

```
1- The 'cat' command displays the entire content in a single page at once.
2- the 'more' command displays it on multiple pages allowing the user to scroll one page at a time.
thus for smaller files 'cat' is useful where in the longer ones the 'more' is more useful.

this is a use case to show the difference.

ubuntu@ubuntu-course: $
```

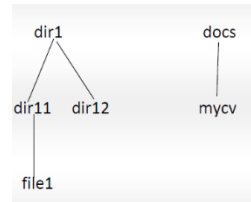
3. What is the difference between rm and rmdir using man?

rm: rm removes each specified file. By default, it does not remove directories.

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

4. Create the following hierarchy under your home directory:

```
ubuntu@ubuntu-course: ~  
ubuntu@ubuntu-course:~$ mkdir -p dir1/dir11 dir1/dir12 docs/  
ubuntu@ubuntu-course:~$ touch dir1/dir11/file1 docs/mycv  
ubuntu@ubuntu-course:~$ ls -R  
.:  
dir1 docs lab1  
  
./dir1:  
dir11 dir12  
  
./dir1/dir11:  
file1  
  
./dir1/dir12:  
  
./docs:  
mycv  
ubuntu@ubuntu-course:~$
```



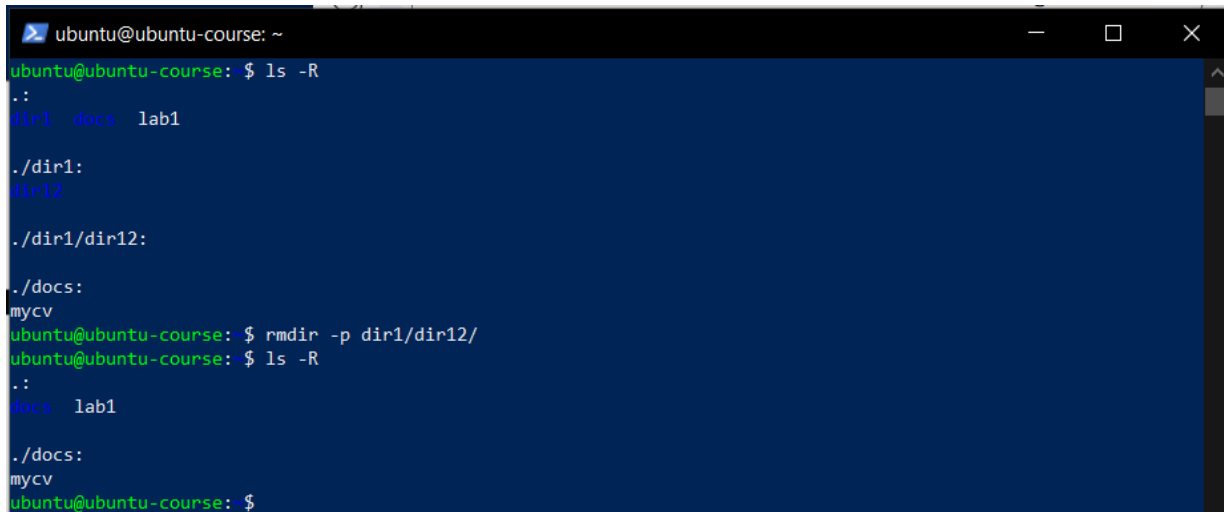
a. Remove dir11 in one-step. What did you notice? And how did you overcome that?

I cannot remove the dir11 using **rmdir** because the directory is not empty, so I go with **rm -rf** to remove the directory.

```
ubuntu@ubuntu-course: ~  
ubuntu@ubuntu-course:~$ rmdir dir1/dir11/  
rmdir: failed to remove 'dir1/dir11/': Directory not empty  
ubuntu@ubuntu-course:~$ rm -rf dir1/dir11/  
ubuntu@ubuntu-course:~$ ls -R  
.:  
dir1 docs lab1  
  
./dir1:  
dir12  
  
./dir1/dir12:  
  
./docs:  
mycv  
ubuntu@ubuntu-course:~$
```

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

The whole directory is removed due to the behavior of the command, where the flag -p removes parent directories only if they become empty after deletion. Since we have already deleted dir11, dir12 is the only dir for the parent dir1, so it will also delete it, leaving us with no dir1 at all. So we can conclude that the rmdir -p remove multiple levels.

A terminal window titled 'ubuntu@ubuntu-course: ~' showing a series of commands and their outputs. The user runs 'ls -R' which lists the directory structure: '.', 'dir1', 'docs', 'lab1', and their subdirectories. Then, the user runs 'rmdir -p dir1/dir12/'. Finally, the user runs 'ls -R' again, showing that 'dir1' and 'dir12' have been removed, leaving only 'docs' and 'lab1' under the root directory.

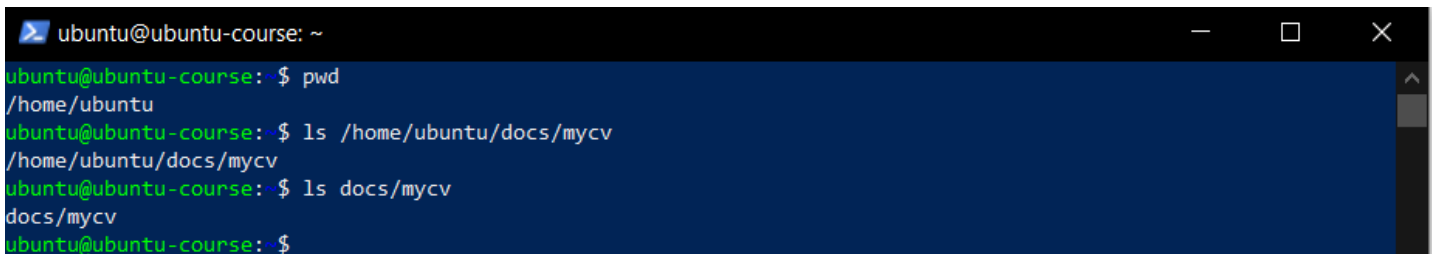
```
ubuntu@ubuntu-course: ~  
ubuntu@ubuntu-course:~$ ls -R  
.:  
dir1 docs lab1  
  
./dir1:  
dir12  
  
./dir1/dir12:  
  
./docs:  
mycv  
ubuntu@ubuntu-course:~$ rmdir -p dir1/dir12/  
ubuntu@ubuntu-course:~$ ls -R  
.:  
docs lab1  
  
./docs:  
mycv  
ubuntu@ubuntu-course:~$
```

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

/home/user/docs/mycv → difference on my machine due to different username (user → ubuntu)

Absolute: /home/user/docs/mycv

Relative: docs/mycv

A terminal window titled 'ubuntu@ubuntu-course: ~' showing the user running 'pwd' and 'ls' commands to verify the file path. The 'pwd' command returns '/home/ubuntu'. The 'ls /home/ubuntu/docs/mycv' command shows the file exists. The 'ls docs/mycv' command also shows the file exists.

```
ubuntu@ubuntu-course: ~  
ubuntu@ubuntu-course:~$ pwd  
/home/ubuntu  
ubuntu@ubuntu-course:~$ ls /home/ubuntu/docs/mycv  
/home/ubuntu/docs/mycv  
ubuntu@ubuntu-course:~$ ls docs/mycv  
docs/mycv  
ubuntu@ubuntu-course:~$
```

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

```
ubuntu@ubuntu-course: ~  
ubuntu@ubuntu-course:~$ cp /etc/passwd ~/mypasswd  
ubuntu@ubuntu-course:~$ ls  
docs  lab1  mypasswd  
ubuntu@ubuntu-course:~$
```

6. Rename this new file to be `oldpasswd`.

```
ubuntu@ubuntu-course: ~  
ubuntu@ubuntu-course:~$ mv mypasswd oldpasswd  
ubuntu@ubuntu-course:~$ ls  
docs  lab1  oldpasswd  
ubuntu@ubuntu-course:~$
```

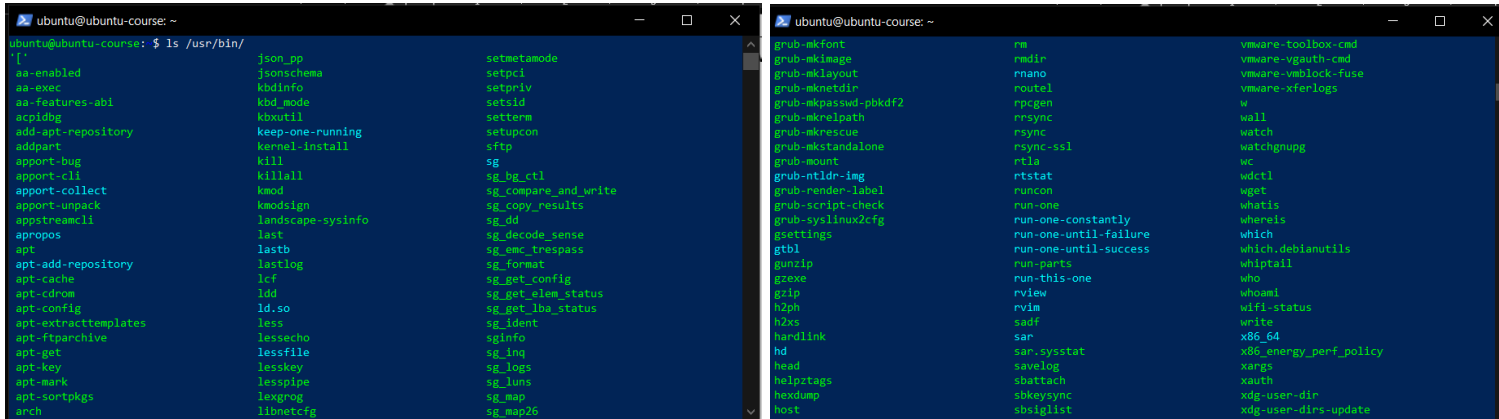
7. You are in `/usr/bin`, list four ways to go to your home directory.

- `cd`
- `cd ~`
- `cd /home/ubuntu` → absolute path
- `cd ../../home/ubuntu/` → moves two directories up from `/usr/bin` (to `/`), and then from `/`, it navigates to the absolute path `/home/ubuntu/`.

```
Select ubuntu@ubuntu-course: ~  
ubuntu@ubuntu-course:/usr/bin$ cd  
ubuntu@ubuntu-course:~$ cd /usr/bin/  
ubuntu@ubuntu-course:/usr/bin$ cd ~  
ubuntu@ubuntu-course:~$ cd /usr/bin/  
ubuntu@ubuntu-course:/usr/bin$ cd /home/ubuntu/  
ubuntu@ubuntu-course:~$ cd /usr/bin/  
ubuntu@ubuntu-course:/usr/bin$ cd ../../home/ubuntu/  
ubuntu@ubuntu-course:~$
```

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

Using `ls /usr/bin/` to list all the commands/files and then navigate to letter w because the files are displayed in alphabetical order.



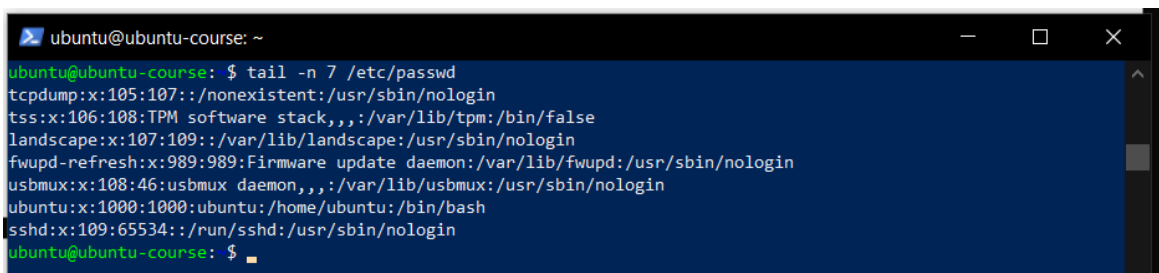
```
ubuntu@ubuntu-course: ~  
ubuntu@ubuntu-course: $ ls /usr/bin/  
[  
aa-enabled  
aa-exec  
aa-features-abi  
acpidbg  
add-apt-repository  
addpart  
apport-bug  
apport-cgi  
apport-collect  
apport-unpack  
appstreamcli  
apropos  
apt  
apt-add-repository  
apt-cache  
apt-cdrom  
apt-config  
apt-extracttemplates  
apt-ftparchive  
apt-get  
apt-key  
apt-mark  
apt-sortpkgs  
arch  
json_pp  
jsonschema  
kbdinfo  
kbd_mode  
kbxutil  
keep-one-running  
kernel-install  
kill  
killall  
kmod  
kmodsign  
landscape-sysinfo  
last  
lastb  
lastlog  
lcf  
ldd  
ld.so  
less  
lessecho  
lessfile  
lesskey  
lesspipe  
lexgrog  
libnetcfg  
setmetamode  
setpci  
setpriv  
setuid  
setterm  
setupcon  
sftp  
sg  
sg_bg_ctl  
sg_compare_and_write  
sg_copy_results  
sg_dd  
sg_decode_sense  
sg_emc_trespass  
sg_format  
sg_get_config  
sg_get_elem_status  
sg_get_lba_status  
sg_ident  
sginfo  
sg_inq  
sg_logs  
sg_luns  
sg_map  
sg_map36  
nw  
redir  
rnano  
routel  
rpcgen  
rrsync  
rsync  
rsync-ssl  
rtia  
rtstat  
runcon  
run-one  
run-one-constantly  
run-one-until-failure  
run-one-until-success  
run-parts  
run-this-one  
rview  
rvim  
sadm  
sarf  
sar.sysstat  
savehog  
sbattach  
sbkeysync  
sblglist  
vmware-toolbox-cmd  
vmware-vgauth-cmd  
vmware-vmblock-fuse  
vmware-xferlogs  
w  
wall  
watch  
watchgnupg  
wc  
wdctl  
wget  
whatis  
whereis  
which  
which.debianutils  
whiptail  
who  
whoami  
wifi-status  
write  
x86_64  
x86_energy_perf_policy  
xargs  
xauth  
xdg-user-dir  
xdg-user-dirs-update
```

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



```
ubuntu@ubuntu-course: ~  
ubuntu@ubuntu-course: $ head -n 4 /etc/passwd  
root:x:0:0:root:/root:/bin/bash  
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin  
bin:x:2:2:bin:/bin:/usr/sbin/nologin  
sys:x:3:3:sys:/dev:/usr/sbin/nologin  
ubuntu@ubuntu-course: $
```

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



```
ubuntu@ubuntu-course: ~  
ubuntu@ubuntu-course: $ tail -n 7 /etc/passwd  
tcpdump:x:105:107::/nonexistent:/usr/sbin/nologin  
tss:x:106:108:TPM software stack,,,:/var/lib/tpm:/bin/false  
landscape:x:107:109::/var/lib/landscape:/usr/sbin/nologin  
fwupd-refresh:x:989:989:Firmware update daemon:/var/lib/fwupd:/usr/sbin/nologin  
usbmux:x:108:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin  
ubuntu:x:1000:1000:ubuntu:/home/ubuntu:/bin/bash  
sshd:x:109:65534::/run/sshd:/usr/sbin/nologin  
ubuntu@ubuntu-course: $
```

11. Display the man page of the passwd file.

```
ubuntu@ubuntu-course: ~  
ubuntu@ubuntu-course: $ man passwd  
ubuntu@ubuntu-course: $
```

```
ubuntu@ubuntu-course: ~  
PASSWD(1) User Commands PASSWD(1)  
  
NAME  
passwd - change user password  
  
SYNOPSIS  
passwd [options] [LOGIN]  
  
DESCRIPTION  
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 this step so that forgotten passwords may be changed.  
  
After the password has been entered, password aging information is checked to see if the user is  
permitted to change the password at this time. If not, passwd refuses to change the password and  
exits.  
  
The user is then prompted twice for a replacement password. The second entry is compared against the  
first and both are required to match in order for the password to be changed.  
  
Then, the password is tested for complexity. As a general guideline, passwords should consist of 6 to  
8 characters including one or more characters from each of the following sets:  


- lower case alphabets
- digits 0 thru 9
- punctuation marks

  
Care must be taken not to include the system default erase or kill characters. passwd will reject any  
password which is not suitably complex.  
  
Hints for user passwords  
The security of a password depends upon the strength of the encryption algorithm and the size of the  
key space. The legacy UNIX System encryption method is based on the NBS DES algorithm. More recent  
methods are now recommended (see ENCRYPT_METHOD). The size of the key space depends upon the  
randomness of the password which is selected.  
  
Compromises in password security normally result from careless password selection or handling. For  
this reason, you should not select a password which appears in a dictionary or which must be written  
down. The password should also not be a proper name, your license number, birth date, or street  
address. Any of these may be used as guesses to violate system security.  
  
You can find advice on how to choose a strong password on  
http://en.wikipedia.org/wiki/Password\_strength  
  
OPTIONS  
The options which apply to the passwd command are:  
Manual page passwd(1) line 1/171 35% (press h for help or q to quit)
```

Activate Windows
Go to Settings to activate Windows.