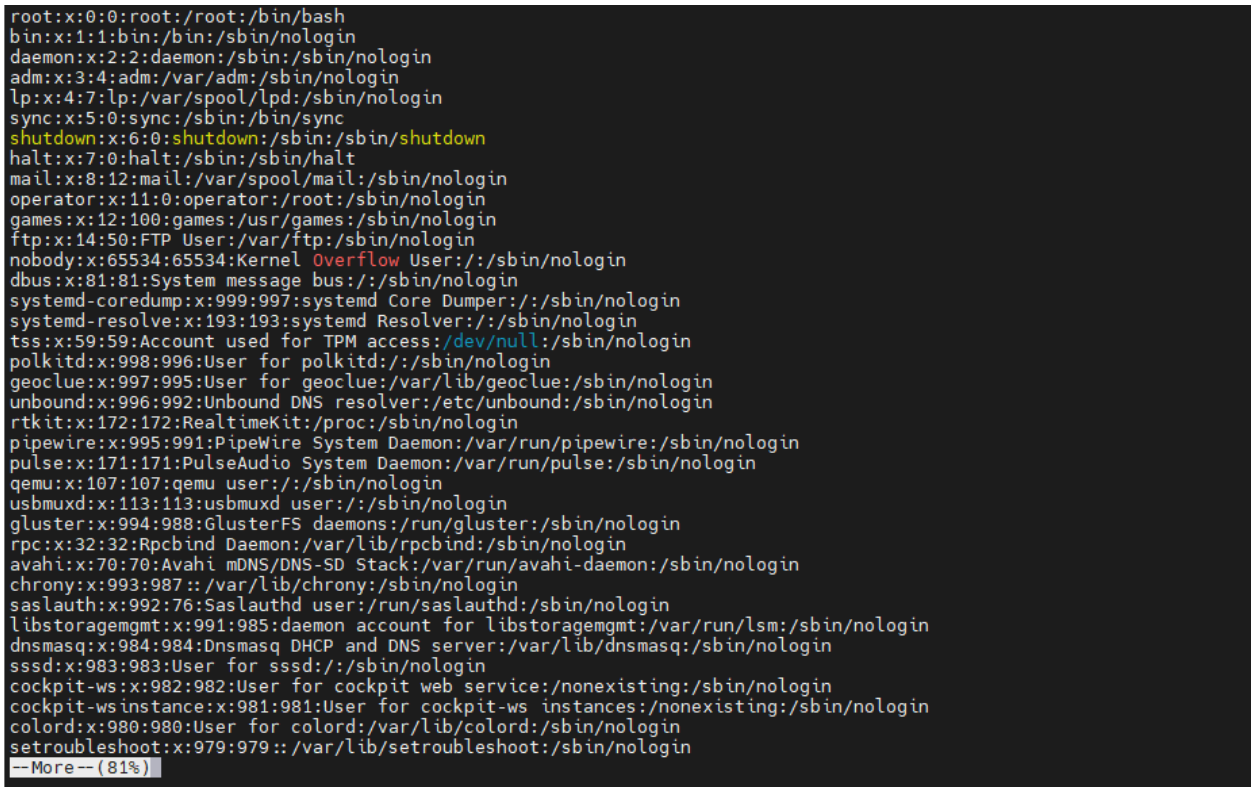
```
[amr@localhost ~]$ cat /etc/redhat-release
CentOS Stream release 8
[amr@localhost ~]$
```

2. What is the difference between cat and more command?

cat: command will display the entire content of the file on the terminal , also it can be used to concatenate multiple files into a single file.

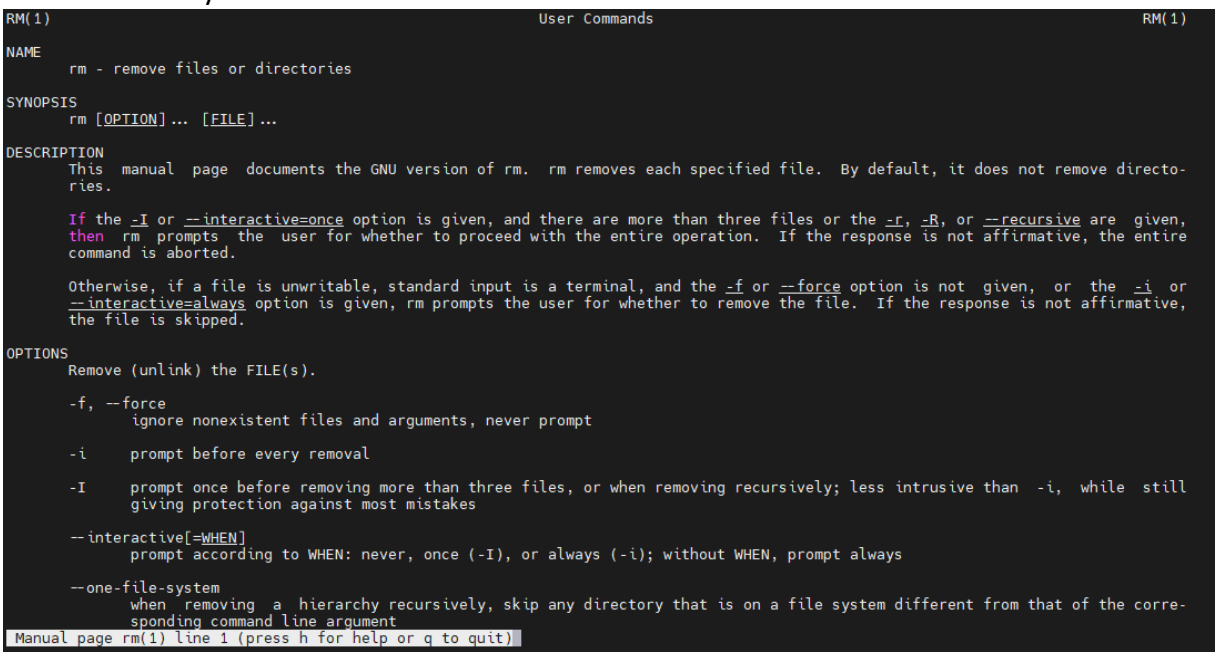
```
[amr@localhost sprints]$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:65534:65534:Kernel Overflow User:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
systemd-coredump:x:999:997:systemd Core Dumper:/:/sbin/nologin
systemd-resolve:x:193:193:systemd Resolver:/:/sbin/nologin
tss:x:59:59:Account used for TPM access:/dev/null:/sbin/nologin
polkitd:x:998:996:User for polkitd:/:/sbin/nologin
geoclue:x:997:995:User for geoclue:/var/lib/geoclue:/sbin/nologin
unbound:x:996:992:Unbound DNS resolver:/etc/unbound:/sbin/nologin
rtkit:x:172:172:RealtimeKit:/proc:/sbin/nologin
pipewire:x:995:991:PipeWire System Daemon:/var/run/pipewire:/sbin/nologin
pulse:x:171:171:PulseAudio System Daemon:/var/run/pulse:/sbin/nologin
qemu:x:107:107:qemu user:/:/sbin/nologin
usbmuxd:x:113:113:usbmuxd user:/:/sbin/nologin
gluster:x:994:988:GlusterFS daemons:/run/gluster:/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin
chrony:x:993:987::/var/lib/chrony:/sbin/nologin
saslauth:x:992:76:Saslauthd user:/run/saslauthd:/sbin/nologin
libstoragemgmt:x:991:985:daemon account for libstoragemgmt:/var/run/lsm:/sbin/nologin
dnsmasq:x:984:984:Dnsmasq DHCP and DNS server:/var/lib/dnsmasq:/sbin/nologin
sssd:x:983:983:User for sssd:/:/sbin/nologin
cockpit-ws:x:982:982:User for cockpit web service:/nonexisting:/sbin/nologin
cockpit-wsinstance:x:981:981:User for cockpit-ws instances:/nonexisting:/sbin/nologin
colord:x:980:980:User for colord:/var/lib/colord:/sbin/nologin
setroubleshoot:x:979:979::/var/lib/setroubleshoot:/sbin/nologin
```

more: command will display content that would fit your screen and you can press enter to see rest of the content line by line and space key can be used to scroll another page.



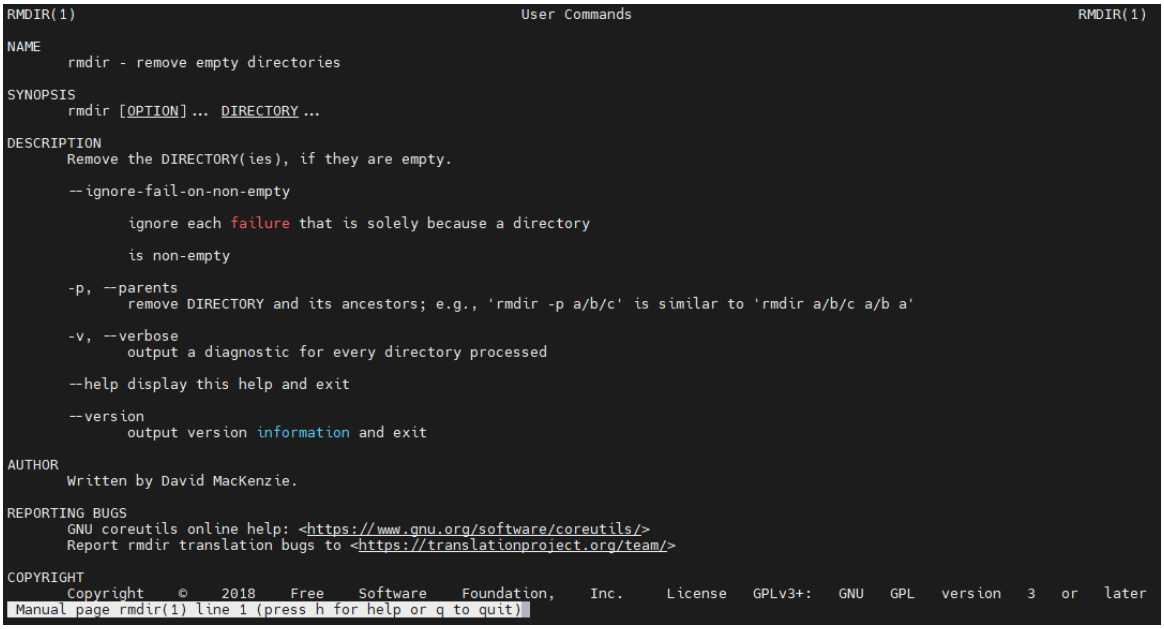
3. What is the difference between rm and rmdir using man?

rm: Can remove both files and non-empty directories and we can use “-r” to delete the directory and its content.



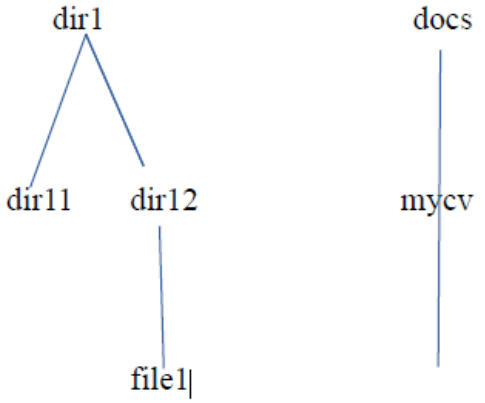
```
RM(1)                                    User Commands                                    RM(1)

NAME
       rm - remove files or directories

SYNOPSIS
       rm [OPTION]... [FILE]...

DESCRIPTION
       This  manual  page  documents the GNU version of rm.  rm removes each specified file.  By default, it does not remove directo-
       ries.

       If the -I or --interactive=once option is given, and there are more than three files or the -r, -R, or --recursive are  given,
       then  rm  prompts  the  user for whether to proceed with the entire operation.  If the response is not affirmative, the entire
       command is aborted.

       Otherwise, if a file is unwritable, standard input is a terminal, and the -f or --force option is not  given,  or  the  -i  or
       --interactive=always option is given, rm prompts the user for whether to remove the file.  If the response is not affirmative,
       the file is skipped.

OPTIONS
       Remove (unlink) the FILE(s).

       -f, --force
              ignore nonexistent files and arguments, never prompt

       -i     prompt before every removal

       -I     prompt once before removing more than three files, or when removing recursively; less intrusive than  -i,  while  still
              giving protection against most mistakes

       --interactive[=WHEN]
              prompt according to WHEN: never, once (-I), or always (-i); without WHEN, prompt always

       --one-file-system
              when  removing  a  hierarchy recursively, skip any directory that is on a file system different from that of the corre-
              sponding command line argument
 Manual page rm(1) line 1 (press h for help or q to quit)
```

rmdir: Can only be used to remove an empty directory and it won’t work for non-empty directories.



```
RMDIR(1)                                 User Commands                                 RMDIR(1)

NAME
       rmdir - remove empty directories

SYNOPSIS
       rmdir [OPTION]... DIRECTORY...

DESCRIPTION
       Remove the DIRECTORY(ies), if they are empty.

       --ignore-fail-on-non-empty

              ignore each failure that is solely because a directory

              is non-empty

       -p, --parents
              remove DIRECTORY and its ancestors; e.g., 'rmdir -p a/b/c' is similar to 'rmdir a/b/c a/b a'

       -v, --verbose
              output a diagnostic for every directory processed

       --help display this help and exit

       --version
              output version information and exit

AUTHOR
       Written by David MacKenzie.

REPORTING BUGS
       GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
       Report rmdir translation bugs to <https://translationproject.org/team/>

COPYRIGHT
       Copyright  ©  2018  Free  Software  Foundation,  Inc.  License  GPLv3+:  GNU  GPL  version  3  or  later
 Manual page rmdir(1) line 1 (press h for help or q to quit)
```

4. Create the following hierarchy under your home directory




```
[amr@localhost ~]$ mkdir dir1 docs
[amr@localhost ~]$ cd dir1
[amr@localhost dir1]$ mkdir dir11 dir12
[amr@localhost dir1]$ cd dir12/
[amr@localhost dir12]$ touch file1
[amr@localhost dir12]$ cd ..
[amr@localhost dir1]$ cd ..
[amr@localhost ~]$ cd docs/
[amr@localhost docs]$ touch mycv
[amr@localhost docs]$ ls
mycv
```

```
[amr@localhost ~]$ ls dir1/
dir11  dir12
[amr@localhost ~]$ ls dir1/dir12
file1
[amr@localhost ~]$
```

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

```
[amr@localhost ~]$ rmdir dir1/dir11
[amr@localhost ~]$ ls dir1
dir12
[amr@localhost ~]$
```

We can delete it easily as it's an empty directory.

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

```
[amr@localhost ~]$ rmdir -p dir1/dir12/
rmdir: failed to remove 'dir1/dir12/': Directory not empty
[amr@localhost ~]$
```

We can't delete the directory as it's not empty.

```
[amr@localhost ~]$ rm -r dir1/dir12/
[amr@localhost ~]$ ls dir1/
[amr@localhost ~]$
```

But if we used " rm –r " we can delete it easily.

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

The absolute path is exactly where to locate the file: /home/amr/docs/mycv

```
[amr@localhost ~]$ realpath docs
/home/amr/docs
[amr@localhost ~]$
```

The relative path is the location of the file relative to my current location : /docs/mycv

```
[amr@localhost ~]$ ls docs/
mycv
```

5. Copy the /etc/passwd file to your home directory making its name is mypasswd.

```
[amr@localhost ~]$ cp /etc/passwd mypasswd
[amr@localhost ~]$
```

6. Rename this new file to be oldpasswd.

```
[amr@localhost ~]$ mv mypasswd oldpasswd
[amr@localhost ~]$ ls
Desktop  dir1  docs  Documents  Downloads  Music  oldpasswd  Pictures  Public  sprints  Templates  Videos
[amr@localhost ~]$
```

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

- cd
- cd ~
- cd $HOME
- cd /home/amr

```
[amr@localhost ~]$ cd /usr/bin
[amr@localhost bin]$ cd
[amr@localhost ~]$ cd /usr/bin
[amr@localhost bin]$ cd ~
[amr@localhost ~]$ cd /usr/bin
[amr@localhost bin]$ cd $HOME
[amr@localhost ~]$ cd /usr/bin
[amr@localhost bin]$ cd /home/amr/
[amr@localhost ~]$ cd /usr/bin
[amr@localhost bin]$ cd ../../home/amr/
[amr@localhost ~]$
```

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

```
[amr@localhost bin]$ ls w*
w     wall    watchgnupg  wc     wget    whereis  whiptail  whoami              word-list-compress  wvgain  wvunpack
wait  watch   wavpack     wdctl  whatis  which    who       wnck-urgency-monitor  write               wvtag
[amr@localhost bin]$
```

9. Display the first 4 lines of /etc/passwd

```
[amr@localhost bin]$ head -n 4 /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
```

10. Display the last 7 lines of /etc/passwd

```
[amr@localhost bin]$ tail -n 7 /etc/passwd
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
clevis:x:977:977:Clevis Decryption Framework unprivileged user:/var/cache/clevis:/sbin/nologin
gdm:x:42:42::/var/lib/gdm:/sbin/nologin
gnome-initial-setup:x:976:975::/run/gnome-initial-setup/:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
tcpdump:x:72:72::/:/sbin/nologin
amr:x:1000:1000:amr:/home/amr:/bin/bash
```

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

```
[amr@localhost ~]$ man passwd && man 5 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"  ser-
       vice 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  per-
              formed  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  pub-
              lic key authentication. 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  pass-
       words,  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 him- or herself 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'.)
 Manual page passwd(5) line 1 (press h for help or q to quit)
```

12.Display the man page of the passwd file.

```
[amr@localhost ~]$ man 5 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  pass-
       words,  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 him- or herself 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'.)

 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.

```
[amr@localhost ~]$ man -k passwd
chgpasswd (8)         - update group passwords in batch mode
chpasswd (8)          - update passwords in batch mode
fgetpwent_r (3)       - get passwd file entry reentrantly
getpwent_r (3)        - get passwd file entry reentrantly
gpasswd (1)           - administer /etc/group and /etc/gshadow
grub2-mkpasswd-pbkdf2 (1) - Generate a PBKDF2 password hash.
lpasswd (1)           - Change group or user password
openssl-passwd (1ssl) - compute password hashes
pam_localuser (8)     - require users to be listed in /etc/passwd
passwd (1)            - update user's authentication tokens
passwd (5)            - password file
passwd2des (3)        - RFS password encryption
pwhistory_helper (8)  - Helper binary that transfers password hashes from passwd or shadow to opasswd
saslpasswd2 (8)       - set a user's sasl password
smbpasswd (5)         - The Samba encrypted password file
sslpasswd (1ssl)      - compute password hashes
vncpasswd (1)         - change the VNC password
[amr@localhost ~]$
```