

Experiment Number: 02

Problem Statement: Execution of advance Linux commands

NAME: Omkar Sunil Khanvilkar

ROLLNO: 7

CLASS: IT-B

BATCH: 2

DATE OF PERFORMANCE: 26/7/2024

- 1) pr :pr command is used to prepare a file for printing by adding suitable footers, headers, and the formatted text. pr command actually adds 5 lines of margin both at the top and bottom of the page. The header part shows the date and time of the last modification of the file with the file name and the page number.

pr fileName

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ pr -a myName.txt  
  
2024-07-20 00:59           myName.txt          Page 1  
  
Hii Omkar  
Hii Omkar  
Hello I am Soham  
  
My Brother name is Soham  
My mother Name is Nandana  
  
My Father Name is Sunil
```

-n To provide number lines which helps in debugging the code -n option is used.

pr -n fileName

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ pr -n Friends.txt
```

```
2024-07-22 23:44
```

```
Friends.txt
```

```
Page 1
```

```
1 Omkar  
2 Soham  
3 Ajay  
4 Raju  
5 Sarthak  
6 Pankaj
```

-k where k is any integer for example if we want to print this content in 3 columns use -3

pr -3 fileName

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ pr -3 Friends.txt
```

```
2024-07-22 23:44
```

```
Friends.txt
```

```
Page 1
```

```
Omkar Ajay Sarthak  
Soham Raju Pankaj
```

-d To Double the spaces input, reduces clutter -d option is used.

pr -d fileName

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ pr -d Friends.txt
```

```
2024-07-22 23:44
```

```
Friends.txt
```

```
Page 1
```

```
Omkar  
Soham  
Ajay  
Raju  
Sarthak  
Pankaj
```

- 2) head Command is use to Display first 10 lines of file

head fileName

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ cat Friends.txt
Omkar
Soham
Ajay
Raju
Sarthak
Pankaj
Rahul
Omi
Raju
Amit
Tapu
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ head Friends.txt
Omkar
Soham
Ajay
Raju
Sarthak
Pankaj
Rahul
Omi
Raju
Amit
```

-k option use to print he first k lines of the file

Head -k fileName

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ head -4 Friends.txt
Omkar
Soham
Ajay
Raju
```

-z option line delimiter is NUL, not newline

Head -z fileName

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ head -z Friends.txt
Omkar
Soham
Ajay
Raju
Sarthak
Pankaj
Rahul
Omi
Raju
Amit
Tapu
```

- 3) tail : tail command is use to print last 10 lines of files

tail fileName

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ cat -n Friends.txt
1 Omkar
2 Soham
3 Ajay
4 Raju
5 Sarthak
6 Pankaj
7 Rahul
8 Omi
9 Raju
10 Amit
11 Tapu
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ tail Friends.txt
Soham
Ajay
Raju
Sarthak
Pankaj
Rahul
Omi
Raju
Amit
Tapu
```

-c option c is use to get last n bytes of the file where n is any integer

tail -c 100 fileName

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ cat -n Friends.txt
      1 Omkar
      2 Soham
      3 Ajay
      4 Raju
      5 Sarthak
      6 Pankaj
      7 Rahul
      8 Omi
      9 Raju
     10 Amit
     11 Tapu
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ tail -c 21 Friends.txt
l
Omi
Raju
Amit
Tapu
```

-n option n is use (where n is any integer) to print last n rows of the file

tail -10 fileName

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ tail -3 Friends.txt
Raju
Amit
Tapu
```

- 4) cut command in Linux is used to extract sections from each line of input, typically from a text file or piped output.

Cut command is always used with options

Cut option fileName

-b [bytes from-to] option b is use to cut bytes from the text data

cut -b range fileName

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ cut -b 1-2 Friends.txt
Om
So
Aj
Ra
Sa
Pa
Ra
Om
Ra
Am
Ta
```

-d Specify the delimiter. Default is tab. We can use any delimiter (character) and using fields we can access all fields

cut -d'delimiter_Char' -f1/2/3 fileName

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ cat StudentData.txt
Omkar,10,ITB
Soham,12,ITC
Ajay,17,ITC
```

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ cut -d',' -f1 StudentData.txt
Omkar
Soham
Ajay
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ cut -d',' -f2 StudentData.txt
10
12
17
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ cut -d',' -f3 StudentData.txt
ITB
ITC
ITC
```

-c option c is use to cut the specific number of character from each row

cut -c range fileName

Option c have same functionality like option b but option c support multibyte characters (e.g., UTF-8 encoded characters). But b not support multibyte character

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ cut -b 1-5 ter.txt
Héll
Cut c
12345
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ cut -c 1-5 ter.txt
Héll
Cut c
12345
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ cat ter.txt
Héll, Wörl! Linux is awésome
Cut ciòmand is useful
1234567890
```

- 5) paste command in Linux is used to merge lines of files side by side

paste file1 file2

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ cat num.txt
1
2
3
4
5
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ cat Alphabet.txt
A
B
C
D
E
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ paste num.txt Alphabet.txt
1      A
2      B
3      C
4      D
5      E
```

-s option to paste files serially

Paste -s file1 file2

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ paste -s num.txt Alphabet.txt
1      2      3      4      5
A      B      C      D      E
```

-d To use a comma as a delimiter:

Paste -d',' file1 file2

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ paste -d ',' num.txt Alphabet.txt
1,A
2,B
3,C
4,D
5,E
```

Paste -d',' file1 file2 file3..

It also Use to merge more than 2 files

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ paste -d '-' studentName.txt studentRoll.txt studentDiv.txt
Omkar-10-ITB
Soham-12-ITC
Ajay-17-ITC
```

6) sort - sort lines of text files

sort filename

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ sort Friends.txt
Ajay
Amit
Omi
Omkar
Pankaj
Rahul
Raju
Raju
Sarthak
Soham
Tapu
```

-n sort the file having numerical data

sort -n filename

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ sort -n Num1.txt
0
8
9
78
123
324
```

-r option is use to perform reverse sort

sort -r filename

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ sort -nr Num1.txt
324
123
78
9
8
0
```

-u option u is to sort the unique element which eliminate the repeated element

sort -u filename

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ sort -n Num1.txt
0
8
9
45
45
45
45
78
123
324
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ sort -nu Num1.txt
0
8
9
45
78
123
324
```

- 7) Uniq command is a command useful to sort lines of text

Normal uniq command is only use to remove duplicate content but the condition is that the content is must be present on adjacent lines

uniq filename

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ cat Friends.txt
Omkar
Soham
Ajay
Raju
Sarthak
Pankaj
Rahul
Omi
Raju
Amit
Tapu
Omkar
Raju
Raju
Sarthak
Omkar
Sarthak
Omkar
Ajay
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ uniq Friends.txt
Omkar
Soham
Ajay
Raju
Sarthak
Pankaj
Rahul
Omi
Raju
Amit
Tapu
Omkar
Raju
Sarthak
Omkar
Sarthak
Omkar
Ajay
```

But if we need to do it for all names not for adjacent names then we need to use this with sort using pipe operator (same as -u option of sort command)

sort filename | uniq

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ sort Friends.txt | uniq
Ajay
Amit
Omi
Omkar
Pankaj
Rahul
Raju
Sarthak
Soham
Tapu
```

-d option of uniq command is to get only duplicate elements from the file data

sort filename | uniq -d

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ sort Friends.txt | uniq -d
Ajay
Omkar
Raju
Sarthak
```

-u option is use to get non duplicate content from the file data

sort filename | uniq -u

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ sort Friends.txt | uniq -u
Amit
Omi
Pankaj
Rahul
Soham
Tapu
```

-c gives us actual count of data repeated inside the file

sort filename | uniq -c

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ sort Friends.txt | uniq -c
 2 Ajay
 1 Amit
 1 Omi
 4 Omkar
 1 Pankaj
 1 Rahul
 4 Raju
 3 Sarthak
 1 Soham
 1 Tapu
```

We can sort the above output to get more insights by simply piping with the sort -n

```
sort filename | uniq -c | sort -n
```

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ sort Friends.txt | uniq -c | sort -n
      1 Amit
      1 Omi
      1 Pankaj
      1 Rahul
      1 Soham
      1 Tapu
      2 Ajay
      3 Sarthak
      4 Omkar
      4 Raju
```

- 8) tr command is use to translate the particular character or set of characters with some different character

```
cat filename | tr 'a-z' 'A-Z'
```

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ cat studentName.txt
Omkar
Soham
Ajay
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ cat studentName.txt | tr 'a-z' 'A-Z'
OMKAR
SOHAM
AJAY
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$
```

-d option with this command is use to delete the specified character or character set from the content passed to it

```
echo "content" | tr -d 'characterRemove'
```

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ echo "omkar@200
4Sunil@1233Khanvilkar" | tr -d '0-9'
omkar@Sunil@Khanvilkar
```

-c option is always used with the -d option because c help use to print the complement of the output given by option -d means it help to delete other character which are not specified in character attributes

```
echo "content" | tr -cd 'characterNotRemove'
```

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ echo "omkar@2004Sunil@1233Khanvilkar" | tr -cd '0-9'  
20041233omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ echo "o
```

-s option is used to squeeze the character means if the character repeated more in adjacent fashion then it eliminates all extra characters and put single character there

```
echo "content" | tr -s 'characterToBeSqueeze'
```

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ echo "omkar sunil khanvilkar ." | tr -s ' '  
omkar sunil khanvilkar .
```

- 9) tee command in Linux reads from standard input and writes to both standard output and one or more files.

```
Ls -l | tee file1 file2 ... fileN
```

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ ls -l | tee files.txt  
total 72  
-rw-rw-r-- 1 omkar omkar 0 Jul 20 03:02 ab1.txt  
-rw-rw-r-- 1 omkar omkar 0 Jul 20 03:02 ab.txt  
-rw-rw-r-- 1 omkar omkar 4 Jul 12 13:02 ab.xt  
-rw-rw-r-- 1 omkar omkar 10 Jul 23 21:30 Albhabet.txt  
drwxrwxr-x 4 omkar omkar 4096 Jul 15 17:45 cmd  
-rw-rw-r-- 1 omkar omkar 111 Jul 24 20:04 Friends.txt  
-rw-rw-r-- 1 omkar omkar 0 Jul 18 18:58 linkab  
-rw-rw-r-- 1 omkar omkar 33 Jul 24 00:43 max.txt  
-rw-rw-r-- 1 omkar omkar 118 Jul 20 00:59 myName.txt  
-rw-rw-r-- 1 omkar omkar 29 Jul 24 01:52 Num1.txt  
-rw-rw-r-- 1 omkar omkar 10 Jul 23 21:30 num.txt  
-rw-rw-r-- 1 omkar omkar 8 Jul 24 20:57 output.txt  
-rw-rw-r-- 1 omkar omkar 38 Jul 23 20:37 StudentData.txt  
-rw-rw-r-- 1 omkar omkar 12 Jul 23 21:39 studentDiv.txt  
-rw-rw-r-- 1 omkar omkar 17 Jul 24 00:39 studentName.txt  
-rw-rw-r-- 1 omkar omkar 9 Jul 23 21:39 studentRoll.txt  
drwxrwxr-x 5 omkar omkar 4096 Jul 10 23:56 summer  
drwxrwxr-x 5 omkar omkar 4096 Jul 11 18:39 summerCP  
-rw-rw-r-- 1 omkar omkar 67 Jul 23 21:14 ter.txt  
-rw-rw-r-- 1 omkar omkar 25 Jul 24 00:20 test.txt  
drwxrwxr-x 3 omkar omkar 4096 Jul 11 00:00 winter
```

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ cat files.txt
total 72
-rw-rw-r-- 1 omkar omkar    0 Jul 20 03:02 ab1.txt
-rw-rw-r-- 1 omkar omkar    0 Jul 20 03:02 ab.txt
-rw-rw-r-- 1 omkar omkar    4 Jul 12 13:02 ab.xt
-rw-rw-r-- 1 omkar omkar   10 Jul 23 21:30 Alphabet.txt
drwxrwxr-x 4 omkar omkar 4096 Jul 15 17:45 cmd
-rw-rw-r-- 1 omkar omkar  111 Jul 24 20:04 Friends.txt
-rw-rw-r-- 1 omkar omkar     0 Jul 18 18:58 linkab
-rw-rw-r-- 1 omkar omkar   33 Jul 24 00:43 max.txt
-rw-rw-r-- 1 omkar omkar 118 Jul 20 00:59 myName.txt
-rw-rw-r-- 1 omkar omkar   29 Jul 24 01:52 Num1.txt
-rw-rw-r-- 1 omkar omkar   10 Jul 23 21:30 num.txt
-rw-rw-r-- 1 omkar omkar     8 Jul 24 20:57 output.txt
-rw-rw-r-- 1 omkar omkar   38 Jul 23 20:37 StudentData.txt
-rw-rw-r-- 1 omkar omkar   12 Jul 23 21:39 studentDiv.txt
-rw-rw-r-- 1 omkar omkar   17 Jul 24 00:39 studentName.txt
-rw-rw-r-- 1 omkar omkar     9 Jul 23 21:39 studentRoll.txt
drwxrwxr-x 5 omkar omkar 4096 Jul 10 23:56 summer
drwxrwxr-x 5 omkar omkar 4096 Jul 11 18:39 summerCP
-rw-rw-r-- 1 omkar omkar    67 Jul 23 21:14 ter.txt
-rw-rw-r-- 1 omkar omkar    25 Jul 24 00:20 test.txt
drwxrwxr-x 3 omkar omkar 4096 Jul 11 00:00 winter
```

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ echo -e "omkar\nsoham\nraju\nNandana\nSunil" | tee MyFriends1.txt MyFriends2.txt
omkar
soham
raju
Nandana
Sunil
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ cat MyFriends1.txt MyFriends2.txt
omkar
soham
raju
Nandana
Sunil
omkar
soham
raju
Nandana
Sunil
```

-a option is use to append the content to the existed files

```
Ls -l | tee -a file1 file2 ... fileN
```

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ echo "rahul" |  
tee -a MyFriends2.txt  
rahul  
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse$ cat MyFriends2.  
txt  
omkar  
soham  
raju  
Nandana  
Sunil  
rahul
```

File Permission related commands:

Changing permission relative manner

Category	Operation	Permission
u <- user	+ assign	r <- read
g <- group	- removal	w <- write
o <- other / a-all	= assign abs permission	x <- execute

chmod command is use to change the mode of the file for different types of entities like user/group/others we use it for absolute as well as for relative mode changing

```
chmod category operation permission fileName
```

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse/winter$ ls -l  
total 12  
-rw-rw-r-- 1 omkar omkar 0 Jul 10 23:59 f1  
-rwxrw-r-- 1 omkar omkar 118 Jul 20 00:59 myName.txt  
-rw-rw-r-- 1 omkar omkar 29 Jul 24 01:52 Num1.txt  
drwxrwxr-x 2 omkar omkar 4096 Jul 10 23:53 seed  
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse/winter$ chmod g-  
w myName.txt  
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse/winter$ ls -l  
total 12  
-rw-rw-r-- 1 omkar omkar 0 Jul 10 23:59 f1  
-rwxr--r-- 1 omkar omkar 118 Jul 20 00:59 myName.txt  
-rw-rw-r-- 1 omkar omkar 29 Jul 24 01:52 Num1.txt  
drwxrwxr-x 2 omkar omkar 4096 Jul 10 23:53 seed  
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse/winter$
```

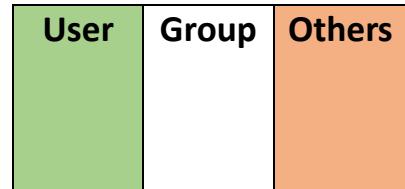
-v verbose command is help us to get to know about what changes actually done in the permissions from which state to which state the given file changes in terms of permissions

chmod -v category operation permission fileName

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse/winter$ ls -l
total 12
-rw-rw-r-- 1 omkar omkar    0 Jul 10 23:59 f1
-rwxr--r-- 1 omkar omkar 118 Jul 20 00:59 myName.txt
-rwxr--r-- 1 omkar omkar   29 Jul 24 01:52 Num1.txt
d-wx----- 2 omkar omkar 4096 Jul 26 01:45 seed
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse/winter$ chmod -v
a+r seed/
mode of 'seed/' changed from 0300 (-wx-----) to 0744 (rwxr--r--)
```

Changing permission absolute manner

Number	Permission notation
0 = 000	____
1 = 001	_x__
2 = 010	_w_
3 = 011	_wx
4 = 100	r__
5 = 101	r_x
6 = 110	rw_
7 = 111	rwx



Proper Positions in command

chmod absolute permission filename

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse/winter/seed$ ls
-l
total 4
--w--w---- 1 omkar omkar 6 Jul 26 01:45 f1.txt
--w--w---- 1 omkar omkar 0 Jul 26 01:36 f2.txt
--w--w---- 1 omkar omkar 0 Jul 26 01:36 f3.txt
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse/winter/seed$ chmod 744 f1.txt
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse/winter/seed$ ls
-l
total 4
-rwxr--r-- 1 omkar omkar 6 Jul 26 01:45 f1.txt
--w--w---- 1 omkar omkar 0 Jul 26 01:36 f2.txt
--w--w---- 1 omkar omkar 0 Jul 26 01:36 f3.txt
```

-R option R is use to recursively change the files permissions which are present inside a directory

chmod -R absolute permission filename

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse/winter$ ls -l seed/
total 4
-rwxr--r-- 1 omkar omkar 6 Jul 26 01:45 f1.txt
--w--w---- 1 omkar omkar 0 Jul 26 01:36 f2.txt
--w--w---- 1 omkar omkar 0 Jul 26 01:36 f3.txt
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse/winter$ chmod -R 777 seed/
omkar@omkar-VMware-Virtual-Platform:~/Desktop/greenHouse/winter$ ls -l seed/
total 4
-rwxrwxrwx 1 omkar omkar 6 Jul 26 01:45 f1.txt
-rwxrwxrwx 1 omkar omkar 0 Jul 26 01:36 f2.txt
-rwxrwxrwx 1 omkar omkar 0 Jul 26 01:36 f3.txt
```

10) chown is use to change the owner for the particular file

Sudo chown -c newOwner filename

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$ ls -l
total 20
-rw-rw-r-- 1 omkar omkar 31 Jul 26 14:47 capital1
-rw-rw-r-- 1 omkar omkar 42 Jul 26 15:09 capital2
-rw-rw-r-- 1 omkar omkar 43 Jul 26 15:15 capital3
-rwx-wx-wx 1 omkar omkar 180 Jul 26 15:01 capitals
-rw-rw-r-- 1 omkar omkar 116 Jul 26 15:17 Mycaps
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$
```

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$ sudo chown -c soham capital2
changed ownership of 'capital2' from omkar to soham
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$
```

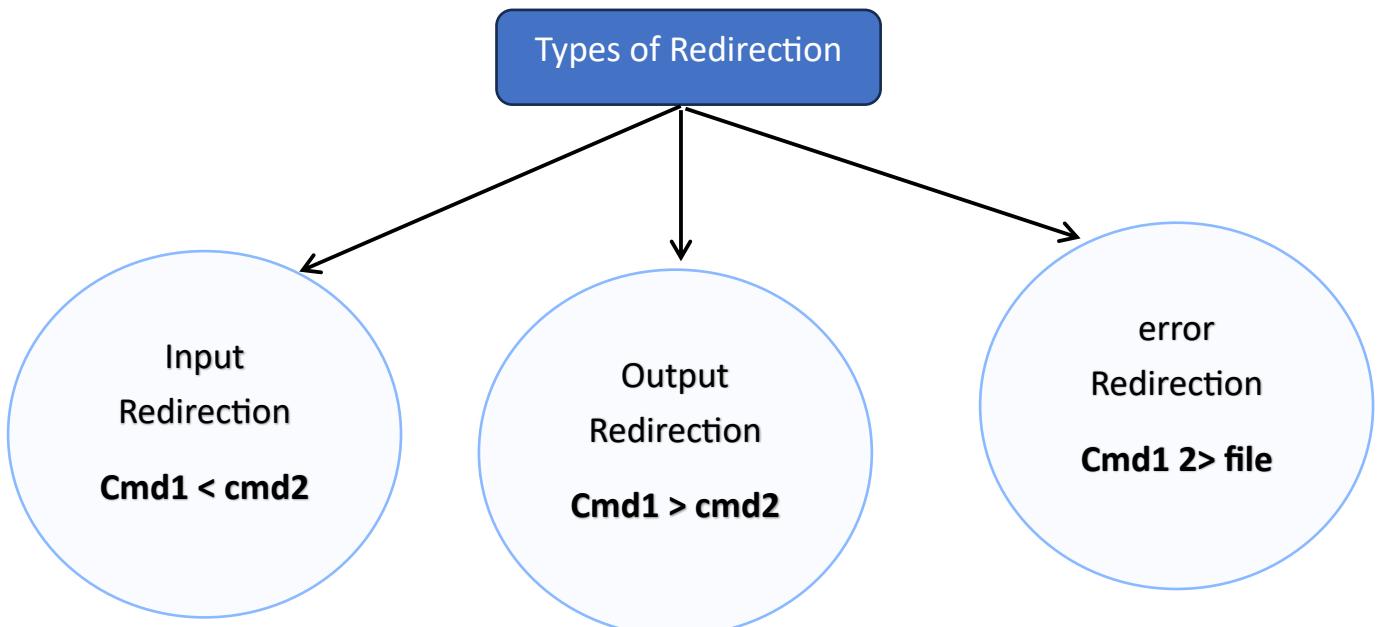
```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$ ls -l capital2
-rw-rw-r-- 1 soham omkar 42 Jul 26 15:09 capital2
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$
```

11) chgrp is use to change the group ownership

Sudo chgrp -c newGroup filename

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Ubuntu-Files/Ubuntu/Assigmn  
ent$ ls -l  
total 20  
-rw-rw-r-- 1 omkar omkar 31 Jul 26 14:47 capital1  
-rw-rw-r-- 1 soham omkar 49 Jul 26 20:36 capital2  
-rw-rw-r-- 1 omkar omkar 43 Jul 26 15:15 capital3  
-rwx-wx-wx 1 omkar omkar 180 Jul 26 15:01 capitals  
-rw-rw-r-- 1 omkar omkar 116 Jul 26 15:17 Mycaps  
  
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Ubuntu-Files/Ubuntu/Assigmn  
ent$ sudo chgrp -c "soham" capital3  
changed group of 'capital3' from omkar to soham  
  
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Ubuntu-Files/Ubuntu/Assigmn  
ent$ ls -l  
total 20  
-rw-rw-r-- 1 omkar omkar 31 Jul 26 14:47 capital1  
-rw-rw-r-- 1 soham omkar 49 Jul 26 20:36 capital2  
-rw-rw-r-- 1 omkar soham 43 Jul 26 15:15 capital3  
-rwx-wx-wx 1 omkar omkar 180 Jul 26 15:01 capitals  
-rw-rw-r-- 1 omkar omkar 116 Jul 26 15:17 Mycaps
```

Redirection: Provide a powerful command line controls



Standard input redirection:

```
cat < fileName
```

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Ubuntu-Files/Ubuntu/Assignment$ cat < capital2
Dispur
Chandigarh
Shimla
Mumbai
Bengaluru
Bhopal
```

Standard output redirection:

```
cat filename > NewFile
```

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Ubuntu-Files/Ubuntu/greenHouse$ cat Friends.txt
Omkar
Soham
Ajay
Raju
Sarthak
Pankaj
Rahul
Omi
Raju
Amit
Tapu
Omkar
Raju
Raju
Sarthak
Omkar
Sarthak
Omkar
Ajay
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Ubuntu-Files/Ubuntu/greenHouse$ cat Friends.txt > newFriend.txt
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Ubuntu-Files/Ubuntu/greenHouse$ cat newFriend.txt
Omkar
Soham
Ajay
Raju
Sarthak
Pankaj
Rahul
Omi
Raju
Amit
Tapu
Omkar
Raju
Raju
Sarthak
Omkar
Sarthak
Omkar
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Ubuntu-Files/Ubuntu/greenHouse$
```

Standard error redirection:

```
cat filename 2> NewFile
```

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Ubuntu-Files/Ubuntu/greenHouse$ cat omkar.txt 2> error.txt
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Ubuntu-Files/Ubuntu/greenHouse$ cat error.txt
cat: omkar.txt: No such file or directory
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Ubuntu-Files/Ubuntu/greenHouse$
```

12) grep command is used to find out the strings or set of strings in the particular file or directory which are in particular pattern given in REGEX form

grep "REGEX" fileName

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Ubuntu-Files/Ubuntu/greenHouse$ grep "om  
*" MyFriends2.txt  
omkar  
soham
```

-i ignore case for matching

grep -i "REGEX" fileName

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Ubuntu-Files/Ubuntu/greenHouse$ grep -i  
"OMKAR" newFriend.txt  
Omkar  
Omkar  
Omkar  
Omkar  
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Ubuntu-Files/Ubuntu/greenHouse$ █
```

-v doesn't display lines matching expression

grep -v "REGEX" fileName

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Ubuntu-Files/Ubuntu/greenHouse$ cat Stud  
entData.txt  
Omkar,10,ITB  
Soham,12,ITC  
Ajay,17,ITC  
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Ubuntu-Files/Ubuntu/greenHouse$ grep -iv  
"omkar" StudentData.txt  
Soham,12,ITC  
Ajay,17,ITC  
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Ubuntu-Files/Ubuntu/greenHouse$ █
```

-n display line numbers along of occurrences

grep -n "REGEX" fileName

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Ubuntu-Files/Ubuntu/greenHouse$ grep -in  
"omkar" StudentData.txt  
1:Omkar,10,ITB  
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Ubuntu-Files/Ubuntu/greenHouse$ grep -in  
"Ajay" StudentData.txt  
3:Ajay,17,ITC  
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Ubuntu-Files/Ubuntu/greenHouse$ █
```

-c counting number of occurrences

```
grep -c "REGEX" fileName
```

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Ubuntu-Files/Ubuntu/greenHouse$ grep -ic
"Omkar" newFriend.txt
4
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Ubuntu-Files/Ubuntu/greenHouse$ grep -i
"Omkar" newFriend.txt
Omkar
Omkar
Omkar
Omkar
Omkar
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Ubuntu-Files/Ubuntu/greenHouse$
```

-e exp for matching

```
grep -e "REGEX1" -e "REGEX2" fileName
```

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Ubuntu-Files/Ubuntu/greenHouse$ grep -ie
"om*" -ie "Ra*" newFriend.txt
Omkar
Soham
Raju
Sarthak
Rahul
Omi
Raju
Omkar
Raju
Raju
Sarthak
Omkar
Sarthak
Omkar
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Ubuntu-Files/Ubuntu/greenHouse$
```

-f file take patterns from file

```
grep -f pattern.txt fileName
```

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Ubuntu-Files/Ubuntu/greenHouse$ cat MyFr
iends2.txt
omkar
soham
raju
Nandana
Sunil
rahul
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Ubuntu-Files/Ubuntu/greenHouse$ grep -if
MyFriends2.txt newFriend.txt
Omkar
Soham
Raju
Rahul
Raju
Omkar
Raju
Raju
Omkar
Omkar
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Ubuntu-Files/Ubuntu/greenHouse$
```

-E treat pattern as an extended reg. exp

grep -E fileName

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Ubuntu-Files/Ubuntu/greenHouse$ cat StudentData.txt
Omkar,10,ITB
Soham,12,ITC
Ajay,17,ITC
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Ubuntu-Files/Ubuntu/greenHouse$ grep -E "Omkar|ITC" StudentData.txt
Omkar,10,ITB
Soham,12,ITC
Ajay,17,ITC
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Ubuntu-Files/Ubuntu/greenHouse$ grep -E "Omkar|ITB" StudentData.txt
Omkar,10,ITB
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Ubuntu-Files/Ubuntu/greenHouse$
```

-F matches multiple fixed strings (fgrep)

grep -F "fixed_String" fileName

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Ubuntu-Files/Ubuntu/greenHouse$ grep -iF "soham" newFriend.txt
Soham
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Ubuntu-Files/Ubuntu/greenHouse$
```

Assignment 2 Questions

1. Change your password to a password you would like to use for the remainder of the semester.

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop$ passwd
Changing password for omkar.
Current password:
New password:
BAD PASSWORD: The password contains the user name in some form
New password:
Retype new password:
passwd: password updated successfully
```

2. Display the system's date.

```
omkar@omkar-VMware-Virtual-Platform:~$ date +"%d-%m-%Y"
27-07-2024
omkar@omkar-VMware-Virtual-Platform:~$
```

3. Count the number of lines in the /etc/passwd file.

```
omkar@omkar-VMware-Virtual-Platform:/etc/pam.d$ wc -l passwd
6 passwd
```

4. Find out who else is on the system.

```
omkar@omkar-VMware-Virtual-Platform:/etc/pam.d$ who
omkar    seat0          2024-07-26 14:21 (login screen)
omkar    tty2          2024-07-26 14:21 (tty2)
```

5. Direct the output of the man pages for the date command to a file named *mydate*.

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$ man date > mydate
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$ cat mydate
DATE(1)                               User Commands               DATE(1)

NAME
      date - print or set the system date and time

SYNOPSIS
      date [OPTION]... [+FORMAT]
      date [-u|--utc|--universal] [MMDDhhmm[[CC]YY][.ss]]

DESCRIPTION
      Display date and time in the given FORMAT. With -s, or with
      [MMDDhhmm[[CC]YY][.ss]], set the date and time.

      Mandatory arguments to long options are mandatory for short op-
      tions too.
```

```

omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignments$ cat mydate
DATE(1)                               User Commands           DATE(1)

NAME
    date - print or set the system date and time

SYNOPSIS
    date [OPTION]... [+FORMAT]
    date [-u|--utc|--universal] [MMDDhhmm[[CC]YY][.ss]]

DESCRIPTION
    Display date and time in the given FORMAT. With -s, or with
    [MMDDhhmm[[CC]YY][.ss]], set the date and time.

    Mandatory arguments to long options are mandatory for short op-
    tions too.

    -d, --date=STRING
        display time described by STRING, not 'now'

    --debug
        annotate the parsed date, and warn about questionable us-
        age to stderr

    -f, --file=DATEFILE
        like --date; once for each line of DATEFILE

    -I[FMT], --iso-8601[=FMT]
        output date/time in ISO 8601 format. FMT='date' for date
        only (the default), 'hours', 'minutes', 'seconds', or
        'ns' for date and time to the indicated precision. Exam-
        ple: 2006-08-14T02:34:56.060

    --resolution
        output the available resolution of timestamps Example:
        0.000000001

    -R, --rfc-email
        output date and time in RFC 5322 format. Example: Mon,
        14 Aug 2006 02:34:56 -0600

    --rfc-3339=FMT
        output date/time in RFC 3339 format. FMT='date', 'sec-
        onds', or 'ns' for date and time to the indicated preci-
        sion. Example: 2006-08-14 02:34:56.060

    -r, --reference=FILE
        display the last modification time of FILE

    -s, --set=STRING
        set time described by STRING

    -u, --utc, --universal
        print or set Coordinated Universal Time (UTC)

    --help display this help and exit

    --version
        output version information and exit

All options that specify the date to display are mutually exclu-
sive. I.e.: --date, --file, --reference, --resolution.

FORMAT controls the output. Interpreted sequences are:

%   a literal %

%a  locale's abbreviated weekday name (e.g., Sun)
%A  locale's full weekday name (e.g., Sunday)
%b  locale's abbreviated month name (e.g., Jan)
%B  locale's full month name (e.g., January)
%c  locale's date and time (e.g., Thu Mar  3 23:05:25 2005)
%C  century; like %Y, except omit last two digits (e.g., 20)
%d  day of month (e.g., 01)
%D  date; same as %m/%d/%y
%e  day of month, space padded; same as %_d
%F  full date; like %+Y-%m-%d
%g  last two digits of year of ISO week number (see %G)
%G  year of ISO week number (see %V); normally useful only
    with %V
%h  same as %b
%H  hour (00..23)

```

```

%V    ISO week number, with Monday as first day of week
      (01..53)
%w    day of week (0..6); 0 is Sunday
%W    week number of year, with Monday as first day of week
      (00..53)
%x    locale's date representation (e.g., 12/31/99)
%X    locale's time representation (e.g., 23:13:48)
%y    last two digits of year (00..99)
%Y    year
%z    +hhmm numeric time zone (e.g., -0400)
%:z    +hh:mm numeric time zone (e.g., -04:00)
%::z    +hh:mm:ss numeric time zone (e.g., -04:00:00)
%:::z  numeric time zone with : to necessary precision (e.g.,
      -04, +05:30)
%Z    alphabetic time zone abbreviation (e.g., EDT)

By default, date pads numeric fields with zeroes. The following
optional flags may follow '%':
-    (hyphen) do not pad the field
_    (underscore) pad with spaces
0    (zero) pad with zeros
+    pad with zeros, and put '+' before future years with >4
      digits
^    use upper case if possible
#    use opposite case if possible

After any flags comes an optional field width, as a decimal num-
ber; then an optional modifier, which is either E to use the lo-
cale's alternate representations if available, or 0 to use the
locale's alternate numeric symbols if available.

EXAMPLES
Convert seconds since the Epoch (1970-01-01 UTC) to a date
$ date --date=@2147483647

Show the time on the west coast of the US (use tzselect(1) to
find TZ)
$ TZ='America/Los_Angeles' date

Show the local time for 9AM next Friday on the west coast of the
US
$ date --date='TZ="America/Los_Angeles" 09:00 next Fri'

DATE STRING
The --date=STRING is a mostly free format human readable date
string such as "Sun, 29 Feb 2004 16:21:42 -0800" or "2004-02-29
16:21:42" or even "next Thursday". A date string may contain
items indicating calendar date, time of day, time zone, day of
week, relative time, relative date, and numbers. An empty
string indicates the beginning of the day. The date string for-
mat is more complex than is easily documented here but is fully
described in the info documentation.

AUTHOR
Written by David MacKenzie.

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

COPYRIGHT
Copyright © 2023 Free Software Foundation, Inc. License GPLv3+:
GNU GPL version 3 or later <https://gnu.org/licenses/gpl.html>.
This is free software: you are free to change and redistribute
it. There is NO WARRANTY, to the extent permitted by law.

SEE ALSO
Full documentation <https://www.gnu.org/software/coreutils/date>
or available locally via: info '(coreutils) date invocation'

GNU coreutils 9.4           April 2024          DATE(1)

```

6. Create a subdirectory called *mydir*.

```

omkar@omkar-Virtual-Platform:~/Desktop/Assignment$ ls
mydate
omkar@omkar-Virtual-Platform:~/Desktop/Assignment$ mkdir mydir
omkar@omkar-Virtual-Platform:~/Desktop/Assignment$ ls
mydate  mydir
omkar@omkar-Virtual-Platform:~/Desktop/Assignment$
```

7. Move the file *mydate* into the new subdirectory.

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$ ls mydir
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$ mv mydate mydir/
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$ ls
mydir
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$ ls mydir/
mydate
```

8. Go to the subdirectory *mydir* and copy the file *mydate* to a new file called *ourdate*

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$ cd mydir/
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment/mydir$ cp mydate ourdate
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment/mydir$ ls
mydate ourdate
```

9. List the contents of *mydir*.

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment/mydir$ ls
mydate ourdate
```

10. Do a long listing on the file *ourdate* and note the permissions.

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment/mydir$ ls -l ourdate
-rw-rw-r-- 1 omkar omkar 6969 Jul 26 14:35 ourdate
```

Permissions : user => RW | group=> RW | others=> R

11. Display the name of the current directory starting from the root.

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment/mydir$ pwd
/home/omkar/Desktop/Assignment/mydir
```

12. Move the files in the directory *mydir* back to the HOME directory.

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment/mydir$ mv mydate ../../../../../../
omkar/
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment/mydir$ mv ourdate ../../../../../../
/omkar/
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment/mydir$ ls ../../../../../../omkar/
Desktop    Downloads   mydate   Pictures   snap      Videos
Documents  Music      ourdate  Public     Templates
```

13. List all the files in your HOME directory.

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment/mydir$ cd ~
omkar@omkar-VMware-Virtual-Platform:~$ ls
Desktop    Downloads   mydate   Pictures   snap      Videos
Documents  Music      ourdate  Public     Templates
```

14. Display the first 5 lines of *mydate*.

```
omkar@omkar-VMware-Virtual-Platform:~$ head -5 mydate
DATE(1)                               User Commands                  DATE(1)

NAME
      date - print or set the system date and time
```

15. Display the last 8 lines of *mydate*.

```
omkar@omkar-VMware-Virtual-Platform:~$ tail -8 mydate
      This is free software: you are free to change and redistribute
      it. There is NO WARRANTY, to the extent permitted by law.
```

SEE ALSO

Full documentation <<https://www.gnu.org/software/coreutils/date>>
or available locally via: info '(coreutils) date invocation'

GNU coreutils 9.4

April 2024

DATE(1)

16. Remove the directory *mydir*.

```
omkar@omkar-VMware-Virtual-Platform:~$ cd Desktop/Assignment/
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$ ls
mydir
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$ rmdir mydir/
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$ ls
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$ █
```

17. Redirect the output of the long listing of files to a file named *list*.

```
omkar@omkar-VMware-Virtual-Platform:~$ ls -l > list
omkar@omkar-VMware-Virtual-Platform:~$ cat list
total 52
drwxr-xr-x 5 omkar omkar 4096 Jul 26 14:29 Desktop
drwxr-xr-x 2 omkar omkar 4096 Jul 10 02:45 Documents
drwxr-xr-x 3 omkar omkar 4096 Jul 12 13:54 Downloads
-rw-rw-r-- 1 omkar omkar 0 Jul 26 14:44 list
drwxr-xr-x 2 omkar omkar 4096 Jul 10 02:45 Music
-rw-rw-r-- 1 omkar omkar 6969 Jul 26 14:29 mydate
-rw-rw-r-- 1 omkar omkar 6969 Jul 26 14:35 ourdate
drwxr-xr-x 4 omkar omkar 4096 Jul 10 15:06 Pictures
drwxr-xr-x 2 omkar omkar 4096 Jul 10 02:45 Public
drwx----- 7 omkar omkar 4096 Jul 10 13:59 snap
drwxr-xr-x 2 omkar omkar 4096 Jul 10 02:45 Templates
drwxr-xr-x 3 omkar omkar 4096 Jul 10 02:46 Videos
omkar@omkar-VMware-Virtual-Platform:~$ █
```

18. Select any 5 capitals of states in India and enter them in a file named *capitals1*.

Choose 5 more capitals and enter them in a file named *capitals2*. Choose 5 more capitals and enter them in a file named *capitals3*. Concatenate all 3 files and redirect the output to a file named *capitals*.

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$ echo -e "Mumbai\nPanji\nGan  
dhinagar\nPatna" > capital1  
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$ echo -e "Dispur\nChandigarh  
nShimla\nBengaluru\nBhopal" > capital2  
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$ echo -e "Jaipur\nGangtok\nHy  
derabad\nLucknow\nKolkata" > capital3  
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$ ls  
capital1 capital2 capital3  
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$ cat capital1 capital2 capita  
l3  
Mumbai  
Panji  
Gandhinagar  
Patna  
Dispur  
Chandigarh  
Shimla  
Bengaluru  
Bhopal  
Jaipur  
Gangtok  
Hyderabad  
Lucknow  
Kolkata  
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$ cat capital1 capital2 capita  
l3 > capitals  
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$ cat capitals  
Mumbai  
Panji  
Gandhinagar  
Patna  
Dispur  
Chandigarh  
Shimla  
Bengaluru  
Bhopal  
Jaipur  
Gangtok  
Hyderabad  
Lucknow  
Kolkata  
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$ █
```

19. Concatenate the file *capitals2* at the end of file *capitals*.

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$ cat capital2
Dispur
Chandigarh
Shimla
Bengaluru
Bhopal
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$ cat capital2 >> capitals
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$ cat capitals
Mumbai
Panji
Gandhinagar
Patna
Dispur
Chandigarh
Shimla
Bengaluru
Bhopal
Jaipur
Gangtok
Hyderabad
Lucknow
Kolkata
Dispur
Chandigarh
Shimla
Bengaluru
Bhopal
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$
```

20. Redirect the file *capitals* as an input to the command “wc -l”.

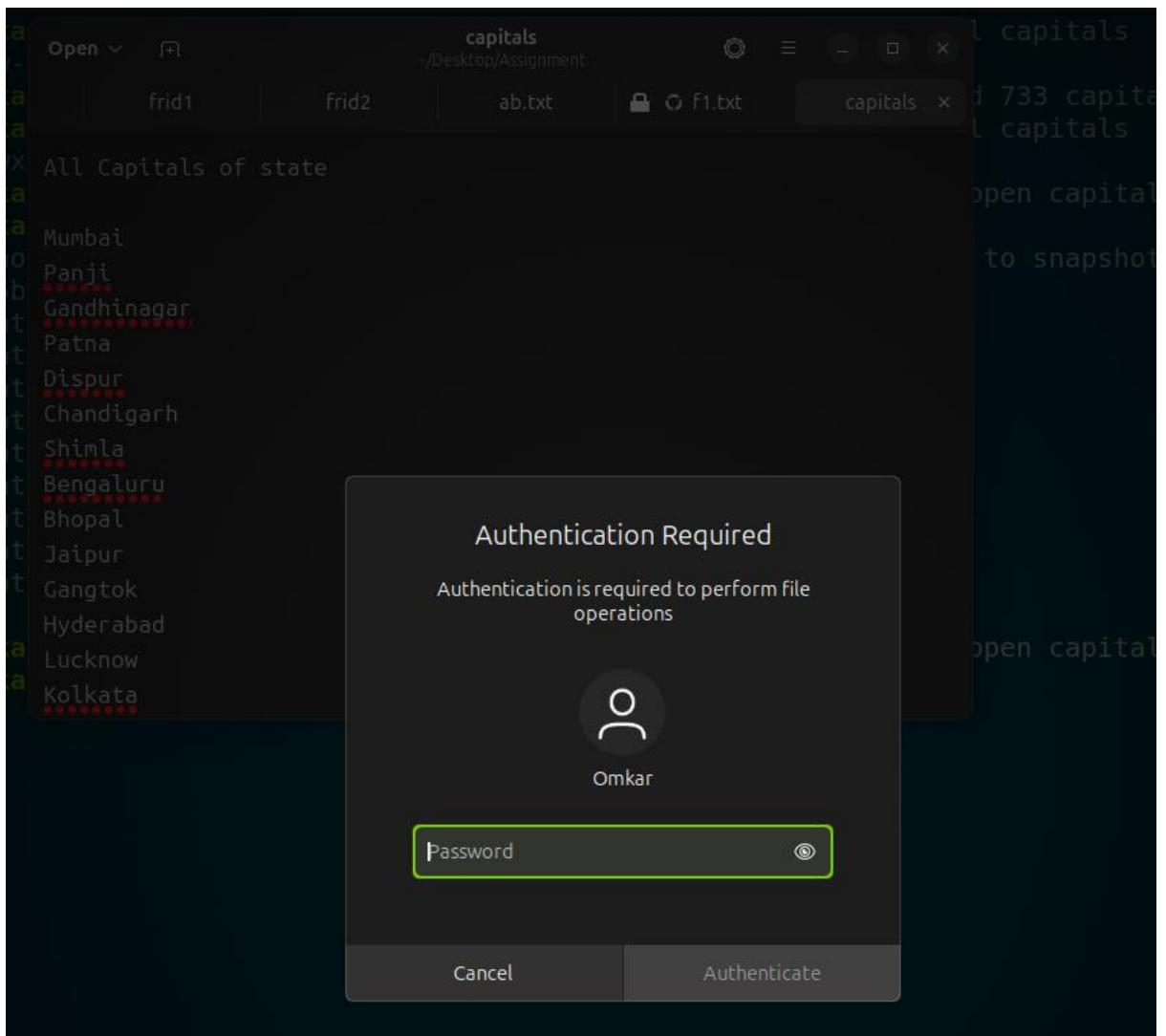
```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$ cat capitals | wc -l
19
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$
```

21. Give read and write permissions to all users for the file *capitals*.

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$ ls -l capitals
-rw-rw-r-- 1 omkar omkar 156 Jul 26 14:53 capitals
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$ chmod a+r,a+w capitals
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$ ls -l capitals
-rw-rw-rw- 1 omkar omkar 156 Jul 26 14:53 capitals
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$
```

22. Give read permissions only to the owner of the file *capitals*. Open the file, make some changes and try to save it. What happens ?

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$ ls -l capitals
-rw-rw-rw- 1 omkar omkar 156 Jul 26 14:53 capitals
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$ chmod 733 capitals
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$ ls -l capitals
-rwx-wx-wx 1 omkar omkar 156 Jul 26 14:53 capitals
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$ xdg-open capitals
```



Asking about the user password for authentication of owner

23. Create an alias to concatenate the 3 files *capitals1*, *capitals2*, *capitals3* and redirect the output to a file named *capitals*. Activate the alias and make it run.

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$ alias MergeCaps='cat capital1 capital2 capital3'  
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$ MergeCaps  
Mumbai  
Panji  
Gandhinagar  
Patna  
Dispur  
Chandigarh  
Shimla  
Bengaluru  
Bhopal  
Itanagar  
New Delhi  
Port Blair  
Jammu  
Kargil
```

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$ ls
capital1 capital2 capital3 capitals Mycaps
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$ MergeCaps > Mycaps
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$ cat Mycaps
Mumbai
Panji
Gandhinagar
Patna
Dispur
Chandigarh
Shimla
Bengaluru
Bhopal
Itanagar
New Delhi
Port Blair
Jammu
Kargil
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$
```

24. What are the environment variables PATH, HOME and TERM set to on your terminal ?

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/
usr/local/games:/snap/bin:/snap/bin
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$ echo $HOME
/home/omkar
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$ echo $TERM
xterm-256color
```

25. Find out the number of times the string “the” appears in the file *mydate*.

```
omkar@omkar-VMware-Virtual-Platform:~/Desktop/Assignment$ cd ~
omkar@omkar-VMware-Virtual-Platform:~$ grep -c 'the' mydate
22
```

26. Find out the line numbers on which the string “date” exists in *mydate*.

```
omkar@omkar-VMware-Virtual-Platform:~$ grep -in "date" mydate
1:DATE(1)                               User Commands                      DATE(1)
4:      date - print or set the system date and time
7:      date [OPTION]... [+FORMAT]
8:      date [-u|--utc|--universal] [MMDDhhmm[[CC]YY][.ss]]
11:     Display date and time in the given FORMAT. With -s, or with
12:     [MMDDhhmm[[CC]YY][.ss]], set the date and time.
17:     -d, --date=STRING
21:         annotate the parsed date, and warn about questionable us-
24:     -f, --file=DATEFILE
25:         like --date; once for each line of DATEFILE
28:         output date/time in ISO 8601 format. FMT='date' for date
30:         'ns' for date and time to the indicated precision. Exam-
38:         output date and time in RFC 5322 format. Example: Mon,
42:         output date/time in RFC 3339 format. FMT='date', 'sec-
43:         onds', or 'ns' for date and time to the indicated preci-
60:         All options that specify the date to display are mutually exclu-
61:         sive. I.e.: --date, --file, --reference, --resolution.
75:         %c      locale's date and time (e.g., Thu Mar 3 23:05:25 2005)
81:         %D      date; same as %m/%d/%y
85:         %F      full date; like %+4Y-%m-%d
144:        %x      locale's date representation (e.g., 12/31/99)
163: By default, date pads numeric fields with zeroes. The following
185: Convert seconds since the Epoch (1970-01-01 UTC) to a date
187:     $ date --date='@2147483647'
192:     $ TZ='America/Los_Angeles' date
197:     $ date --date='TZ="America/Los_Angeles" 09:00 next Fri'
199:DATE STRING
200:    The --date=STRING is a mostly free format human readable date
202:    16:21:42" or even "next Thursday". A date string may contain
203:    items indicating calendar date, time of day, time zone, day of
204:    week, relative time, relative date, and numbers. An empty
205:    string indicates the beginning of the day. The date string for-
225:    Full documentation <https://www.gnu.org/software/coreutils/date>
226:    or available locally via: info '(coreutils) date invocation'
228:GNU coreutils 9.4                         April 2024                      DATE(1)
```

27. Print all lines of *mydate* except those that have the letter “i” in them.

```
omkar@omkar-VMware-Virtual-Platform:~$ grep -v "i" mydate
DATE(1)                               User Commands           DATE(1)

NAME

SYNOPSIS
    date [OPTION]... [+FORMAT]

DESCRIPTION

    -d, --date=STRING
    --debug
        age to stderr

    ple: 2006-08-14T02:34:56-06:00
    0.000000001
    14 Aug 2006 02:34:56 -0600
    --rfc-3339=FMT
    -r, --reference=FILE
    -s, --set=STRING

FORMAT controls the output. Interpreted sequences are:

%A      locale's full weekday name (e.g., Sunday)
%B      locale's full month name (e.g., January)

%d      day of month (e.g., 01)
%D      date; same as %m/%d/%y
%e      day of month, space padded; same as %_d

%G      year of ISO week number (see %V); normally useful only
```

```
%I      hour (01..12)
%j      day of year (001..366)
%k      hour, space padded ( 0..23); same as %_H
%l      hour, space padded ( 1..12); same as %_I
%m      month (01..12)

%N      nanoseconds (000000000..999999999)
known

%q      quarter of year (1..4)

%S      second (00..60)
%t      a tab

(00..53)
(01..53)

(00..53)

%Y      year

-04, +05:30)
```

EXAMPLES

```
$ date --date='@2147483647'
```

```
US
```

```
DATE STRING
```

```
AUTHOR
```

```
REPORTING BUGS
  ject.org/team/>
```

```
COPYRIGHT
```

```
SEE ALSO
```

```
omkar@omkar-VMware-Virtual-Platform:~$ █
```

28. Create the file *monotonic* as follows:

^a?b?b?c?.....x?y?z\$

Run the egrep command for *monotonic* against /usr/dict/words and search for all 4 letter words.

```
omkar@omkar-VMware-Virtual-Platform:~$ echo "^(a?b?c?d?e?f?g?h?i?j?k?l?m?n?o?p?q?r?s?t?u?v?w?x?y?z$" > monotonic.txt
omkar@omkar-VMware-Virtual-Platform:~$ grep -Ef monotonic.txt /usr/share/dict/words
adz
chintz
doz
dz
oz
z
omkar@omkar-VMware-Virtual-Platform:~$ grep -Ef monotonic.txt /usr/share/dict/words | grep -E "^.{4}$"
omkar@omkar-VMware-Virtual-Platform:~$
```

29. List 5 states in north east India in a file *mystates*. List their corresponding capitals in a file *mycapitals*. Use the *paste* command to join the 2 files.

```
omkar@omkar-VMware-Virtual-Platform:~$ echo -e "Arunachal pradesh\nAssam\nManipur\nMeghalaya\nMizoram" > myStates
omkar@omkar-VMware-Virtual-Platform:~$ echo -e "Itanagar\nDispur\nImphal\nShilong\nAizawl" > myCapitals
omkar@omkar-VMware-Virtual-Platform:~$ paste myStates myCapitals
Arunachal pradesh      Itanagar
Assam      Dispur
Manipur   Imphal
Meghalaya Shilong
Mizoram Aizawl
omkar@omkar-VMware-Virtual-Platform:~$ paste -d'=' myStates myCapitals
Arunachal pradesh=Itanagar
Assam=Dispur
Manipur=Imphal
Meghalaya=Shilong
Mizoram=Aizawl
omkar@omkar-VMware-Virtual-Platform:~$
```

30. Use the *cut* command to print the 1st and 3rd columns of the /etc/passwd file for all students in this class.

Column 1

```
omkar@omkar-Virtual-Platform:~$ cut -d':' -f1 /etc/passwd
root
daemon
bin
sys
sync
games
man
lp
mail
news
uucp
proxy
www-data
backup
list
irc
_apt
nobody
systemd-network
systemd-timesync
dhcpcd
messagebus
syslog
systemd-resolve
uuid
usbmux
tss
systemd-oom
kernoops
whoopsie
dnsmasq
avahi
tcpdump
sssd
speech-dispatcher
cups-pk-helper
fwupd-refresh
saned
geoclue
cups-browsed
hplip
polkitd
rtkit
colord
gnome-initial-setup
gdm
nm-openvpn
gnome-remote-desktop
omkar
soham
```

Column 3

```
omkar@omkar-VMware-Virtual-Platform:~$ cut -d':' -f3 /etc/passwd
0
1
2
3
4
5
6
7
8
9
10
13
33
34
38
39
42
65534
998
996
100
101
102
991
103
104
105
990
106
107
999
108
109
110
111
112
989
113
114
115
116
988
117
118
119
120
121
987
1000
1001
```

31. Count the number of people logged in and also trap the users in a file using the *tee* command.

```
omkar@omkar-VMware-Virtual-Platform:~$ users | tee loggedInUsers
omkar omkar
omkar@omkar-VMware-Virtual-Platform:~$ cat loggedInUsers
omkar omkar
omkar@omkar-VMware-Virtual-Platform:~$
```