

CSCI-2201

Lab-1

Anas Alhadi

B00895875

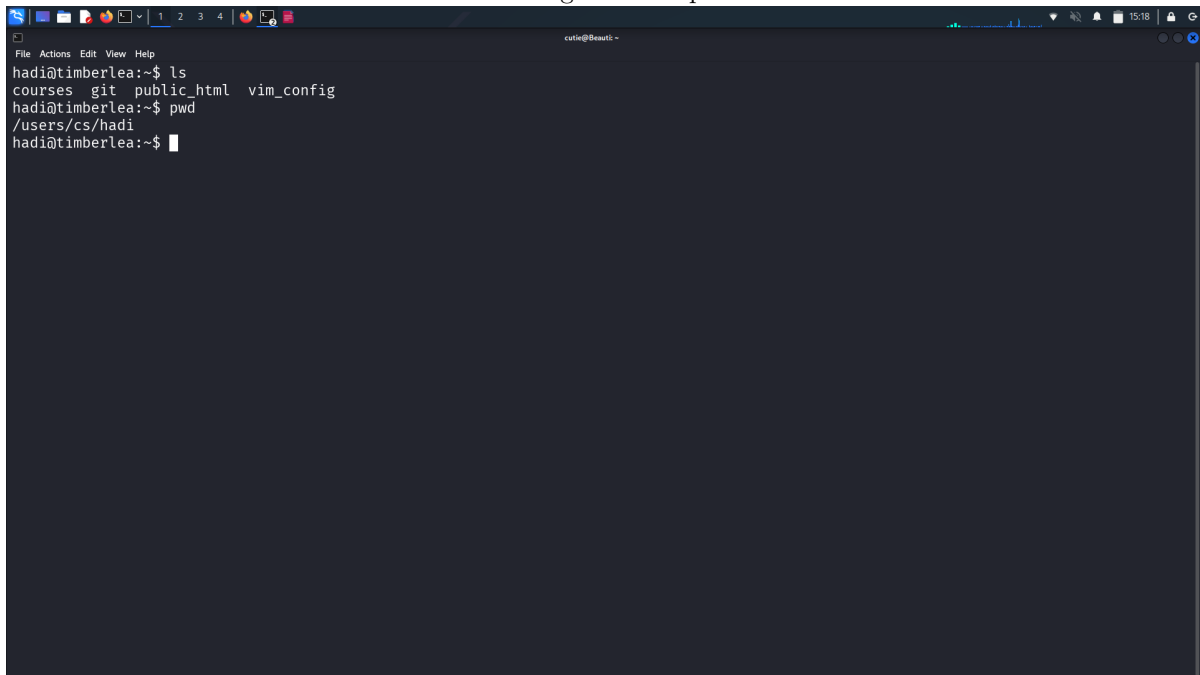
January 24, 2025

Contents

1	Exercise 1.A	2
2	Exercise 1.B	5
3	Exercise 2.A	6
4	Exercise 2.B	8
5	Exercise 3.A	9
6	Exercise 3.B	11

1 Exercise 1.A

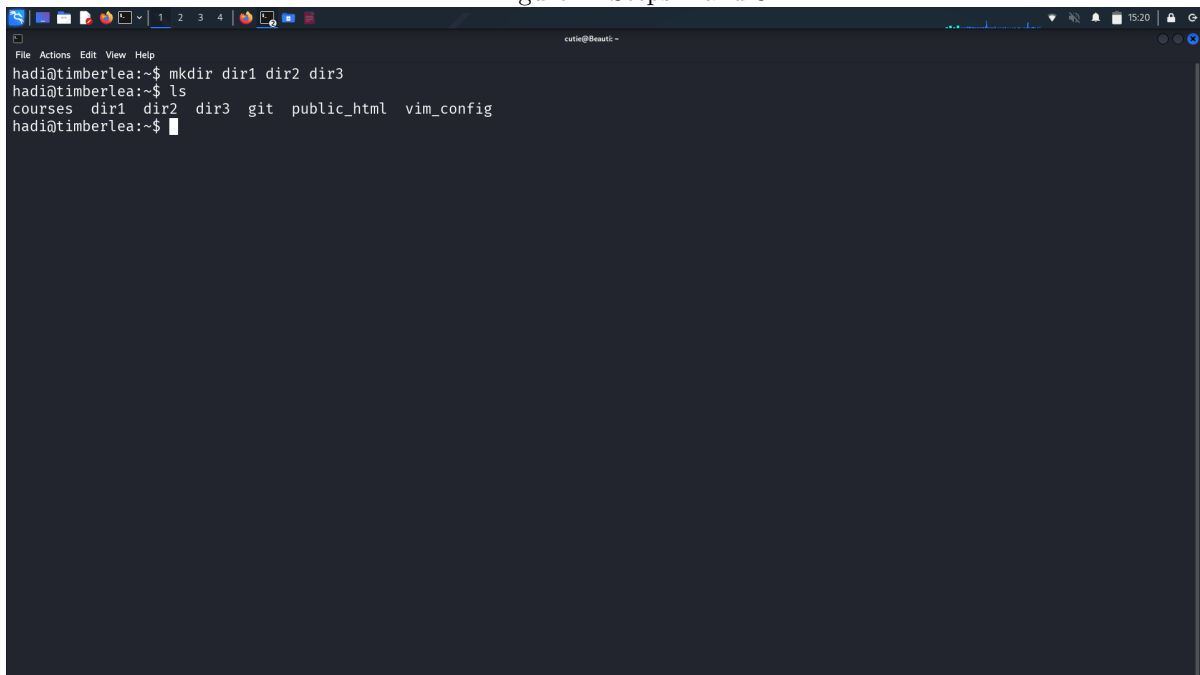
Figure 1: Step 1



A terminal window titled 'cutie@Beauti' with a menu bar (File, Actions, Edit, View, Help) and a toolbar. The terminal shows the following commands and output:

```
hadi@timberlea:~$ ls
courses git public_html vim_config
hadi@timberlea:~$ pwd
/users/cs/hadi
hadi@timberlea:~$
```

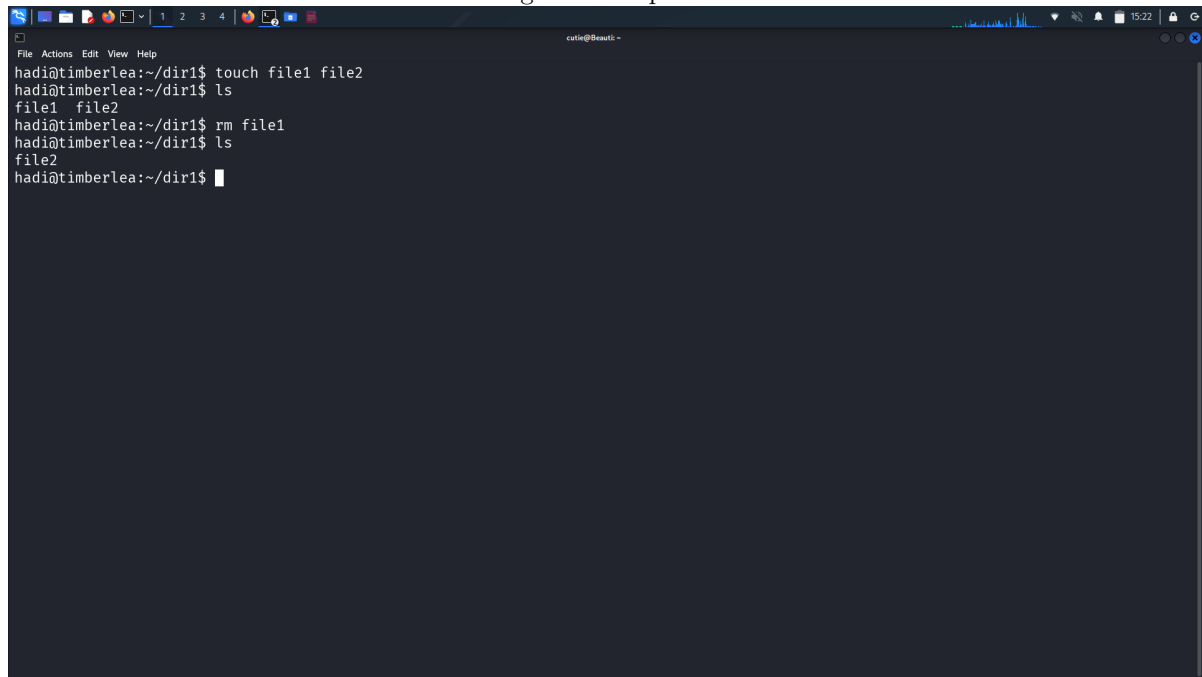
Figure 2: Steps 2 and 3



A terminal window titled 'cutie@Beauti' with a menu bar (File, Actions, Edit, View, Help) and a toolbar. The terminal shows the following commands and output:


```
hadi@timberlea:~$ mkdir dir1 dir2 dir3
hadi@timberlea:~$ ls
courses dir1 dir2 dir3 git public_html vim_config
hadi@timberlea:~$
```

Figure 3: Steps 4 and 5



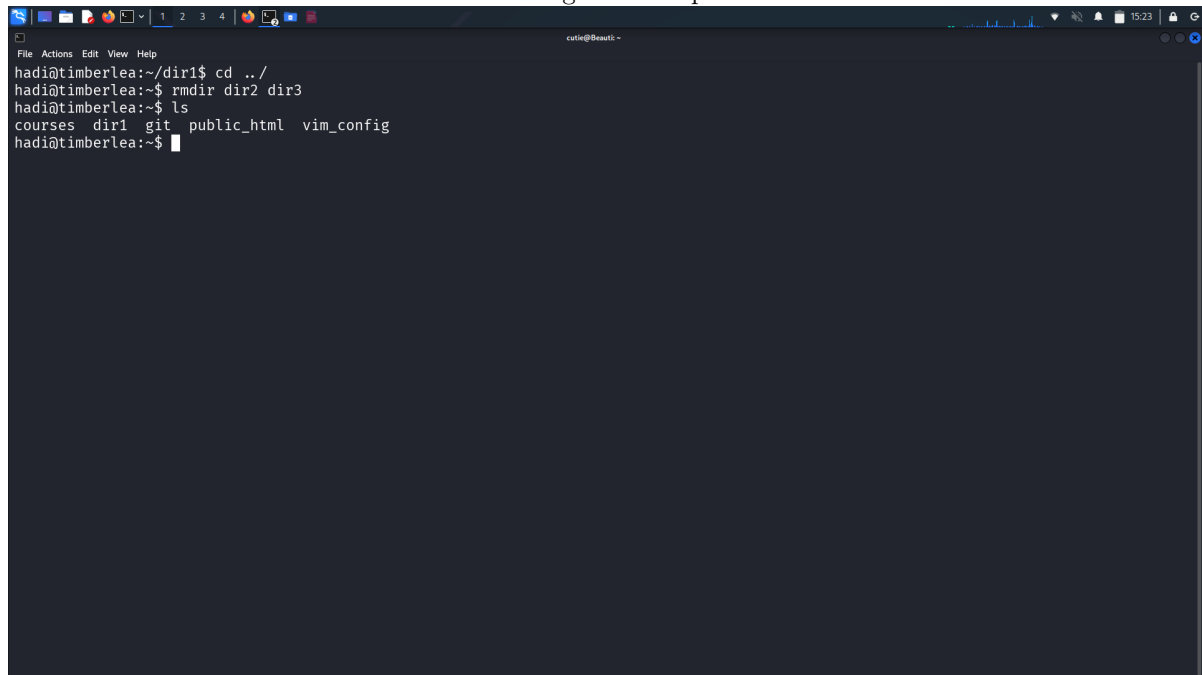
```
File Actions Edit View Help
hadi@timberlea:~/dir1$ touch file1 file2
hadi@timberlea:~/dir1$ ls
file1  file2
hadi@timberlea:~/dir1$ rm file1
hadi@timberlea:~/dir1$ ls
file2
hadi@timberlea:~/dir1$
```

Figure 4: Step 6



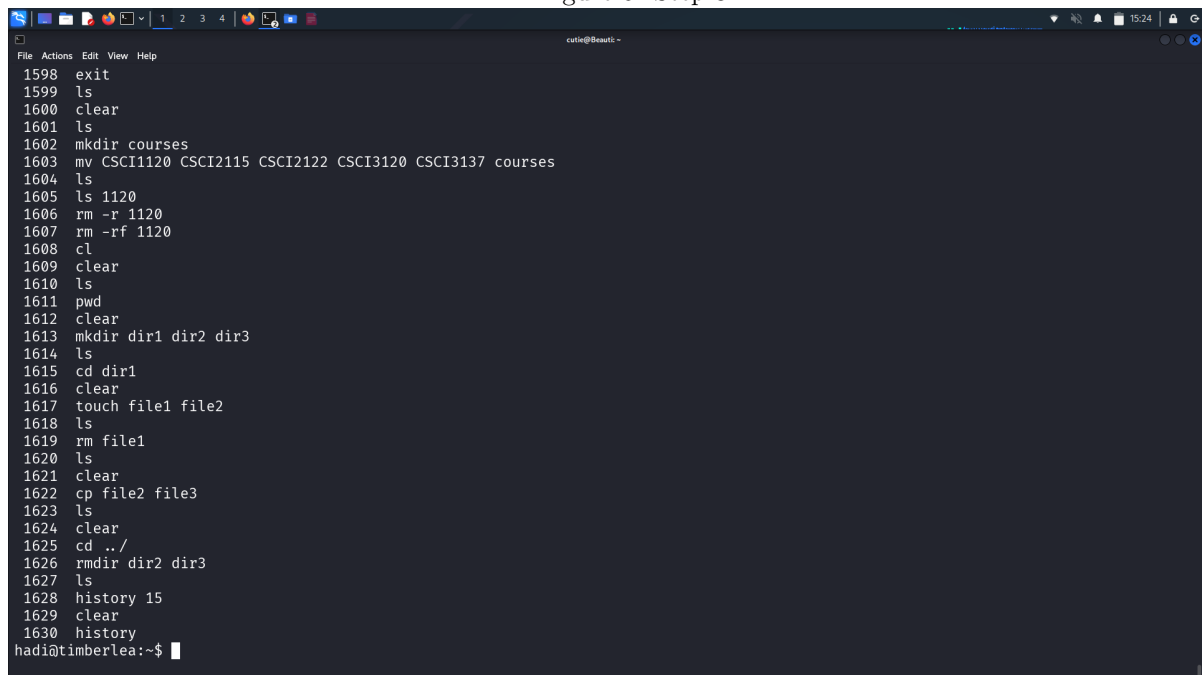
```
File Actions Edit View Help
hadi@timberlea:~/dir1$ cp file2 file3
hadi@timberlea:~/dir1$ ls
file2  file3
hadi@timberlea:~/dir1$
```

Figure 5: Step 7



```
File Actions Edit View Help
hadi@timberlea:~/dir1$ cd ../
hadi@timberlea:~$ rmdir dir2 dir3
hadi@timberlea:~$ ls
courses dir1 git public_html vim_config
hadi@timberlea:~$
```

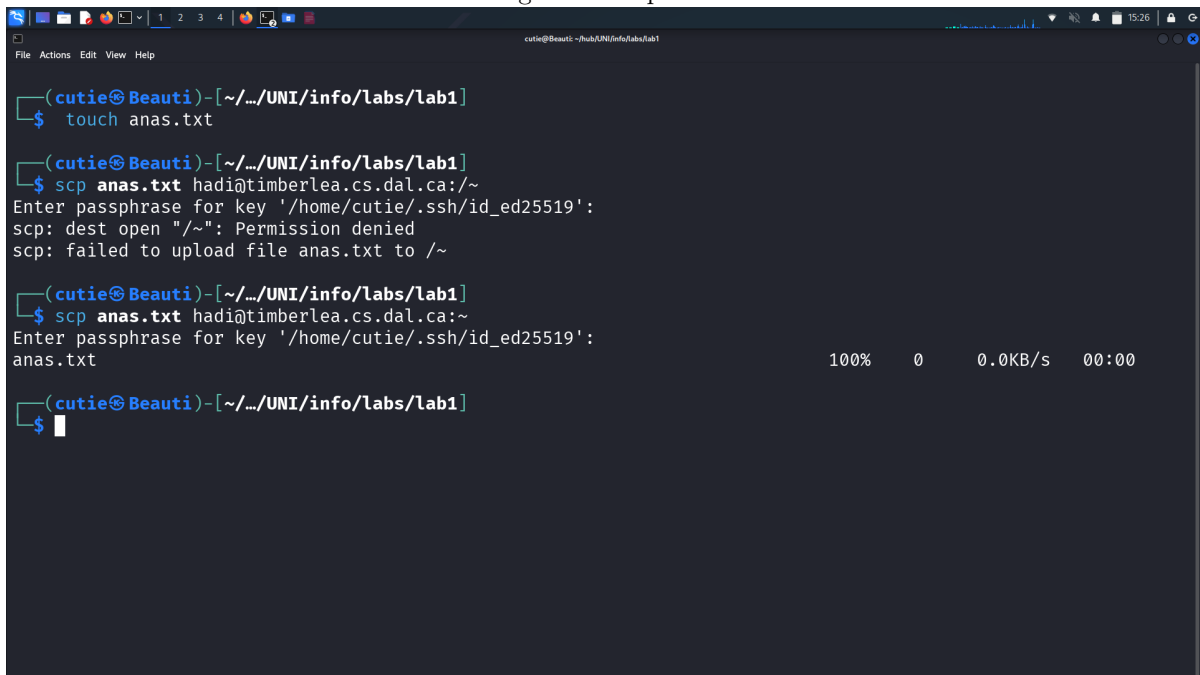
Figure 6: Step 8



```
File Actions Edit View Help
1598 exit
1599 ls
1600 clear
1601 ls
1602 mkdir courses
1603 mv CSCI1120 CSCI2115 CSCI2122 CSCI3120 CSCI3137 courses
1604 ls
1605 ls 1120
1606 rm -r 1120
1607 rm -rf 1120
1608 cl
1609 clear
1610 ls
1611 pwd
1612 clear
1613 mkdir dir1 dir2 dir3
1614 ls
1615 cd dir1
1616 clear
1617 touch file1 file2
1618 ls
1619 rm file1
1620 ls
1621 clear
1622 cp file2 file3
1623 ls
1624 clear
1625 cd ../
1626 rmdir dir2 dir3
1627 ls
1628 history 15
1629 clear
1630 history
hadi@timberlea:~$
```

2 Exercise 1.B

Figure 7: Steps 1 and 2



A terminal window titled 'cutie@Beauti: ~/hub/UNI/info/labs/lab1'. The user 'cutie' is at the prompt. The first command is 'touch anas.txt'. The second command is 'scp anas.txt hadi@timberlea.cs.dal.ca:~'. It prompts for a passphrase, then shows 'Permission denied' and 'failed to upload file anas.txt to /~'. The third command is 'scp anas.txt hadi@timberlea.cs.dal.ca:~'. It prompts for a passphrase, then shows '100% 0 0.0KB/s 00:00' and 'anas.txt'.

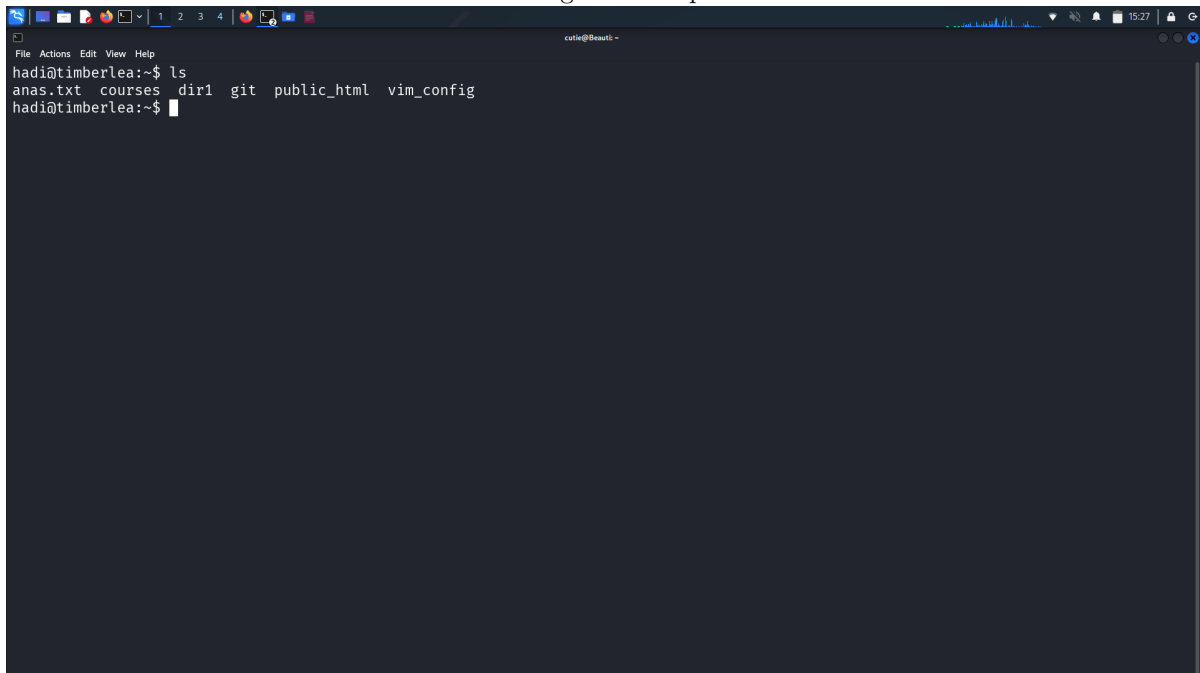
```
(cutie@Beauti)-[~/../UNI/info/labs/lab1]
$ touch anas.txt

(cutie@Beauti)-[~/../UNI/info/labs/lab1]
$ scp anas.txt hadi@timberlea.cs.dal.ca:~
Enter passphrase for key '/home/cutie/.ssh/id_ed25519':
scp: dest open "/~": Permission denied
scp: failed to upload file anas.txt to /~

(cutie@Beauti)-[~/../UNI/info/labs/lab1]
$ scp anas.txt hadi@timberlea.cs.dal.ca:~
Enter passphrase for key '/home/cutie/.ssh/id_ed25519':
anas.txt                                     100%  0  0.0KB/s  00:00

(cutie@Beauti)-[~/../UNI/info/labs/lab1]
$
```

Figure 8: Step 3

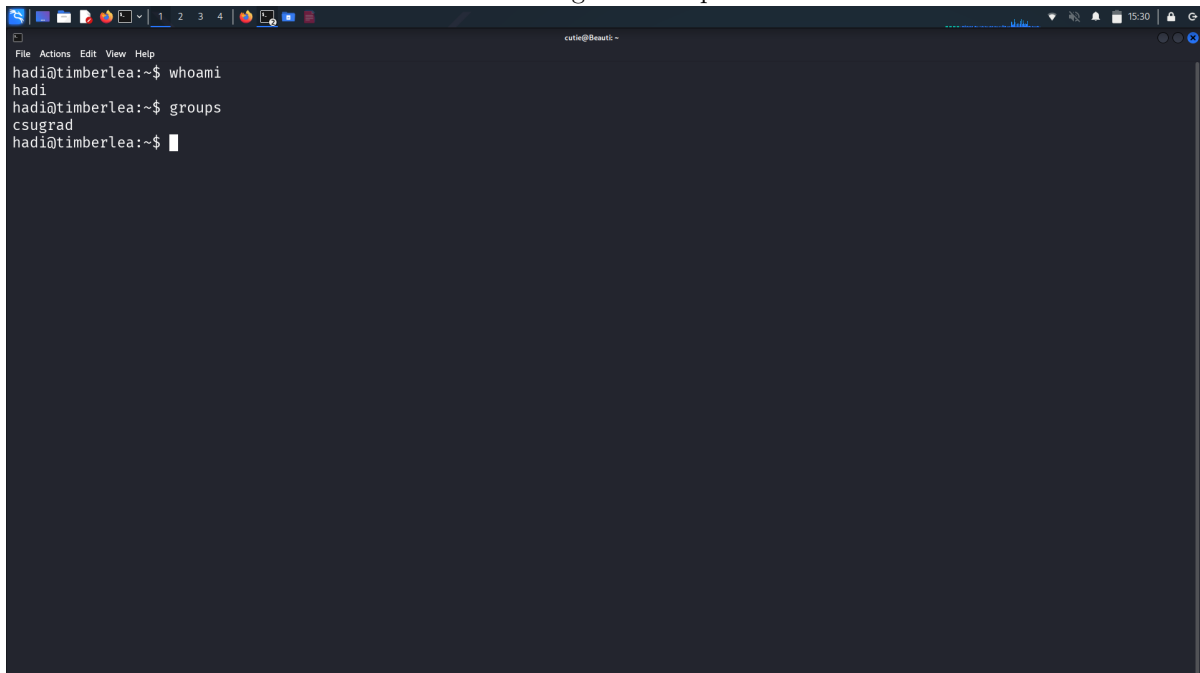


A terminal window titled 'cutie@Beauti: ~'. The user 'hadi' is at the prompt. The command 'ls' is executed, showing the output: 'anas.txt courses dir1 git public_html vim_config'.

```
hadi@timberlea:~$ ls
anas.txt  courses  dir1  git  public_html  vim_config
hadi@timberlea:~$
```

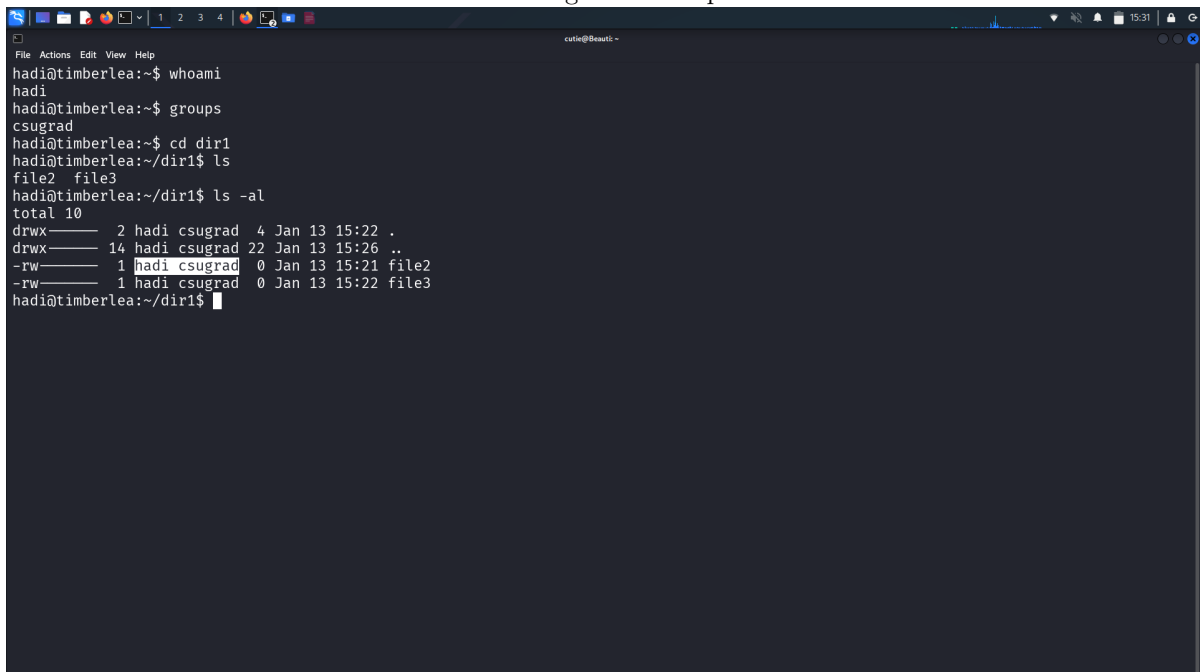
3 Exercise 2.A

Figure 9: Step



```
File Actions Edit View Help
hadi@timberlea:~$ whoami
hadi
hadi@timberlea:~$ groups
csugrad
hadi@timberlea:~$
```

Figure 10: Step

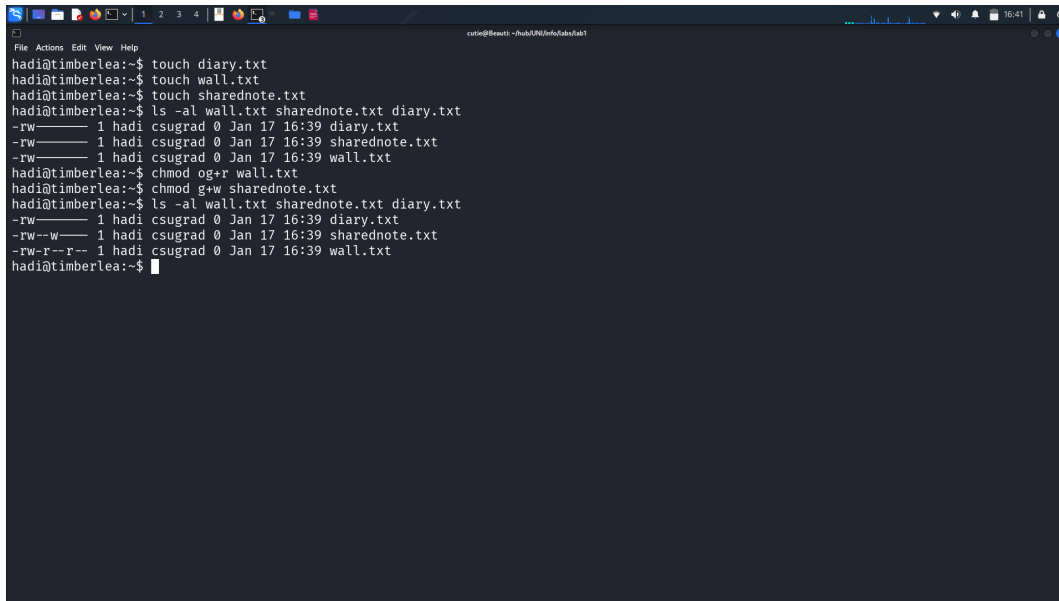


```
File Actions Edit View Help
hadi@timberlea:~$ whoami
hadi
hadi@timberlea:~$ groups
csugrad
hadi@timberlea:~$ cd dir1
hadi@timberlea:~/dir1$ ls
file2 file3
hadi@timberlea:~/dir1$ ls -al
total 10
drwx----- 2 hadi csugrad 4 Jan 13 15:22 .
drwx----- 14 hadi csugrad 22 Jan 13 15:26 ..
-rw----- 1 hadi csugrad 0 Jan 13 15:21 file2
-rw----- 1 hadi csugrad 0 Jan 13 15:22 file3
hadi@timberlea:~/dir1$
```

the **whoami** and **groups** commands do not take file names as arguments and return an error. **whoami** does not expect any user argument and only returns the name of the effective userID. **groups** can take a username as an argument and return the groups that they are a part of.

To check for the group and owner of a file we can run `ls -al <file_name>`

4 Exercise 2.B

A terminal window titled 'cario@beavrt: ~/hadi@timberlea\$' shows a series of commands and their outputs. The user 'hadi' creates three files: 'diary.txt', 'wall.txt', and 'sharednote.txt'. Then, they run 'ls -al' to show the files with permissions '-rw-rw-r--'. Next, they use 'chmod og+r wall.txt' and 'chmod g+w sharednote.txt'. A final 'ls -al' command shows the updated permissions: '-rw-r--r--' for 'diary.txt', '-rw--w--' for 'sharednote.txt', and '-rw-r--r--' for 'wall.txt'.


```
hadi@timberlea:~$ touch diary.txt
hadi@timberlea:~$ touch wall.txt
hadi@timberlea:~$ touch sharednote.txt
hadi@timberlea:~$ ls -al wall.txt sharednote.txt diary.txt
-rw-rw-r-- 1 hadi csugrad 0 Jan 17 16:39 diary.txt
-rw-rw-r-- 1 hadi csugrad 0 Jan 17 16:39 sharednote.txt
-rw-rw-r-- 1 hadi csugrad 0 Jan 17 16:39 wall.txt
hadi@timberlea:~$ chmod og+r wall.txt
hadi@timberlea:~$ chmod g+w sharednote.txt
hadi@timberlea:~$ ls -al wall.txt sharednote.txt diary.txt
-rw-r--r-- 1 hadi csugrad 0 Jan 17 16:39 diary.txt
-rw--w-- 1 hadi csugrad 0 Jan 17 16:39 sharednote.txt
-rw-r--r-- 1 hadi csugrad 0 Jan 17 16:39 wall.txt
hadi@timberlea:~$
```

Explanation:

1. diary.txt: No changes to the permissions were required since only the user "hadi" (me) can read and write to the file
2. wall.txt: The command i used can be reduced to only **o+r** as that includes everyone but still, giving every user in the csugrad group and everyone else the permission to read.
3. sharednote.txt: The command **g+w** assigns writing permission to everyone in the group that the file is assigned to.

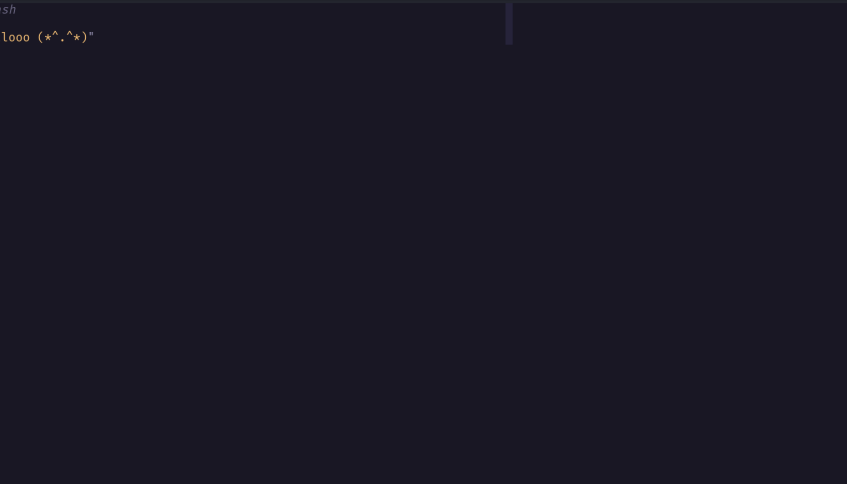
5 Exercise 3.A

Welcome:



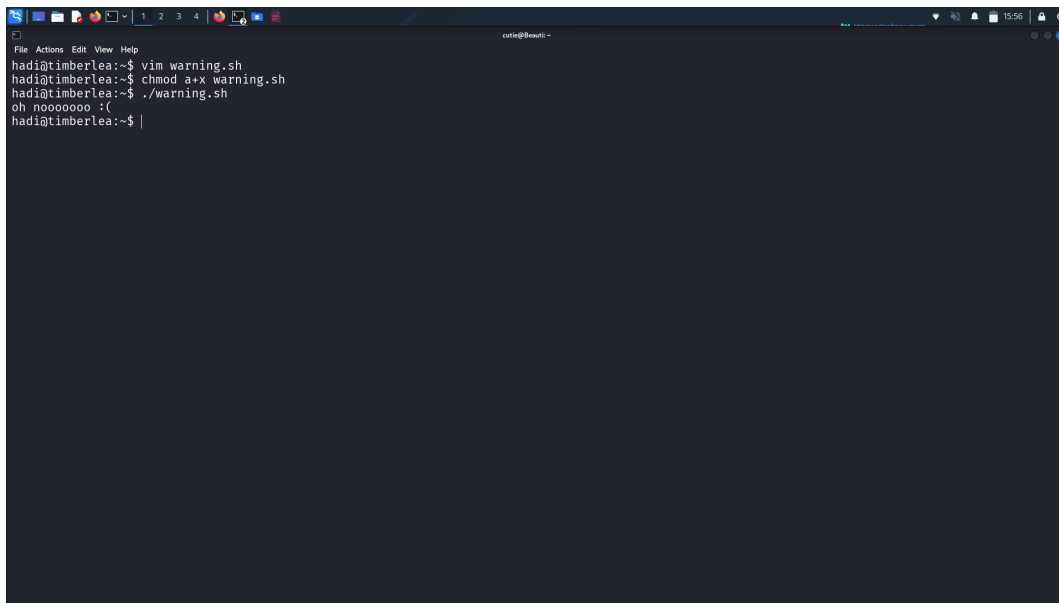
The screenshot shows a terminal window with a dark background. The window title is "curtin@buntu: ~". The terminal output is as follows:

```
hadia@timberlea:~$ rm welcome.sh
hadia@timberlea:~$ vim welcome.sh
hadia@timberlea:~$ chmod u+x welcome.sh
hadia@timberlea:~$ ./welcome.sh
hellooo (*^.*^*)
hadia@timberlea:~$ ls -al welcome.sh
-rwx----- 1 hadi csugrad 36 Jan 13 15:53 welcome.sh
hadia@timberlea:~$
```

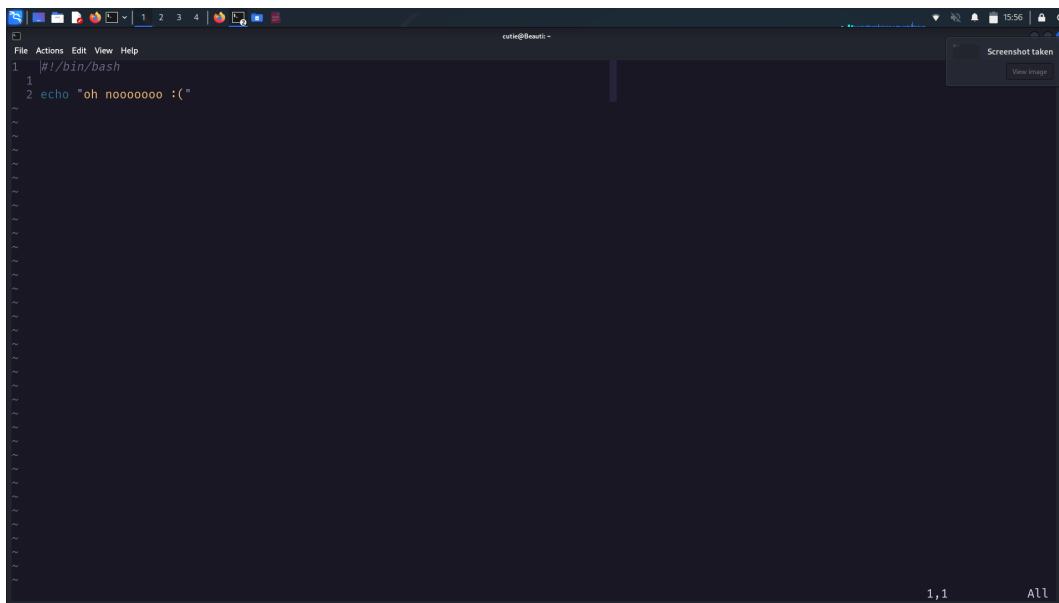


The screenshot shows a terminal window with a dark background. The title bar at the top indicates the user is 'curtin@beauti'. The terminal content shows a shell prompt '#!/bin/bash' followed by a command 'echo "hellooo (*^.*^*)"' on the second line. The cursor is positioned at the end of the command. The bottom right corner of the terminal displays '1,1' and 'Alt'.

Warning:



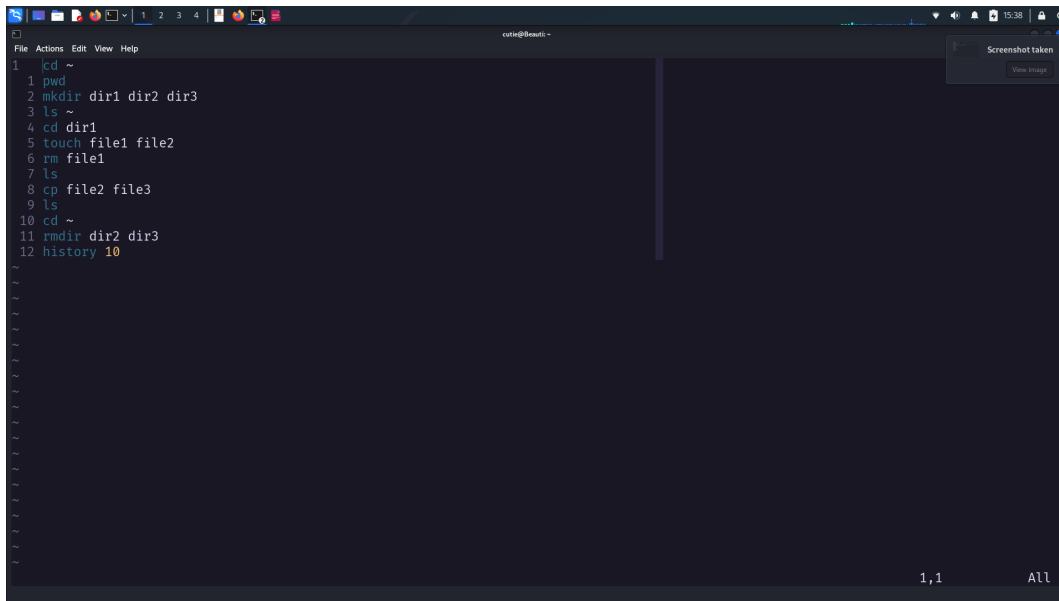
```
hadit@timberlea:~$ vim warning.sh
hadit@timberlea:~$ chmod a+x warning.sh
hadit@timberlea:~$ ./warning.sh
oh nooooooooo :(
hadit@timberlea:~$
```



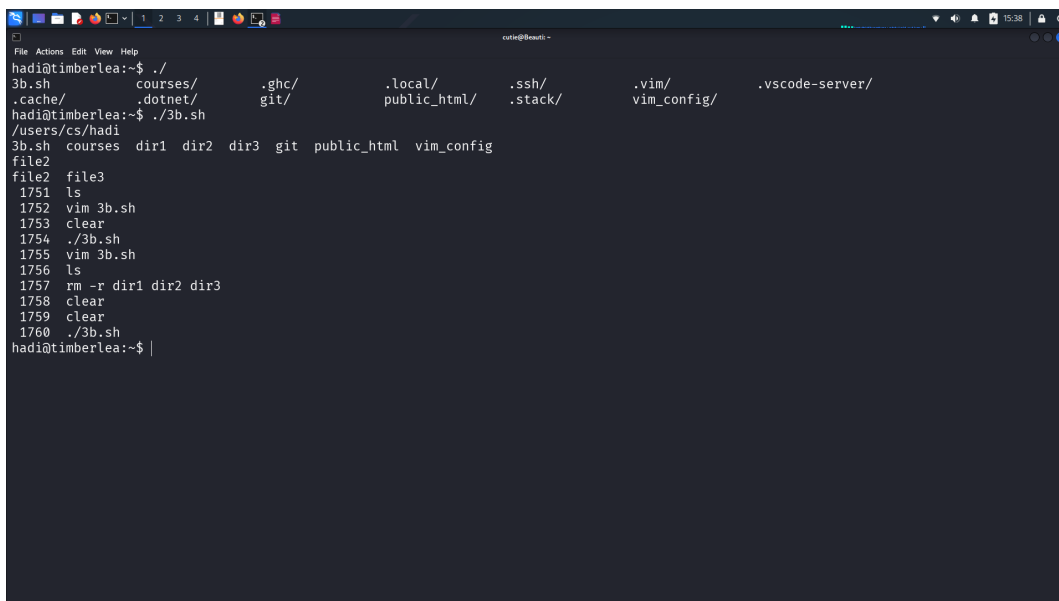
```
1 #!/bin/bash
2 echo "oh nooooooooo :("

1,1 All
```

6 Exercise 3.B



```
1 cd ~
2 pwd
3 mkdir dir1 dir2 dir3
4 ls ~
5 cd dir1
6 touch file1 file2
7 rm file1
8 ls
9 cp file2 file3
10 ls
11 cd ~
12 rmdir dir2 dir3
13 history 10
```



```
hadit@timberlea:~$ ./
3b.sh      courses/      .ghc/      .local/      .ssh/      .vim/      .vscode-server/
.cache/    .dotnet/     git/       public_html/ .stack/    vim_config/
hadit@timberlea:~$ ./3b.sh
/users/cs/hadi
3b.sh      courses      dir1      dir2      dir3      git      public_html      vim_config
file2
file2      file3
1751      ls
1752      vim 3b.sh
1753      clear
1754      ./3b.sh
1755      vim 3b.sh
1756      ls
1757      rm -r dir1 dir2 dir3
1758      clear
1759      clear
1760      ./3b.sh
hadit@timberlea:~$ |
```

Were the outputs different:

No, the output was the same

Why/why not to write a script:

We want to write scripts for complex and repetitive tasks. So no, I probably would not write a script to run basic commands (such as `ls`, `rm` ...). Scripts are better suited for longer commands for example recompiling a C program with external dependencies which would look something like:

```
gcc -std=c18 (long list of .c) -I/(path to library)/include -o exec -(some more flags)
```

In this case you would write a Makefile (or a shell script)