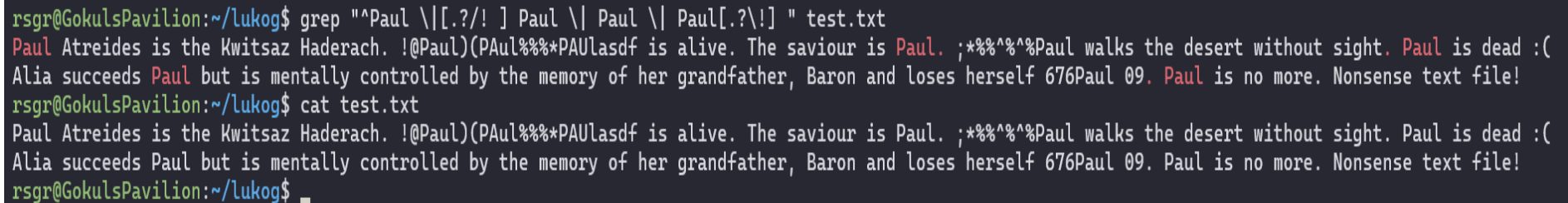


## Gokularamanan R S, 23B1854 - Week 1 Assignment Answers

### Q1) Fear is the mind-killer

```
$ grep "^Paul \|[.?! ] Paul \| Paul \| Paul[.?!]" test.txt
```



```
rsggr@GokulsPavilion:~/lukog$ grep "^Paul \|[.?! ] Paul \| Paul \| Paul[.?!]" test.txt
Paul Atreides is the Kwitsaz Haderach. !@Paul)(PAul%%*PAUlasdf is alive. The saviour is Paul. ;*%^^%Paul walks the desert without sight. Paul is dead :(
Alia succeeds Paul but is mentally controlled by the memory of her grandfather, Baron and loses herself 676Paul 09. Paul is no more. Nonsense text file!
rsggr@GokulsPavilion:~/lukog$ cat test.txt
Paul Atreides is the Kwitsaz Haderach. !@Paul)(PAul%%*PAUlasdf is alive. The saviour is Paul. ;*%^^%Paul walks the desert without sight. Paul is dead :(
Alia succeeds Paul but is mentally controlled by the memory of her grandfather, Baron and loses herself 676Paul 09. Paul is no more. Nonsense text file!
rsggr@GokulsPavilion:~/lukog$
```

Figure 1. screenshot of terminal with the cmd and output

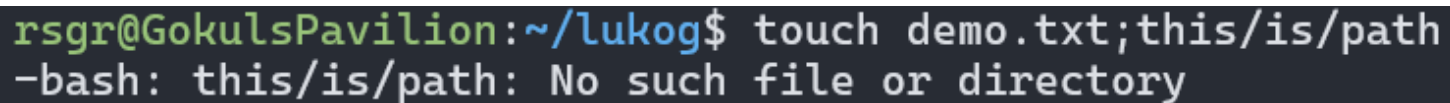
Explanation :

- I am checking for four cases :
  1. Start of the text - "**^Paul**" (^Paul looks for first word of every line, hence checks whether Paul is at the start of the text (line))
  2. In the middle of the sentence - " **Paul** " (2 spaces before and after Paul checks for Paul in the middle of a sentence)
  3. At the ending of a sentence (followed by {.,?!} - punctuation marks) - " **Paul[.?!]**" (a space before Paul and followed immediately by a punctuation mark- full stop or question mark or exclamation mark (escape sequence))
  4. At the start of a new sentence, but not the start of the text (line) - "**[.?!] Paul** " (exactly opposite to the previous case. Here, since Paul is at the start of the sentence, it must be preceded by a punctuation mark followed by a space and the succeeded by a space)

## Q2) A Basic File Manager

### Part 1.

```
$ touch demo.txt;this/is/path
```



```
rsgr@GokulsPavilion:~/lukog$ touch demo.txt;this/is/path  
-bash: this/is/path: No such file or directory
```

*Figure 2. Screenshot of terminal with the cmd and output*

Explanation :

- `$ touch demo.txt;this/is/path` consists of 2 commands – `touch demo.txt` and `this/is/path`
- The first one, `touch demo.txt` creates a txt file named “demo” in the current working directory, while the other checks whether the provided name is a valid path/file name in the current working directory.
- `touch demo.txt` is a background process, therefore, there’s no output provided in the shell after the completion of the process. Whereas, `this/is/path` is a foreground process. After the shell checks for any existing file/ path in the current working directory, it returns the complete path if it exists, otherwise returns something similar to what has been given : `-bash: this/is/path: No such file or directory`
- Since the provided path `this/is/path` doesn’t exist in the working directory, the shell says, “No such file or directory”
- `;` is used to separate commands in bash shell and executes them sequentially, similar to `&&`

*Figure 3. Screenshot showing contents of ~/lukog before and after execution of cmds demo.txt and this/is/path*

```

rsgr@GokulsPavilion:~/lukog$ ls -l
total 56
-rwxr-xr-x 1 rsgr rsgr 16376 May 23 14:48 a.out
-rw-r--r-- 1 rsgr rsgr 0 May 28 10:00 del4.jpg
-rw-r--r-- 1 rsgr rsgr 65 May 28 10:10 dew.cpp
-rw-r--r-- 1 rsgr rsgr 88 May 27 17:45 hello_readme.txt
-rw-r--r-- 1 rsgr rsgr 0 May 27 17:28 lukog
drwxr-xr-x 3 rsgr rsgr 4096 May 27 17:36 minor1
-rw-r--r-- 1 rsgr rsgr 101 May 27 16:35 new.cpp
-rw-r--r-- 1 rsgr rsgr 687 Jun 1 22:27 novel.txt
-rw-r--r-- 1 rsgr rsgr 80 Jun 1 22:36 path.txt
-rw-r--r-- 1 rsgr rsgr 310 Jun 2 00:09 test.txt
-rw-r--r-- 1 rsgr rsgr 112 May 28 10:41 the-real-deal.txt
-rw-r--r-- 1 rsgr rsgr 116 Jun 2 07:35 timestamp.txt
-rw-r--r-- 1 rsgr rsgr 44 Jun 1 23:49 v.txt
rsgr@GokulsPavilion:~/lukog$ touch demo.txt;this/is/path
-bash: this/is/path: No such file or directory
rsgr@GokulsPavilion:~/lukog$ ls -l
total 56
-rwxr-xr-x 1 rsgr rsgr 16376 May 23 14:48 a.out
-rw-r--r-- 1 rsgr rsgr 0 May 28 10:00 del4.jpg
-rw-r--r-- 1 rsgr rsgr 0 Jun 2 08:14 demo.txt
-rw-r--r-- 1 rsgr rsgr 65 May 28 10:10 dew.cpp
-rw-r--r-- 1 rsgr rsgr 88 May 27 17:45 hello_readme.txt
-rw-r--r-- 1 rsgr rsgr 0 May 27 17:28 lukog
drwxr-xr-x 3 rsgr rsgr 4096 May 27 17:36 minor1
-rw-r--r-- 1 rsgr rsgr 101 May 27 16:35 new.cpp
-rw-r--r-- 1 rsgr rsgr 687 Jun 1 22:27 novel.txt
-rw-r--r-- 1 rsgr rsgr 80 Jun 1 22:36 path.txt
-rw-r--r-- 1 rsgr rsgr 310 Jun 2 00:09 test.txt
-rw-r--r-- 1 rsgr rsgr 112 May 28 10:41 the-real-deal.txt
-rw-r--r-- 1 rsgr rsgr 116 Jun 2 07:35 timestamp.txt
-rw-r--r-- 1 rsgr rsgr 44 Jun 1 23:49 v.txt
rsgr@GokulsPavilion:~/lukog$

```

In the directory ~/lukog, the directories /this/is/path don't exist. If they did, i.e., ~/lukog/this/is/path did exist, then the shell would have provided an output similar to -bash: ./this/is/path : Is a directory

## Part 2

```
((tr "/" "&" < path.txt) > pathi.txt) && (for i in $(cat pathi.txt); do (touch $(echo $i));done)
```

Explanation :

- There are two parts : `((tr "/" "&" < path.txt) > pathi.txt)` and `(for i in $(cat pathi.txt); do (touch $(echo $i));done)`
- In the first one, `((tr "/" "&" < path.txt) > pathi.txt)` , I am replacing all occurrences of '/' in path.txt with '&', using the tr command. Then, I am redirecting this output to a text file called pathi.txt (which now contains ).
- Using &&, I am sequentially executing the for loop commands after the tr command.
- In the second part, `(for i in $(cat pathi.txt); do (touch $(echo $i));done)` is a for loop statement, wherein I am reading through every line of pathi.txt using `(for i in $(cat pathi.txt)` (cat command is used to display the contents of text files). Then, for every line in pathi.txt, I am creating a file with the same name as the line using `do (touch $(echo $i));done)`
- `touch $(echo $i )` is used to create a file with filename as each line in pathi.txt

The newly created files  
named after each line of  
pathi.txt

```
rsgr@GokulsPavilion:~/lukog$ ls -l
total 68
-rw-r--r-- 1 rsgr rsgr 50 Jun 2 13:51 3rdquestion.txt
-rwxr-xr-x 1 rsgr rsgr 16376 May 23 14:48 a.out
-rw-r--r-- 1 rsgr rsgr 0 May 28 10:00 del4.jpg
-rw-r--r-- 1 rsgr rsgr 65 May 28 10:10 dew.cpp
-rw-r--r-- 1 rsgr rsgr 413 Jun 2 13:34 hello_readme.txt
-rw-r--r-- 1 rsgr rsgr 35 Jun 2 13:30 key.txt
-rw-r--r-- 1 rsgr rsgr 0 May 27 17:28 lukog
drwxr-xr-x 3 rsgr rsgr 4096 Jun 2 14:56 minor1
-rw-r--r-- 1 rsgr rsgr 101 May 27 16:35 new.cpp
-rw-r--r-- 1 rsgr rsgr 687 Jun 1 22:27 novel.txt
-rw-r--r-- 1 rsgr rsgr 80 Jun 1 22:36 path.txt
-rw-r--r-- 1 rsgr rsgr 310 Jun 2 00:09 test.txt
-rw-r--r-- 1 rsgr rsgr 112 May 28 10:41 the-real-deal.txt
-rw-r--r-- 1 rsgr rsgr 174 Jun 2 13:35 timestamp.txt
-rw-r--r-- 1 rsgr rsgr 44 Jun 1 23:49 v.txt
-rw-r--r-- 1 rsgr rsgr 80 Jun 2 13:55 working.txt
rsgr@GokulsPavilion:~/lukog$ ((tr "[/]" "[&]" < path.txt) > pathi.txt) && (for i in $(cat pathi.txt); do (touch $(echo $i));done)
rsgr@GokulsPavilion:~/lukog$ ls -l
total 72
-rw-r--r-- 1 rsgr rsgr 50 Jun 2 13:51 3rdquestion.txt
-rw-r--r-- 1 rsgr rsgr 0 Jun 2 16:02 'Began;end&'
-rw-r--r-- 1 rsgr rsgr 0 Jun 2 16:02 'Hello_myworld.pdf;.&this&is&not&end'
-rw-r--r-- 1 rsgr rsgr 0 Jun 2 16:02 'Ohon;ends&'
-rwxr-xr-x 1 rsgr rsgr 16376 May 23 14:48 a.out
-rw-r--r-- 1 rsgr rsgr 0 May 28 10:00 del4.jpg
-rw-r--r-- 1 rsgr rsgr 0 Jun 2 16:02 'demo.txt;this&is&path'
-rw-r--r-- 1 rsgr rsgr 65 May 28 10:10 dew.cpp
-rw-r--r-- 1 rsgr rsgr 413 Jun 2 13:34 hello_readme.txt
-rw-r--r-- 1 rsgr rsgr 35 Jun 2 13:30 key.txt
-rw-r--r-- 1 rsgr rsgr 0 May 27 17:28 lukog
drwxr-xr-x 3 rsgr rsgr 4096 Jun 2 14:56 minor1
-rw-r--r-- 1 rsgr rsgr 101 May 27 16:35 new.cpp
-rw-r--r-- 1 rsgr rsgr 687 Jun 1 22:27 novel.txt
-rw-r--r-- 1 rsgr rsgr 80 Jun 1 22:36 path.txt
-rw-r--r-- 1 rsgr rsgr 80 Jun 2 16:02 pathi.txt
-rw-r--r-- 1 rsgr rsgr 310 Jun 2 00:09 test.txt
-rw-r--r-- 1 rsgr rsgr 112 May 28 10:41 the-real-deal.txt
-rw-r--r-- 1 rsgr rsgr 203 Jun 2 15:35 timestamp.txt
-rw-r--r-- 1 rsgr rsgr 44 Jun 1 23:49 v.txt
-rw-r--r-- 1 rsgr rsgr 80 Jun 2 13:55 working.txt
rsgr@GokulsPavilion:~/lukog$ _
```

Contents of the directory before

Contents of the directory after

```
rsgr@GokulsPavilion:~/lukog$ cat path.txt
demo.txt;this/is/path
Hello_myworld.pdf;./this/is/not/end
Began;end/
Ohon;ends/
rsgr@GokulsPavilion:~/lukog$ cat pathi.txt
demo.txt;this&is&path
Hello_myworld.pdf;.&this&is&not&end
Began;end&
Ohon;ends&
rsgr@GokulsPavilion:~/lukog$ _
```

### Part 3

```
(echo './' > dot.txt) && (for i in $(cat path.txt); do ( echo $i > temp-line.txt) && ((tr "[/]" "[&]" < temp-line.txt) > pre-temp-name.txt)&& (echo "" > quote.txt) && (paste -d " " quote.txt pre-temp-name.txt quote.txt > temp-name.txt) && ((cut -d ';' -f2 temp-line.txt) > pre-temp-path.txt) && (paste -d " " dot.txt pre-temp-path.txt > temp-path.txt) && (mkdir -p $(cat pre-temp-path.txt)) && (mv $(cat pre-temp-name.txt) $(cat temp-path.txt)) && ((cut -d ';' -f1 temp-line.txt) > name.txt) && (cp pre-temp-name.txt name.txt temp-path.txt pre-temp-path.txt $(cat temp-path.txt)) && cd $(cat temp-path.txt) && (mv $(cat pre-temp-name.txt) $(cat name.txt)) && ( rm pre-temp-name.txt name.txt temp-path.txt pre-temp-path.txt ) && cd ~/lukog; done)
```

My working directory is ~/lukog . Hence, I am coming back to it so that I can move all files to their respective subdirectories. (I tried using a nested for- loop, but it failed for some reason (it worked for the 1st and 2nd lines of path.txt but failed for the 3rd and 4th) - (for i in \$(seq \$(grep -o '/' pre-temp-path.txt | wc -l));do cd ../;done))

#### Explanation:

- **(echo './' > dot.txt) :** I am creating a file named “dot.txt” which contains basically ./ The purpose of dot.txt is for moving files to specific directories within the parent directory. (WILL BE EXPLAINED LATER WHEN mv IS BEING DISCUSSED)
- The for loop :
  - **for i in \$(cat path.txt)** I am iterating through path.txt, reading through each line
  - **do (echo \$i > temp-line.txt)** Here, I am writing each line to a file named temp-line.txt. If i = demo.txt;this/is/path, then “ demo.txt;this/is/path ” would be present in temp-file.txt . This file will be overwritten every iteration.

- `&& ((tr "[/]" "[&]" < temp-line.txt) > pre-temp-name.txt)` The content in temp-name.txt that was of the form `<name>;dir1/dir2/dir3..` Is stored as `<name>;dir1&dir2&dir3`
- `&& (echo "" > quote.txt) && (paste -d " quote.txt pre-temp-name.txt quote.txt > temp-name.txt)` In the last part, the files that were saved are of the form `'demo.txt;this/is/path'` with the quotes included, hence I am adding single quotes to the file and saving it.

- `&& ((cut -d ';' -f1 temp-line.txt) > temp-name.txt)` Now, using cut operation, I am specifically selecting the field 1 (all characters before the delimiter ';') and writing it to a file named temp-name.txt .

Therefore, if temp-line.txt contains " demo.txt;this/is/path " , then "demo.txt" will be written to temp-name.txt .

- Similarly, `&& ((cut -d ';' -f2 temp-line.txt) > pre-temp-path.txt)` Here, again using the cut command, I am selecting the field 2 (all characters after the delimiter ';') and writing it to a file named pre-temp-path.txt .

Therefore, if pre-temp-path.txt contains " demo.txt;this/is/path " , then "this/is/path" will be written to pre-temp-path.txt .

- `&& (paste -d " dot.txt pre-temp-path.txt > temp-path.txt)` Here, I am combining the contents of dot.txt (contains "./") and pre-temp-path.txt . The delimiter used is ' ' - empty string, so the merged content is without any break or space.

If pre-temp-path.txt contains "this/is/path" , then the new file temp-path.txt will contain "/this/is/path"

This is done so as to ensure that files can be moved from the current working directory to a subdirectory within the current working directory.

- `&& mkdir -p $(cat pre-temp-path.txt)` Since the paths given in `path.txt` alongside the filename may not exist, I am creating them. If `pre-temp-path.txt` contains “this/is/path”, then a new directory `this/is/path` is recursively created under the current directory.
- `(cp pre-temp-name.txt name.txt temp-path.txt pre-temp-path.txt $(cat temp-path.txt))` I am copying all these files to each of the subdirectories so as to facilitate renaming the *filename;path* to *filename* . Each of the files will be different in each subdirectory, each containing info related to the respective filename.
- `&& mv $(cat pre-temp-name.txt) $(cat temp-path.txt); done)` Here, I am moving all the files created in the previous part (part 2) to their respective directories (*name;path* was the way these file names were provided in `path.txt`, Hence, I am moving these newly created files, such as `demo.txt`, `Hello_myworld.pdf` to paths such as `/this/is/path` and `/this/is/not` respectively).
- `cd $(cat temp-path.txt)` I am moving to each of the subdirectories in order to rename the files.
- `(mv $(cat pre-temp-name.txt) $(cat name.txt)) && ( rm pre-temp-name.txt name.txt temp-path.txt pre-temp-path.txt )` Renaming the files (example, `demo.txt;this&is&path` becomes `demo.txt`) and then deleting the extra files that I copied in order to rename.

Now, if I try to use `mv $(cat temp-name.txt) $(cat pre-temp-path.txt)`, this is equivalent to `mv demo.txt /this/is/path` . However, this raises an error since we are actually inside a directory (say `desktop`, not `root` or `home`). Hence, we have use `mv demo.txt ./this/is/path` which says that we are moving `demo.txt` from (say) `~/Desktop/demo.txt` to `~/Desktop/this/is/path`

Hence, I am creating `dot.txt`, `temp-path.txt`



```

rsgr@GokulsPavilion:~/lukog$ ((tr "[/]" "[&]" < path.txt > pathi.txt) && (for i in $(cat pathi.txt); do (touch $(echo $i));done)
rsgr@GokulsPavilion:~/lukog$ (echo './' > dot.txt) && (for i in $(cat pathi.txt); do ( echo $i > temp-line.txt) && ((tr "[/]" "[&]" < temp-line.txt) > pre-temp-name.txt) && (echo "" > quote.txt) && (paste -d ' ' quote.txt pre-temp-name.txt quote.txt > temp-name.txt) && ((cut -d ';' -f2 temp-line.txt) > pre-temp-path.txt) && (paste -d ' ' dot.txt pre-temp-path.txt > temp-path.txt) && (mkdir -p $(cat pre-temp-path.txt)) && (mv $(cat pre-temp-name.txt) $(cat temp-path.txt)) && ((cut -d ';' -f1 temp-line.txt) > name.txt) && (cp pre-temp-name.txt name.txt temp-path.txt pre-temp-path.txt $(cat temp-path.txt)) && cd $(cat temp-path.txt) && (mv $(cat pre-temp-name.txt) $(cat name.txt)) && ( rm pre-temp-name.txt name.txt temp-path.txt pre-temp-path.txt ) && cd ~/lukog; done)
rsgr@GokulsPavilion:~/lukog$ ls -l
total 152
-rw-r--r-- 1 rsgr rsgr 50 Jun  2 18:37 3rdquestion.txt
-rwxr-xr-x 1 rsgr rsgr 16376 May 23 14:48 a.out
-rw-r--r-- 1 rsgr rsgr 0 May 28 10:00 del4.jpg
-rw-r--r-- 1 rsgr rsgr 15 Jun  2 18:17 ding.txt
-rw-r--r-- 1 rsgr rsgr 3 Jun  2 23:51 dot.txt
-rwxr-xr-x 2 rsgr rsgr 4096 Jun  2 23:51 end
-rwxr-xr-x 2 rsgr rsgr 4096 Jun  2 23:51 ends
-rw-r--r-- 1 rsgr rsgr 15 Jun  2 18:16 final.txt
-rw-r--r-- 1 rsgr rsgr 413 Jun  2 13:34 hello_readme.txt
-rw-r--r-- 1 rsgr rsgr 0 Jun  2 20:28 i
-rw-r--r-- 1 rsgr rsgr 0 Jun  2 20:39 ivar.txt
-rw-r--r-- 1 rsgr rsgr 35 Jun  2 13:30 key.txt
-rw-r--r-- 1 rsgr rsgr 2 Jun  2 20:55 left.txt
-rw-r--r-- 1 rsgr rsgr 0 May 27 17:28 lukog
-rwxr-xr-x 5 rsgr rsgr 4096 Jun  2 23:04 minori
-rw-r--r-- 1 rsgr rsgr 5 Jun  2 23:51 name.txt
-rw-r--r-- 1 rsgr rsgr 101 May 27 16:35 new.cpp
-rw-r--r-- 1 rsgr rsgr 0 Jun  2 18:18 nib.txt
-rw-r--r-- 1 rsgr rsgr 21 Jun  2 21:40 nine.txt
-rw-r--r-- 1 rsgr rsgr 607 Jun  1 22:27 novel.txt
-rw-r--r-- 1 rsgr rsgr 2 Jun  2 18:13 part1.txt
-rw-r--r-- 1 rsgr rsgr 14 Jun  2 18:11 part2.txt
-rw-r--r-- 1 rsgr rsgr 80 Jun  1 22:36 path.txt
-rw-r--r-- 1 rsgr rsgr 80 Jun  2 23:50 pathi.txt
-rw-r--r-- 1 rsgr rsgr 11 Jun  2 23:51 pre-temp-name.txt
-rw-r--r-- 1 rsgr rsgr 6 Jun  2 23:51 pre-temp-path.txt
-rwxr-xr-x 3 rsgr rsgr 4096 Jun  2 21:05 project
-rw-r--r-- 1 rsgr rsgr 2 Jun  2 23:51 quote.txt
-rw-r--r-- 1 rsgr rsgr 19 Jun  2 20:39 qwr.txt
-rw-r--r-- 1 rsgr rsgr 21 Jun  2 20:56 repeat.txt
-rw-r--r-- 1 rsgr rsgr 11 Jun  2 23:51 temp-line.txt
-rw-r--r-- 1 rsgr rsgr 13 Jun  2 23:51 temp-name.txt
-rw-r--r-- 1 rsgr rsgr 8 Jun  2 23:51 temp-path.txt
-rw-r--r-- 1 rsgr rsgr 11 Jun  2 20:30 tempo.txt
-rw-r--r-- 1 rsgr rsgr 310 Jun  2 00:09 test.txt
-rw-r--r-- 1 rsgr rsgr 112 May 28 10:41 the-real-deal.txt
-rwxr-xr-x 3 rsgr rsgr 4096 Jun  2 23:51 this
-rw-r--r-- 1 rsgr rsgr 406 Jun  2 23:35 timestamp.txt
-rw-r--r-- 1 rsgr rsgr 44 Jun  1 23:49 v.txt
-rw-r--r-- 1 rsgr rsgr 80 Jun  2 13:55 working.txt

```

```

rsgr@GokulsPavilion:~/lukog$ cd ./this/is/path
rsgr@GokulsPavilion:~/lukog/this/is/path$ ls -l
total 0
-rw-r--r-- 1 rsgr rsgr 0 Jun  2 23:50 demo.txt
rsgr@GokulsPavilion:~/lukog/this/is/path$ cd ../
rsgr@GokulsPavilion:~/lukog/this/is$ cd ./not
rsgr@GokulsPavilion:~/lukog/this/is/not$ cd ./end
rsgr@GokulsPavilion:~/lukog/this/is/not/end$ ls -l
total 0
-rw-r--r-- 1 rsgr rsgr 0 Jun  2 23:50 Hello_myworld.pdf
rsgr@GokulsPavilion:~/lukog/this/is/not/end$ cd ~/lukog/end
rsgr@GokulsPavilion:~/lukog/end$ ls -l
total 0
-rw-r--r-- 1 rsgr rsgr 0 Jun  2 23:50 Began
rsgr@GokulsPavilion:~/lukog/end$ cd ~/lukog/ends
rsgr@GokulsPavilion:~/lukog/ends$ ls -l
total 0
-rw-r--r-- 1 rsgr rsgr 0 Jun  2 23:50 Ohon
rsgr@GokulsPavilion:~/lukog/ends$

```

