Experiment Number: 02

Problem Statement: Execution of advance Linux commands

NAME: **Omkar Sunil Khanvilkar**  ROLLNO: 7

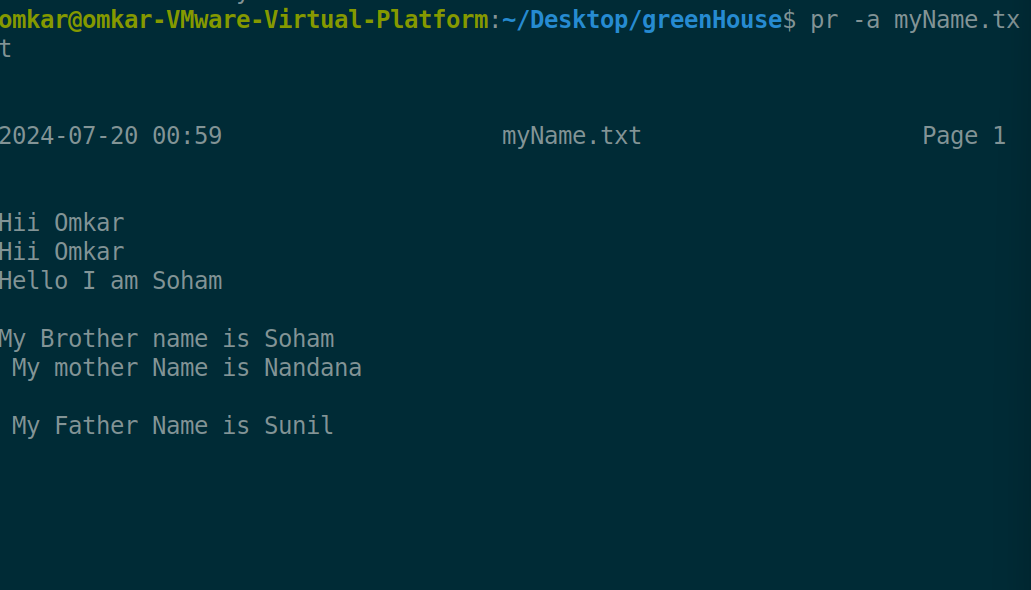
CLASS: IT-B BATCH: 2

DATE OF PERFORMANCE: 26/7/2024

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

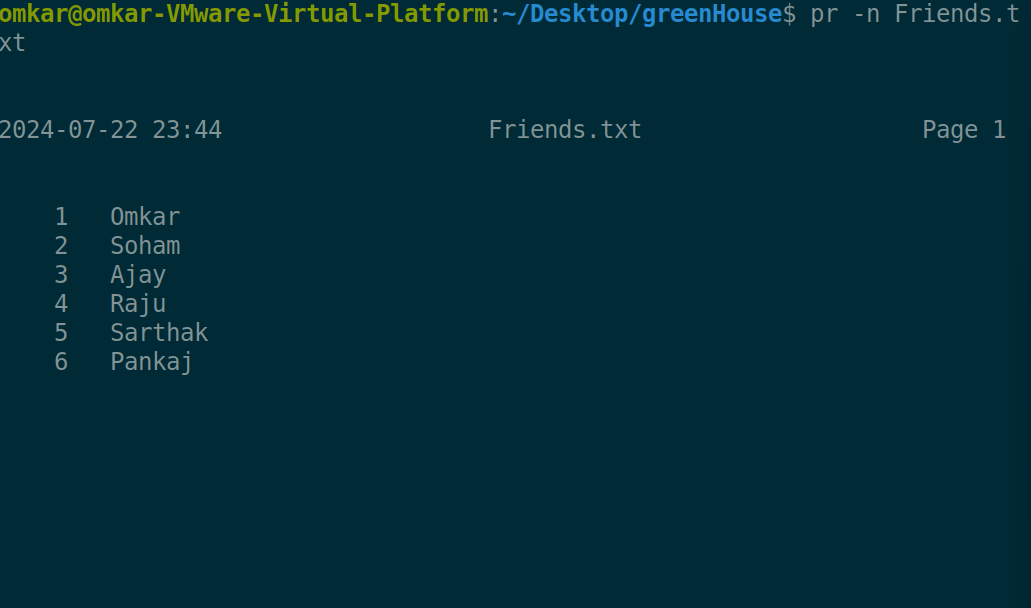
1. pr :pr command is used to prepare a file for printing by adding suitable footers, headers, and the formatted text. **pr** command actually adds 5 lines of margin both at the top and bottom of the page.  The header part shows the date and time of the last modification of the file with the file name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

**pr fileName**



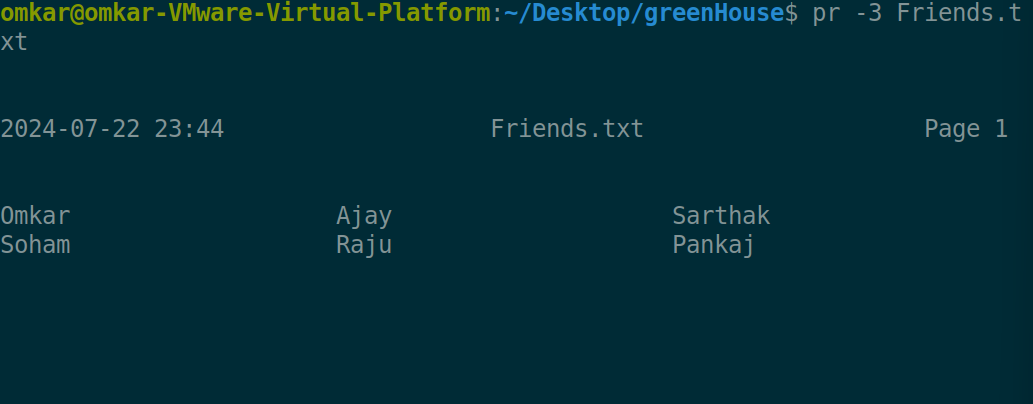
**-n**To provide number lines which helps in debugging the code -n option is used.

**pr -n fileName**



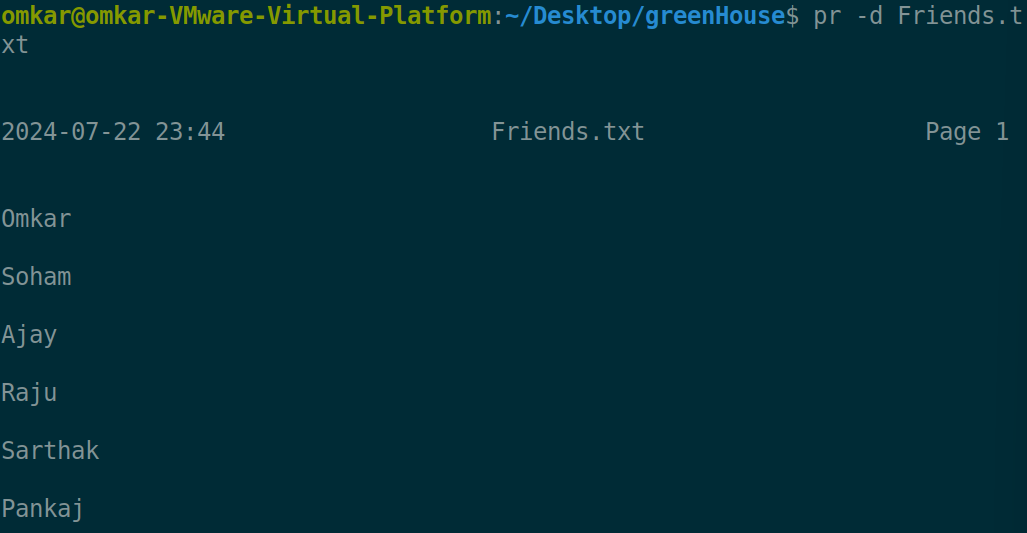
**-k** where k is any integer for example if we want to print this content in 3 columns use -3

**pr -3 fileName**



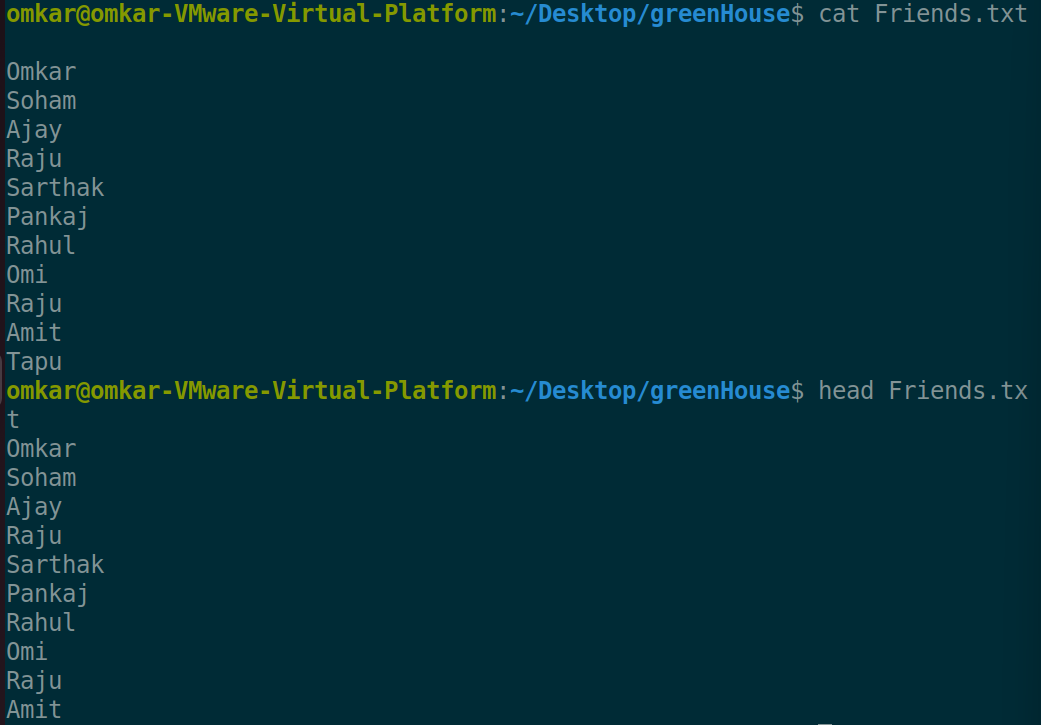
**-d** To Double the paces input, reduces clutter -d option is used.

**pr -d fileName**



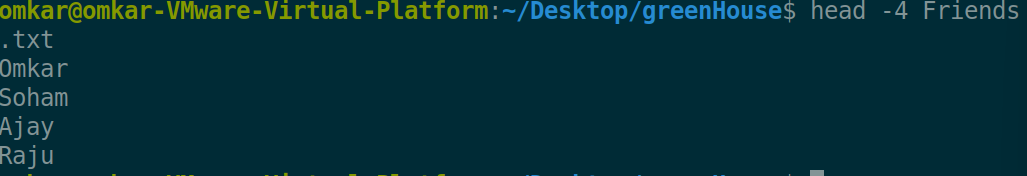
1. head Command is use to Display first 10 lines of file

**head fileName**



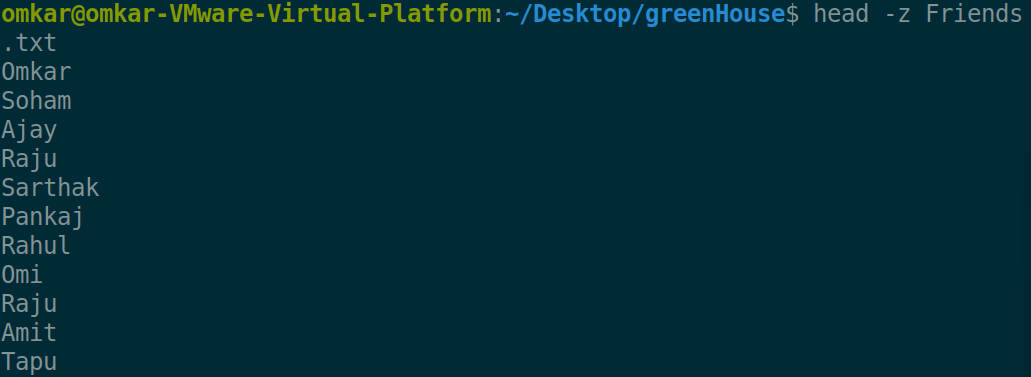
-k option use to print he first k lines of the file

**Head -k fileName**



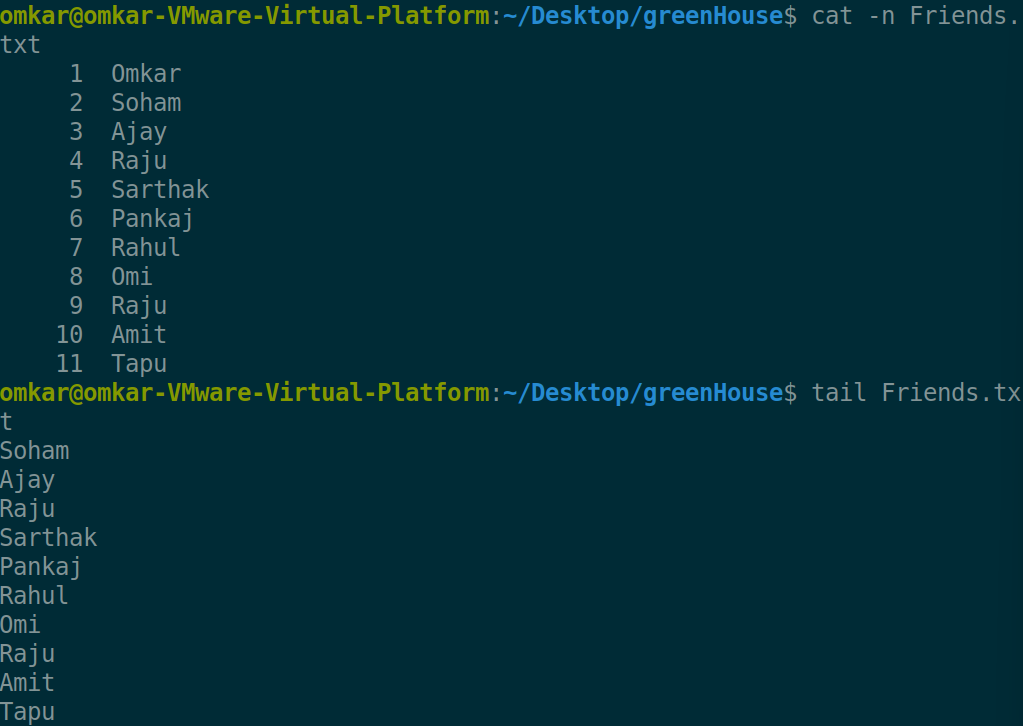
-z option line delimiter is NUL, not newline

**Head -z fileName**



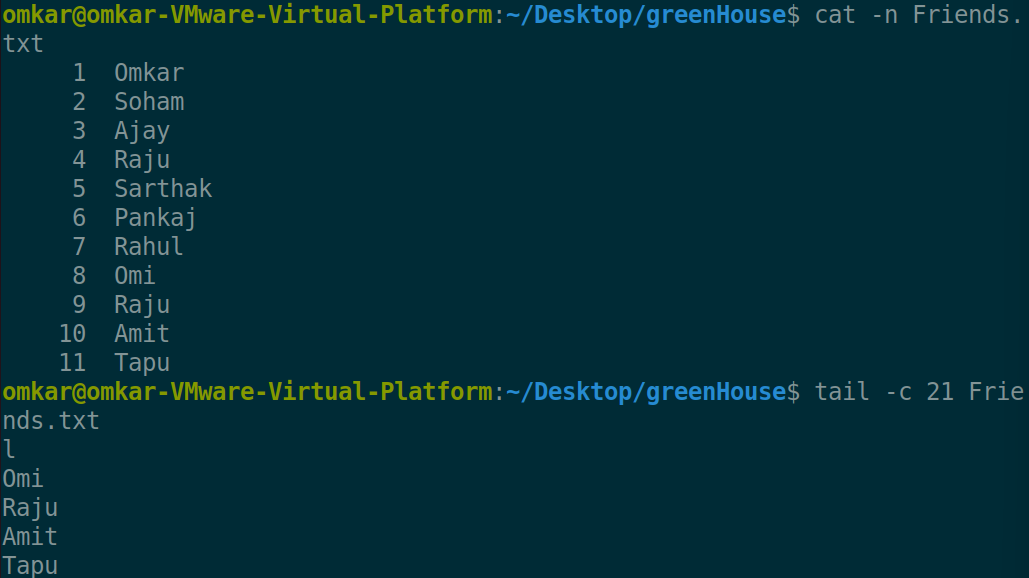
1. tail : tail command is use to print last 10 lines of files

**tail fileName**



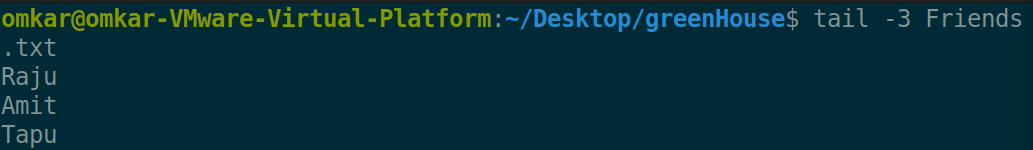
-c option c is use to get last n bytes of the file where n is any integer

**tail -c 100 fileName**



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

**tail -10 fileName**



1. 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

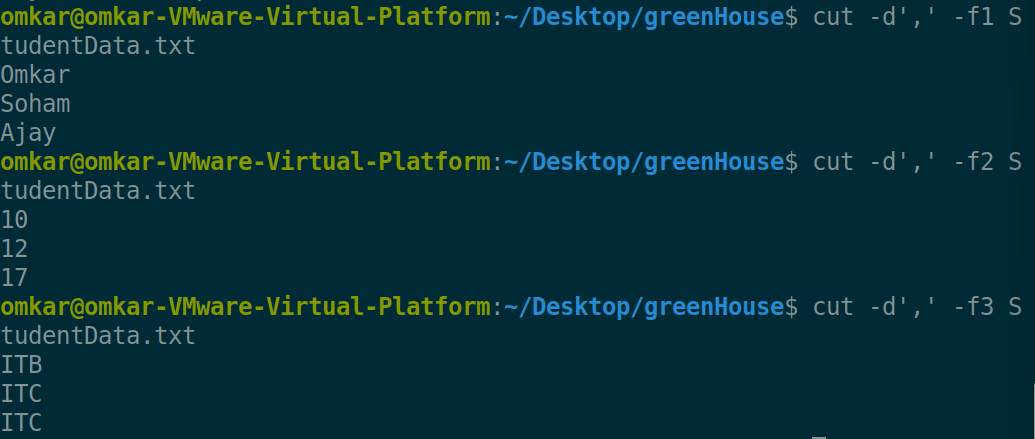
**cut -b range fileName**



-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**





-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



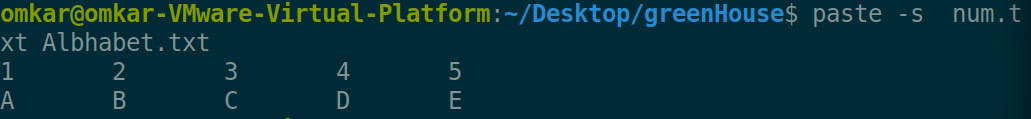
1. paste command in Linux is used to merge lines of files side by side

**paste file1 file2**



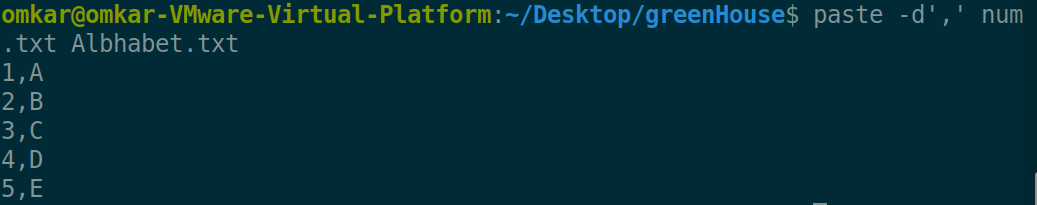
-s option to paste files serially

**Paste -s file1 file2**



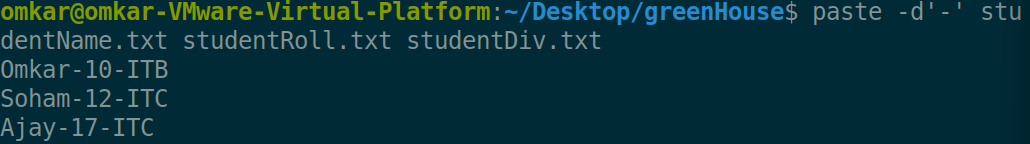
-d To use a comma as a delimiter:

**Paste -d’,’ file1 file2**



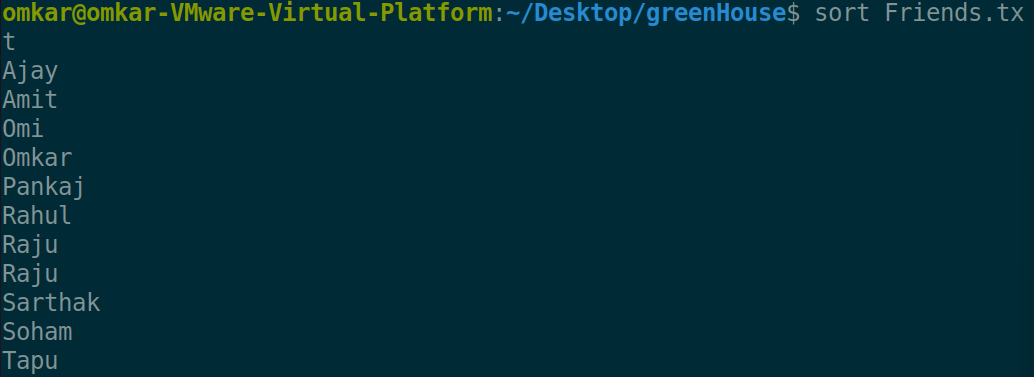
**Paste -d’,’ file1 file2 file3..**

It also Use to merge more than 2 files



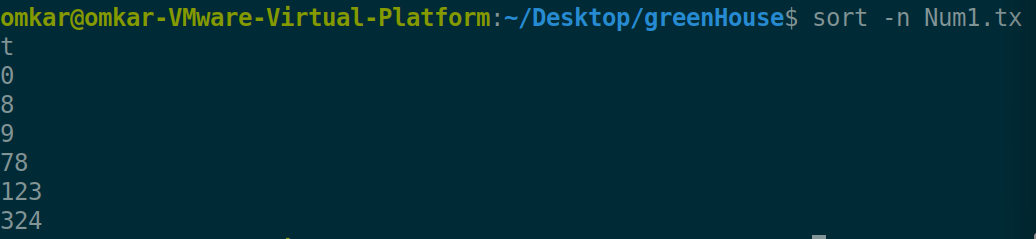
1. sort - sort lines of text files

**sort filename**



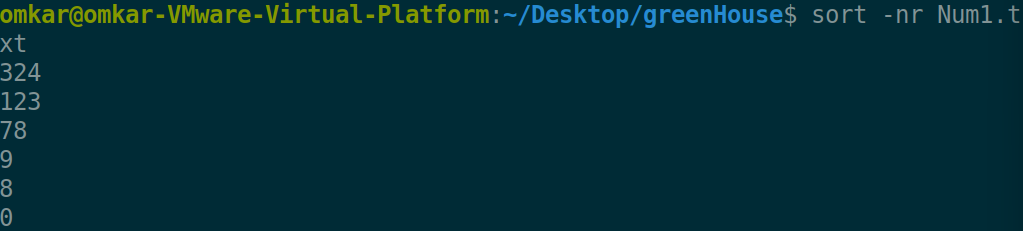
-n sort the file having numerical data

**sort -n filename**



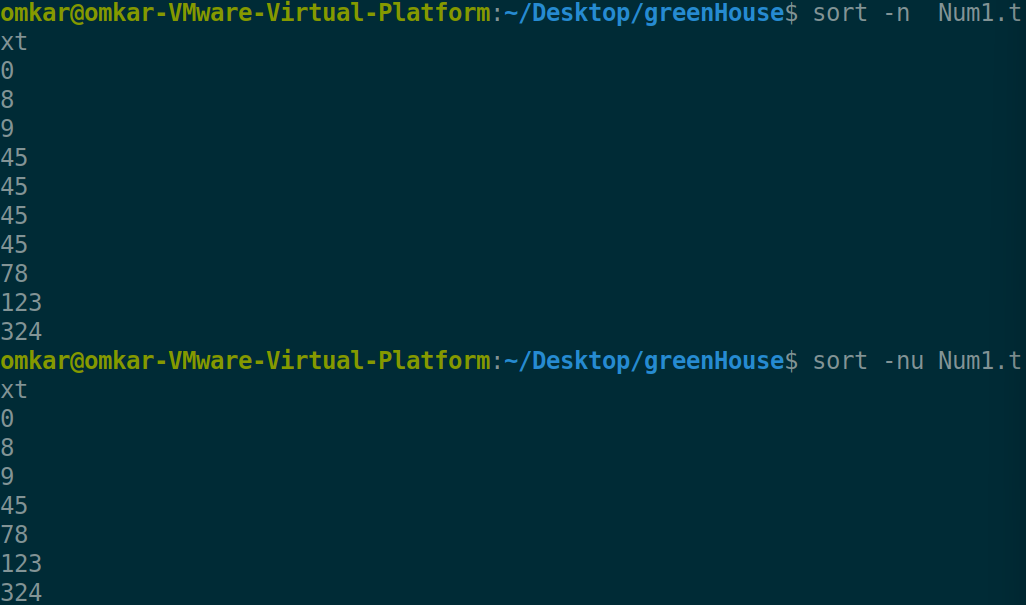
-r option is use to perform reverse sort

**sort -r filename**



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

**sort -u filename**



1. 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

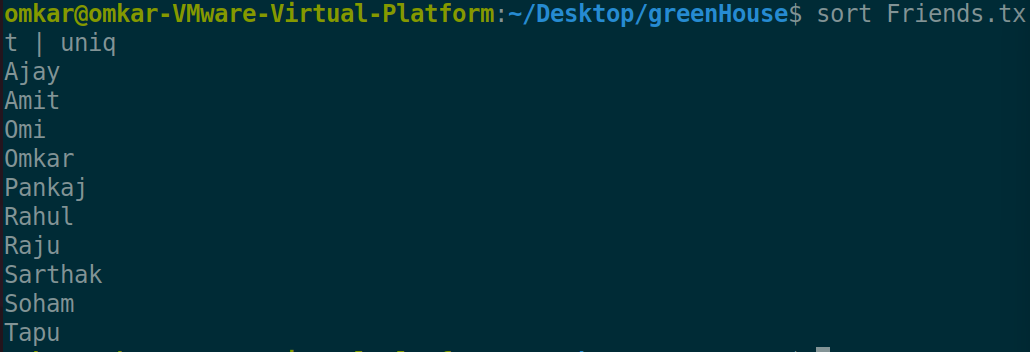
**uniq filename**





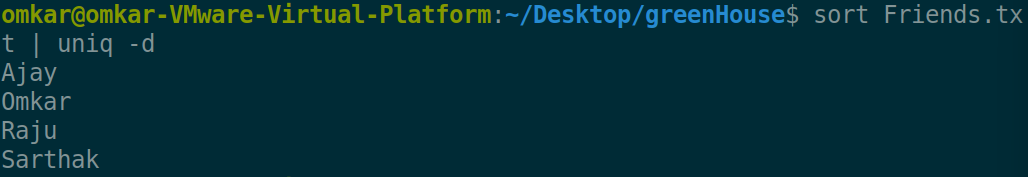
But if we need to do it for all names not for adjacent names then we need to use this with sort using pipe operator (same as -u option of sort command)

**sort filename | uniq**



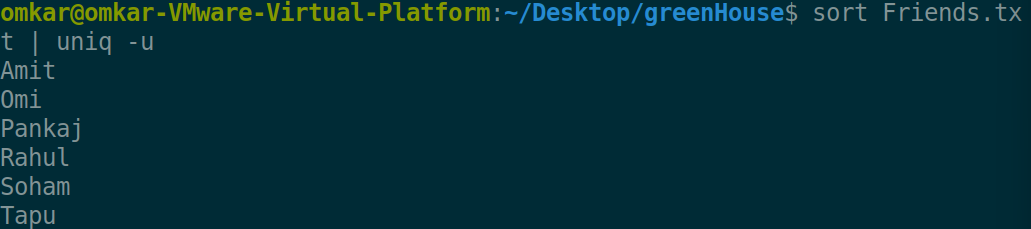
-d option of uniq command is to get only duplicate elements from the file data

**sort filename | uniq -d**



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

**sort filename | uniq -u**



|  |
| --- |
| -c gives us actual count of data repeated inside the file  **sort filename | uniq -c** |
|  |  |



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

**sort filename | uniq -c | sort -n**



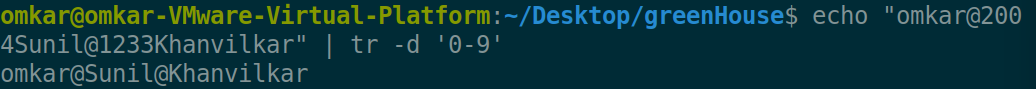
1. tr command is use to translate the particular character or set of characters with some different character

**cat filename | tr ‘a-z’ ‘A-Z’**



-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’**



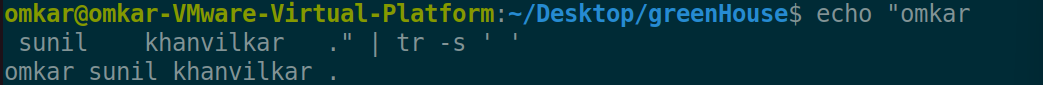
-c option is always use with the -d option because c help use to print the complement of the output given by option -d means it help to delete other character which are not specify in character attributes

**echo “content” | tr -cd ‘characterNotRemove’**



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

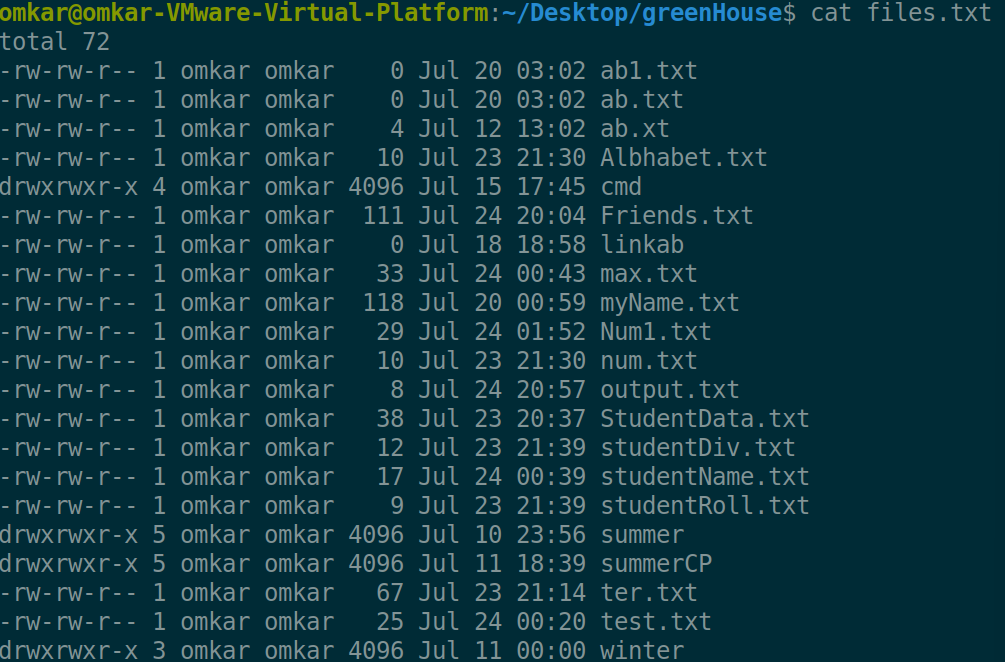
**echo “content” | tr -s ‘characterTobeSqueeze**

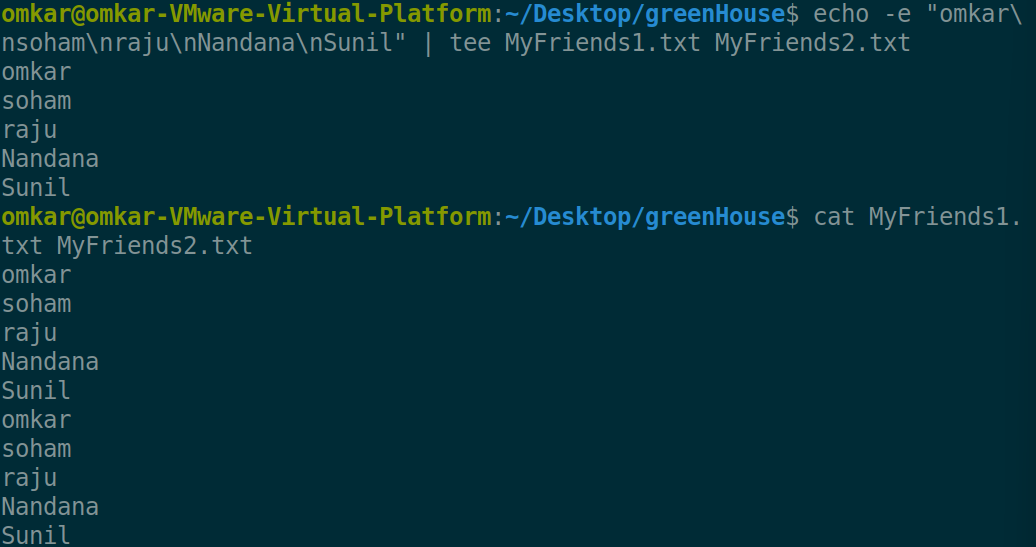


1. tee command in Linux reads from standard input and writes to both standard output and one or more files.

**Ls -l | tee file1 file2 … fileN**







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

**Ls -l | tee -a file1 file2 … fileN**



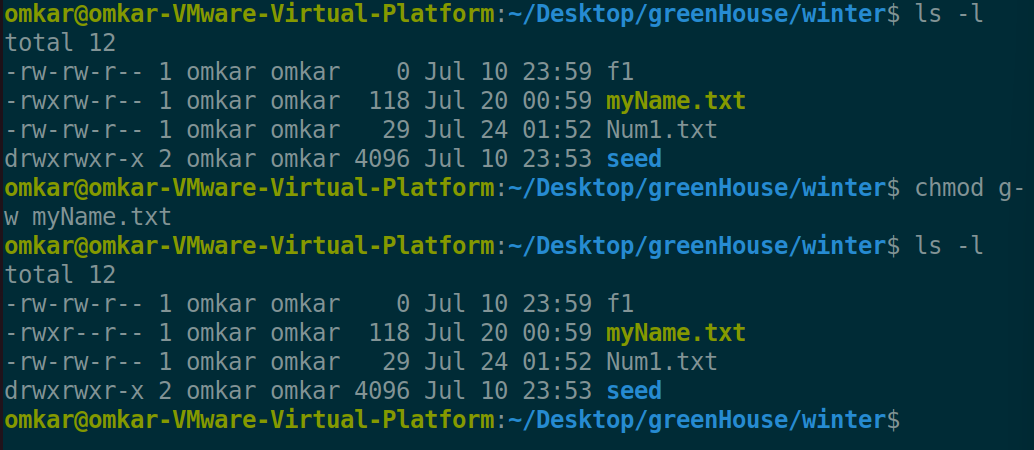
**File Permission related commands:**

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 |

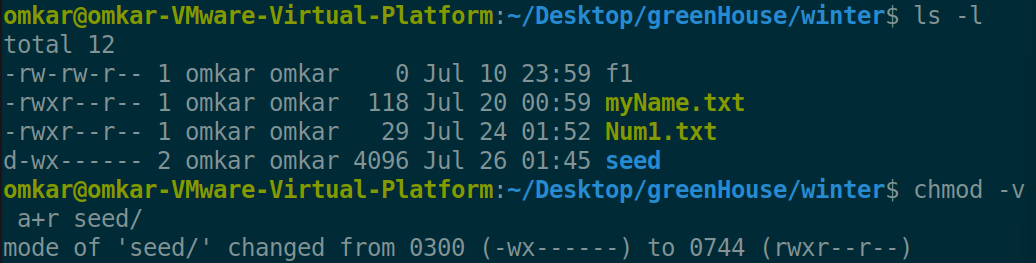
chmod command is use to change the mode of the file for different types of entities like user/group/others we use it for absolute as well as for relative mode changing

**chmod category operation permission fileName**



-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**



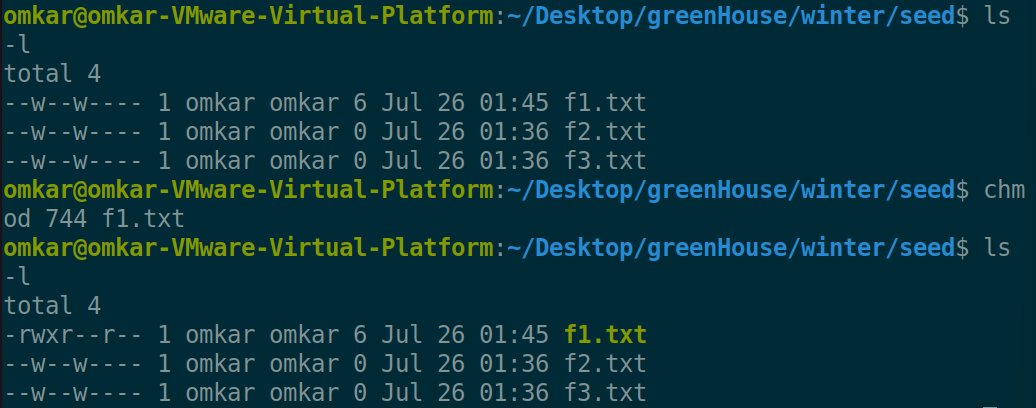
Changing permission absolute manner

|  |  |
| --- | --- |
| **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 |

|  |  |  |
| --- | --- | --- |
| **User** | **Group** | **Others** |

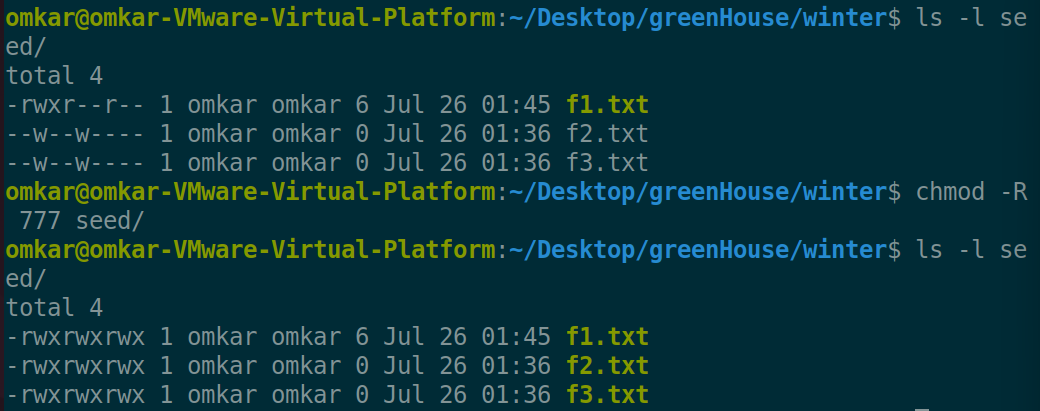
Proper Positions in command

**chmod absolute permission filename**



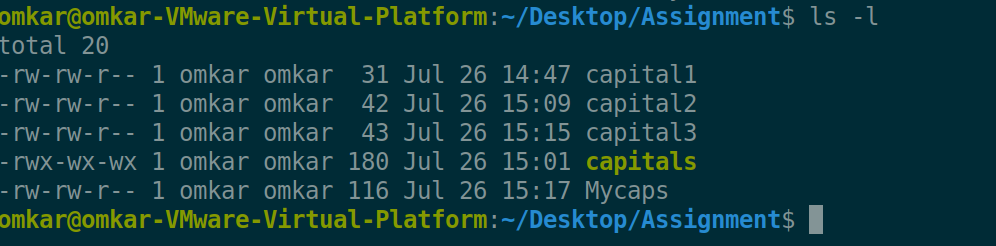
-R option R is use to recursively change the files permissions which are present inside a directory

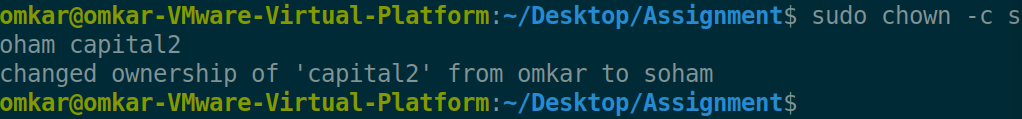
**chmod -R absolute permission filename**

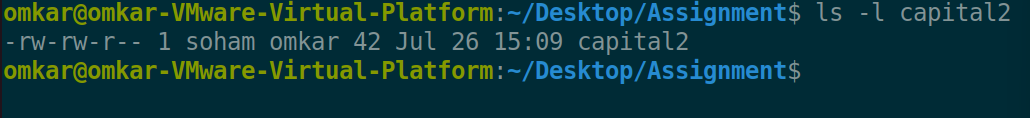


1. chown is use to change the owner for the particular file

**Sudo chown -c newOwner filename**

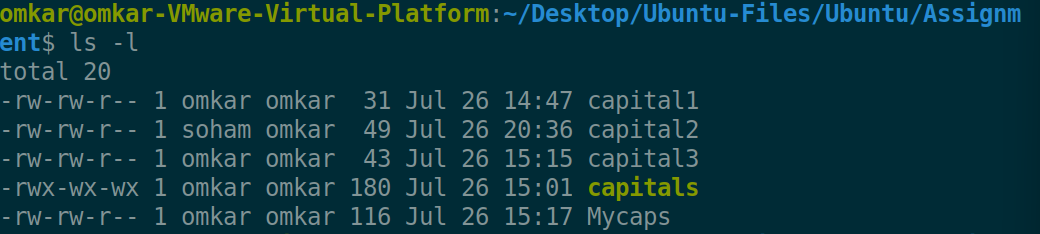


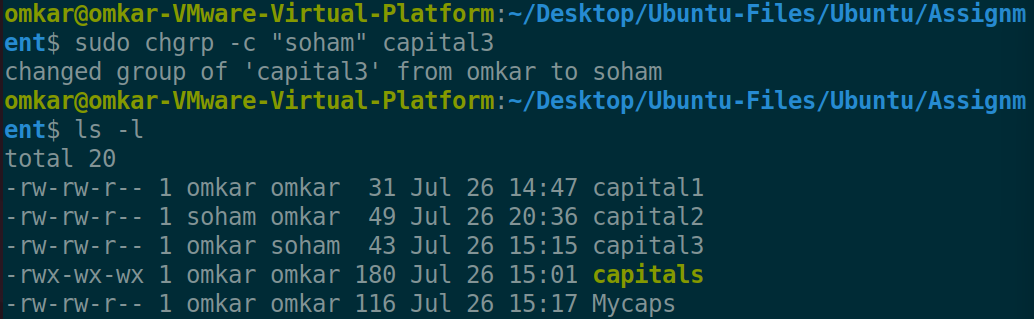




1. chgrp is use to change the group ownership

**Sudo chgrp -c newGroup filename**



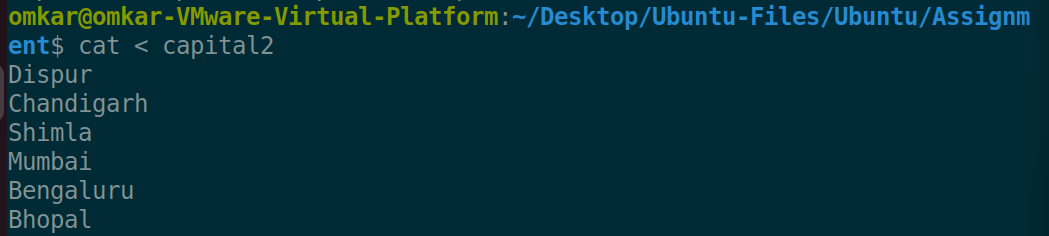


**Redirection: Provide a powerful command line controls**

Types of Redirection

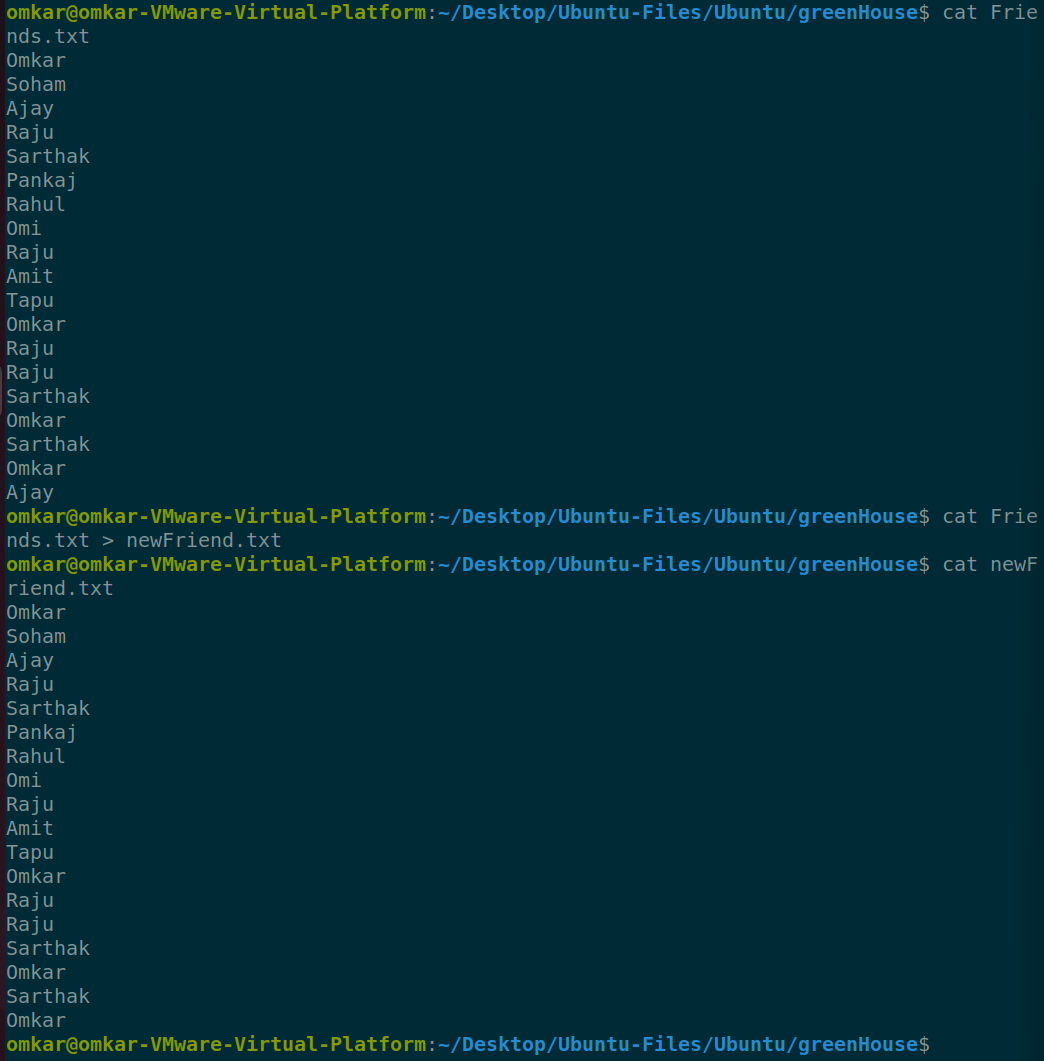
**Standard input redirection:**

**cat < fileName**



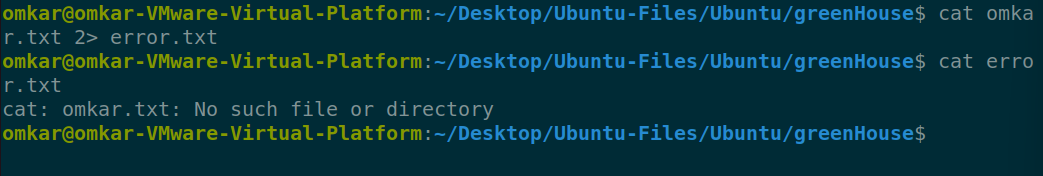
**Standard output redirection:**

**cat filename > NewFile**



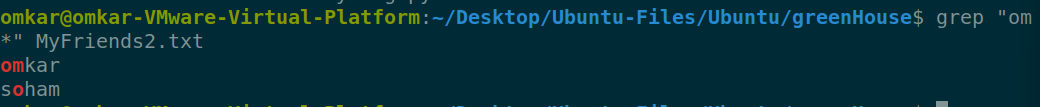
**Standard error redirection:**

**cat filename 2> NewFile**



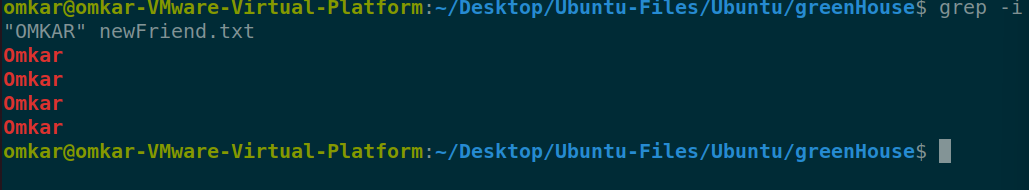
1. grep command is use to find out the strings or set of strings in the particular file or directory which are in particular pattern given in REGEX form

**grep “REGEX” fileName**



-i ignore case for matching

**grep -i “REGEX” fileName**



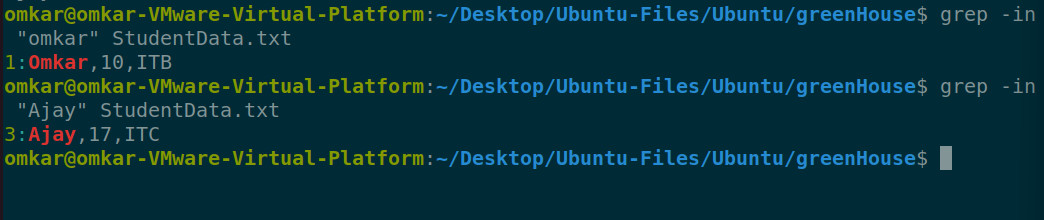
-v doesn’t display lines matching expression

**grep -v “REGEX” fileName**



-n display line numbers along of occurrences

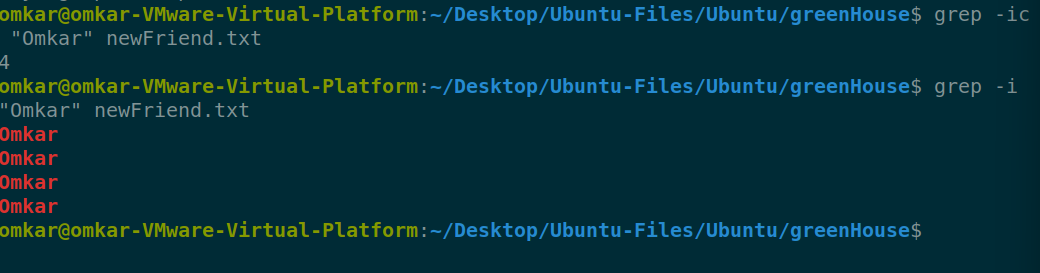
**grep -n “REGEX” fileName**



-c counting number of occurrences

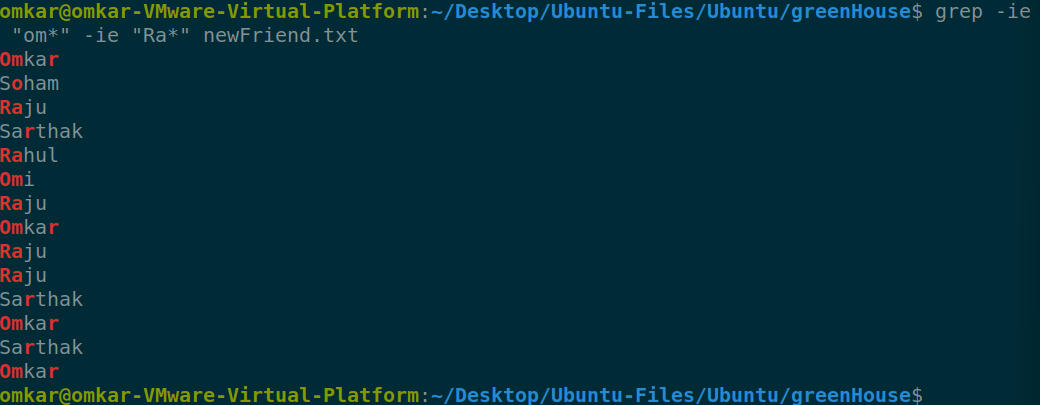
**grep -c “REGEX” fileName**

-l display list of file names



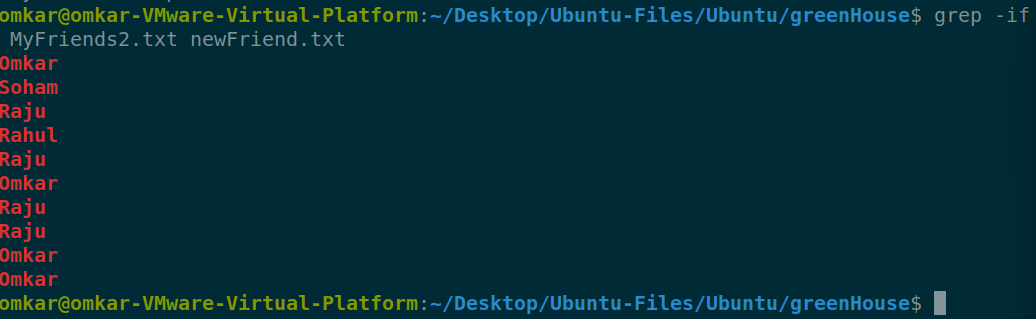
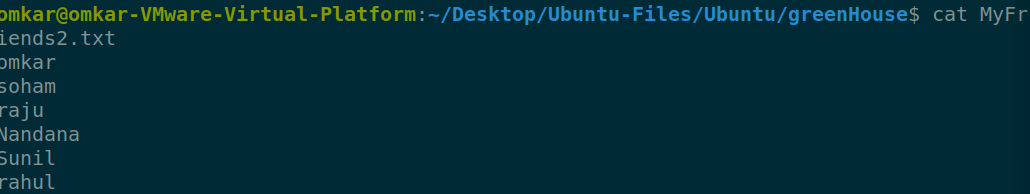
-e exp for matching

**grep -e “REGEX1” -e “REGEX2” fileName**



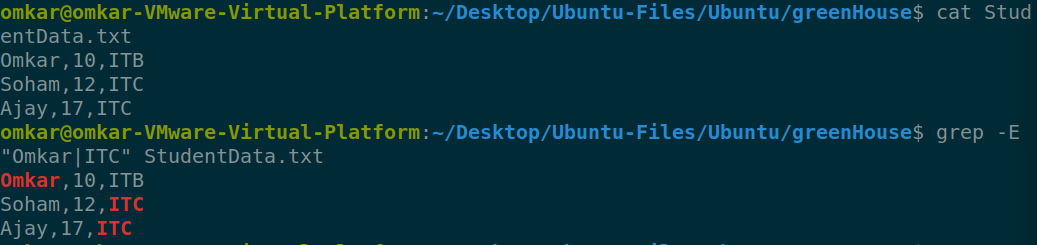
-f file take patterns from file

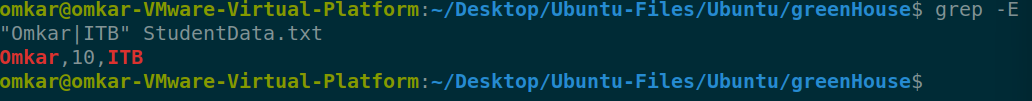
**grep -f pattern.txt fileName**



-E treat pattern as an extended reg. exp

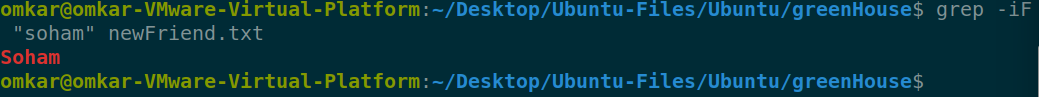
**grep -E fileName**





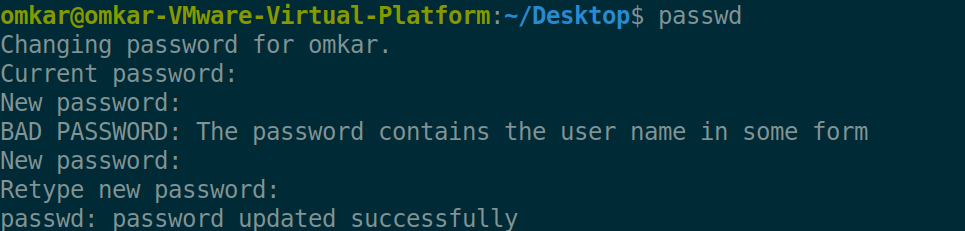
-F matches multiple fixed strings (fgrep)

**grep -F “fixed\_String” fileName**



# **Assignment 2 Questions**

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

****

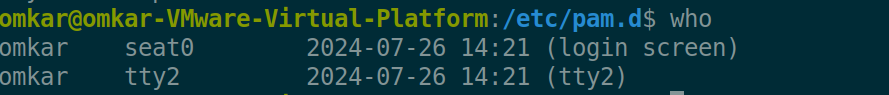
1. **Display the system’s date.**

****

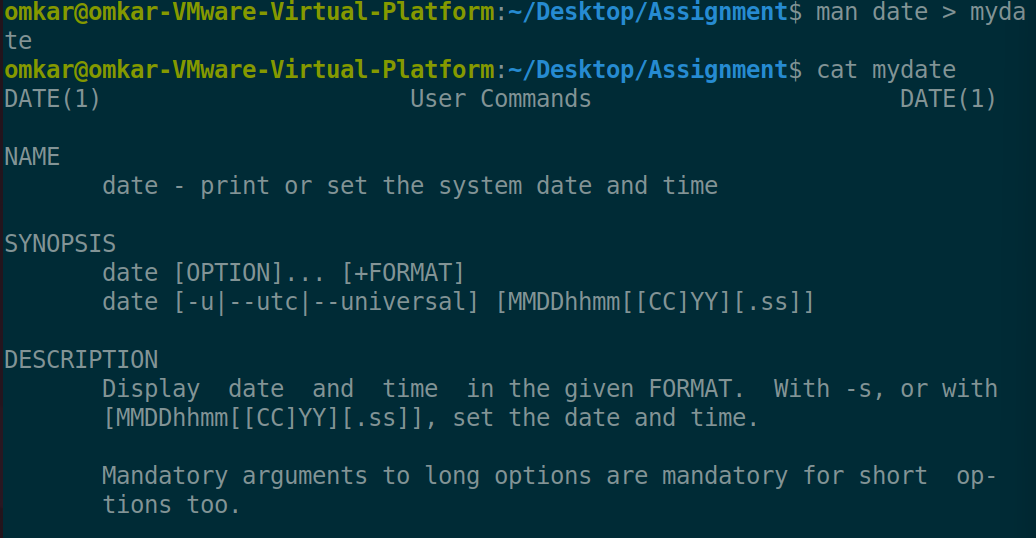
1. **Count the number of lines in the /etc/passwd file.**

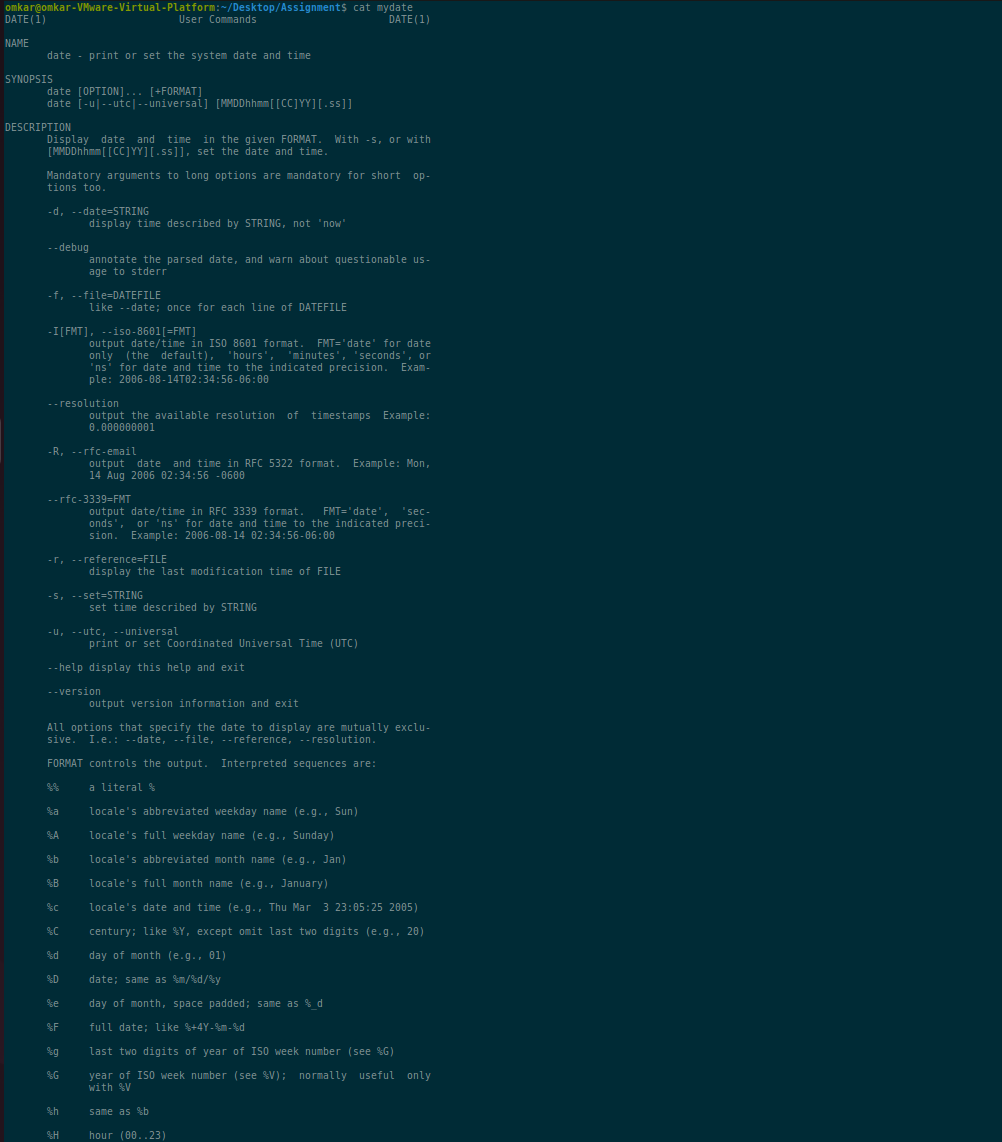
****

1. **Find out who else is on the system.**

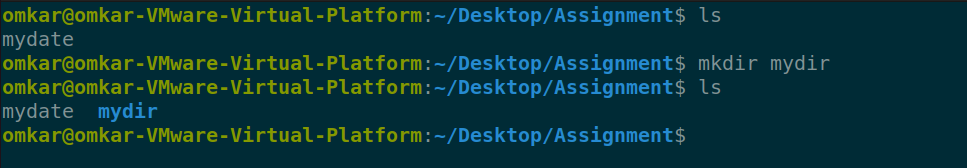
****

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

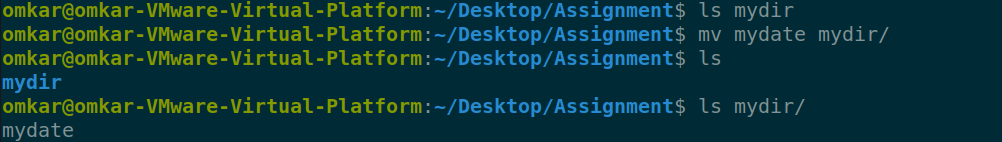
****

**** ****

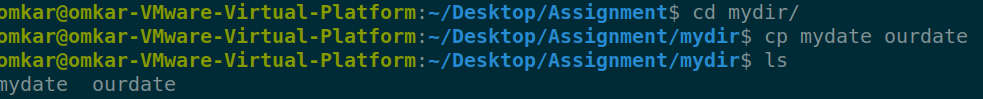
1. **Create a subdirectory called *mydir*.**

****

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

****

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

****

1. **List the contents of *mydir*.**

****

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

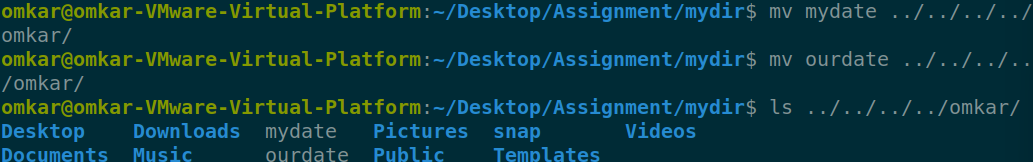
****

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

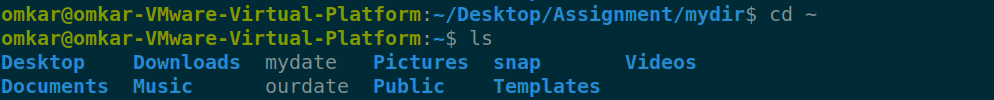
1. **Display the name of the current directory starting from the root.**

****

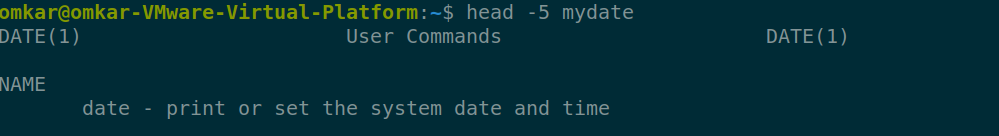
1. **Move the files in the directory *mydir* back to the HOME directory.**

****

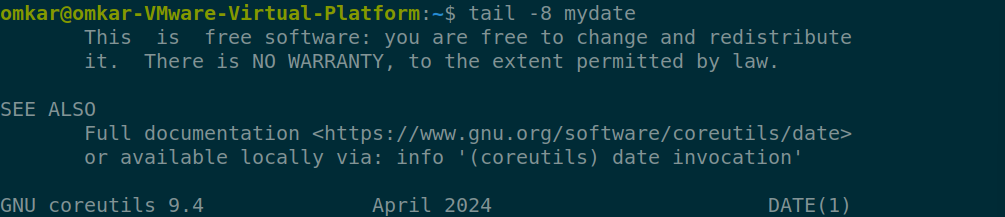
1. **List all the files in your HOME directory.**

****

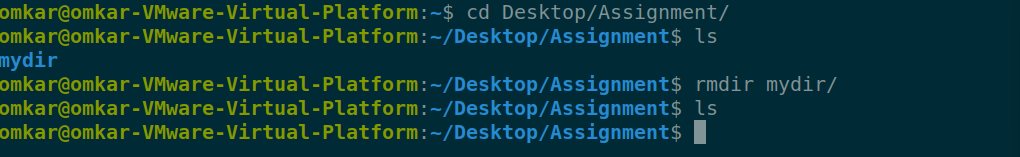
1. **Display the first 5 lines of *mydate*.**

****

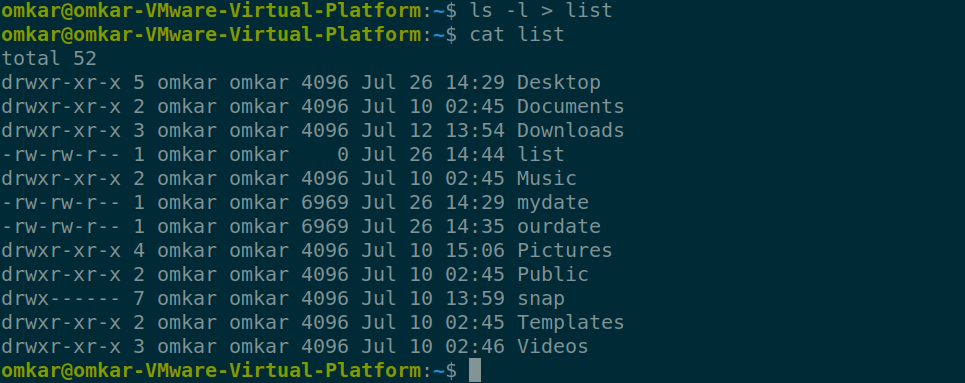
1. **Display the last 8 lines of *mydate*.**

****

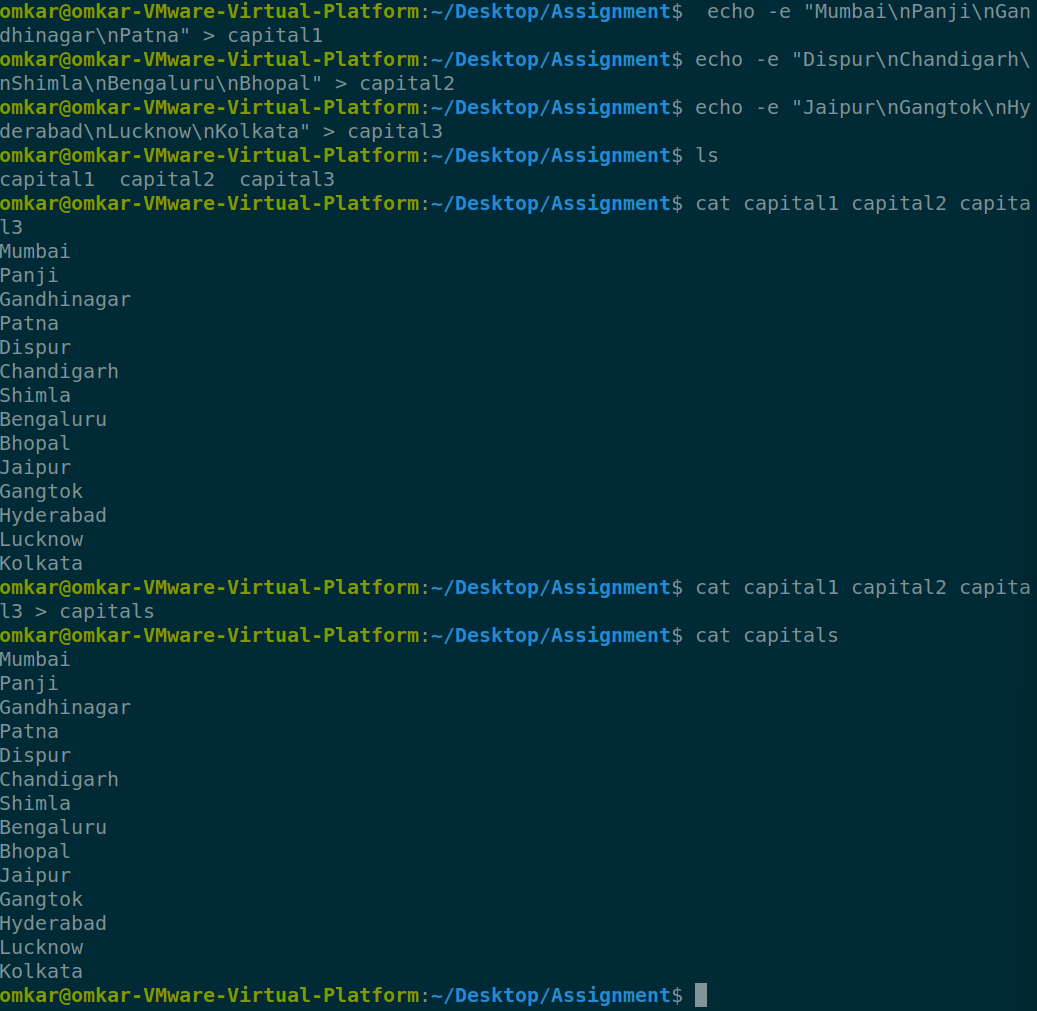
1. **Remove the directory *mydir*.**

****

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

****

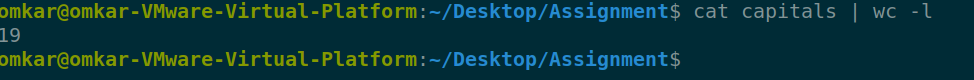
1. **Select any 5 capitals of states in India and enter them in a file named *capitals1*. Choose 5 more capitals and enter them in a file named *capitals2*. Choose 5 more capitals and enter them in a file named *capitals3*. Concatenate all 3 files and redirect the output to a file named *capitals*.**

****

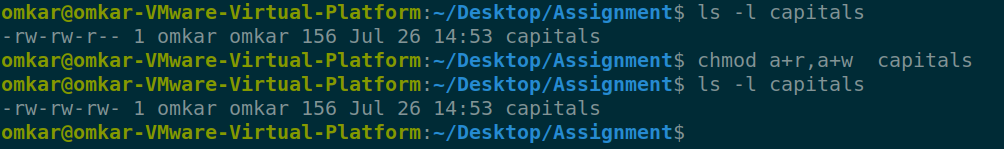
1. **Concatenate the file *capitals2* at the end of file *capitals*.**

****

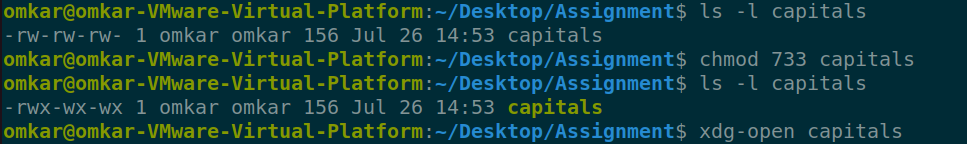
1. **Redirect the file *capitals* as an input to the command “wc –l”.**

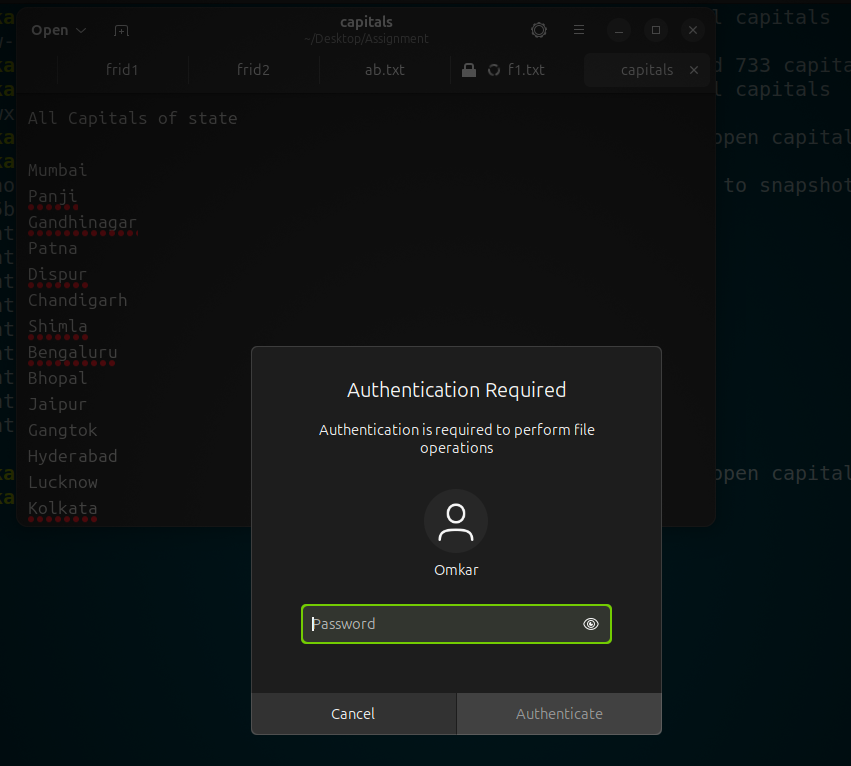
****

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

****

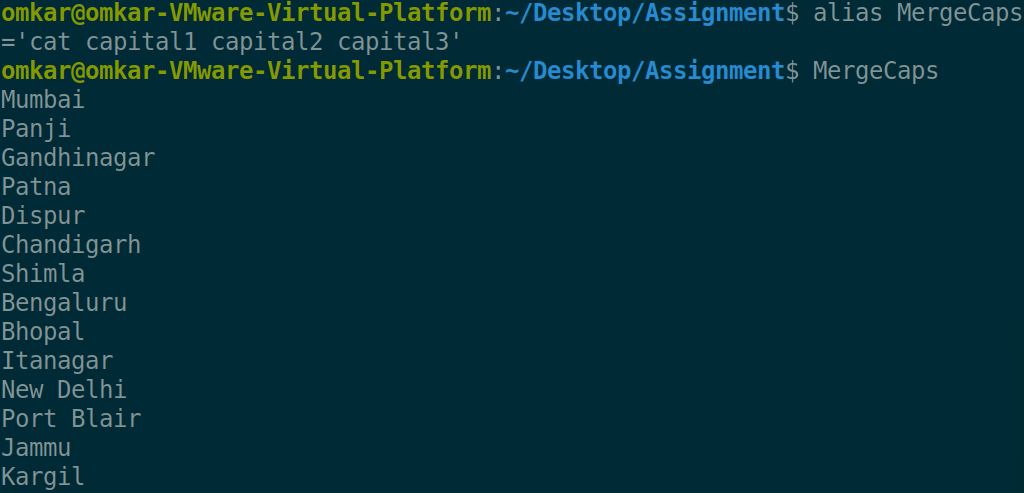
1. **Give read permissions only to the owner of the file *capitals*. Open the file, make some changes and try to save it. What happens ?**

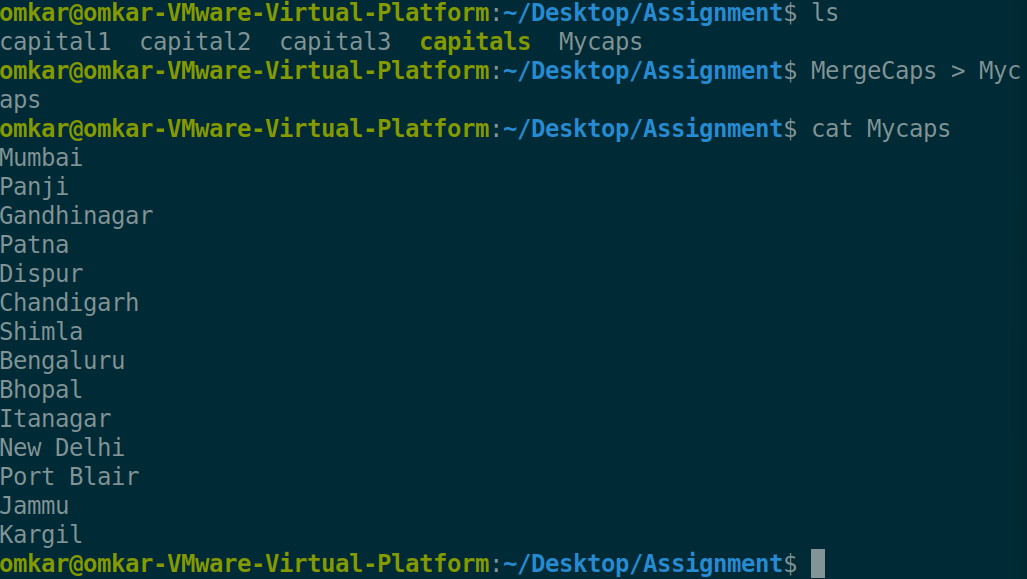
****

****

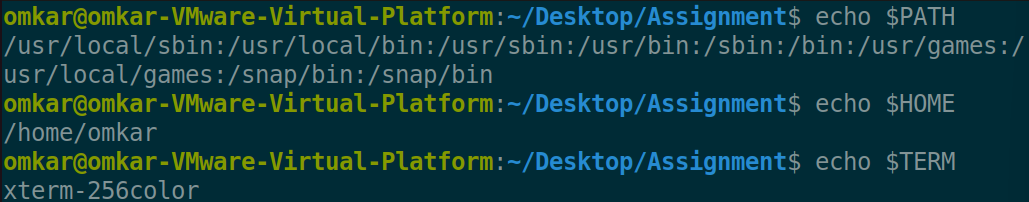
**Asking about the user password for authentication of owner**

1. **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.**

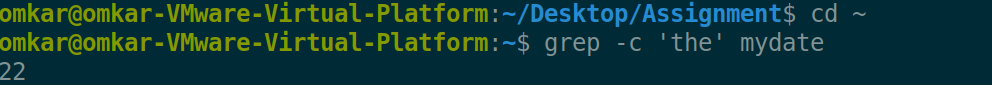
****

****

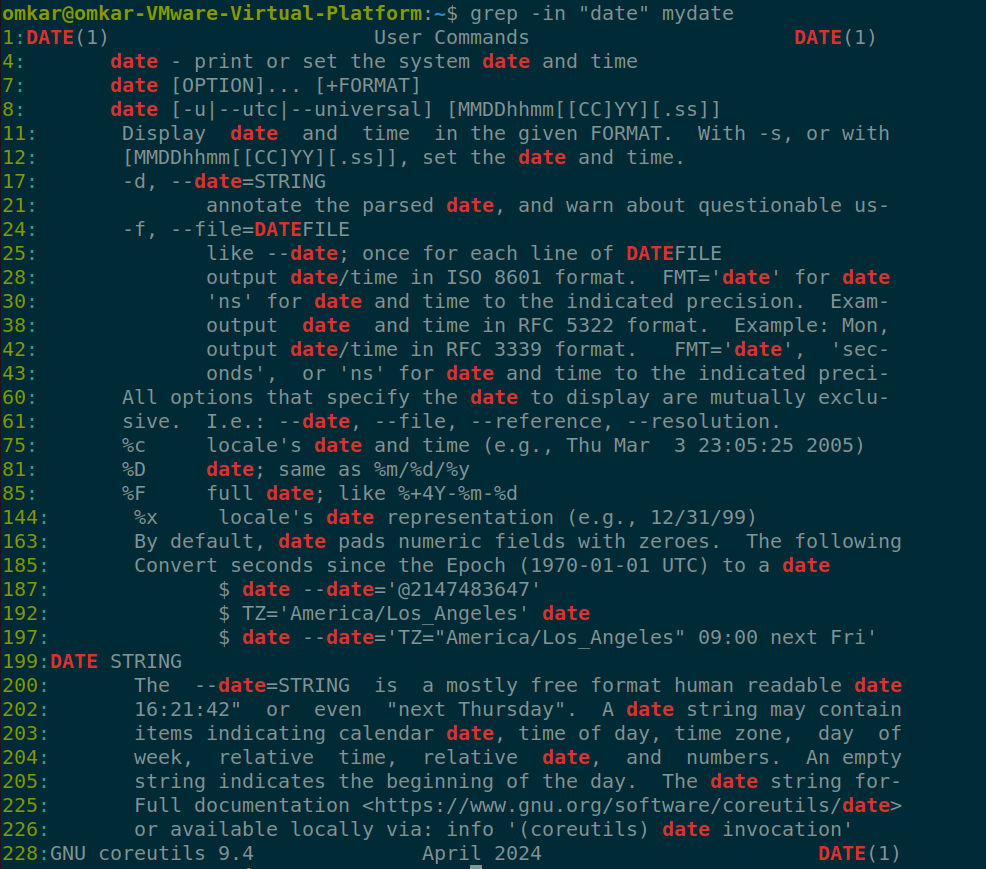
1. **What are the environment variables PATH, HOME and TERM set to on your terminal ?**

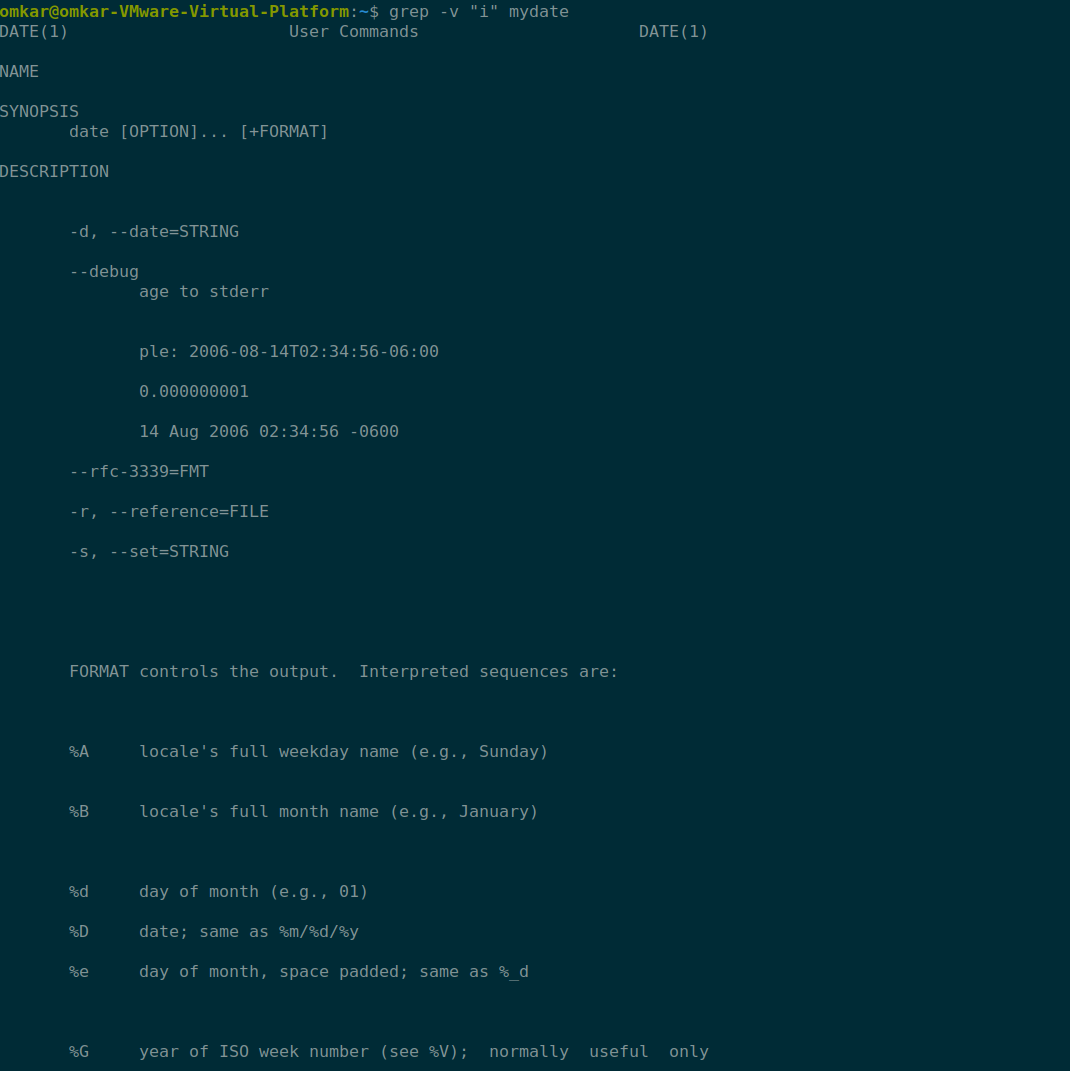
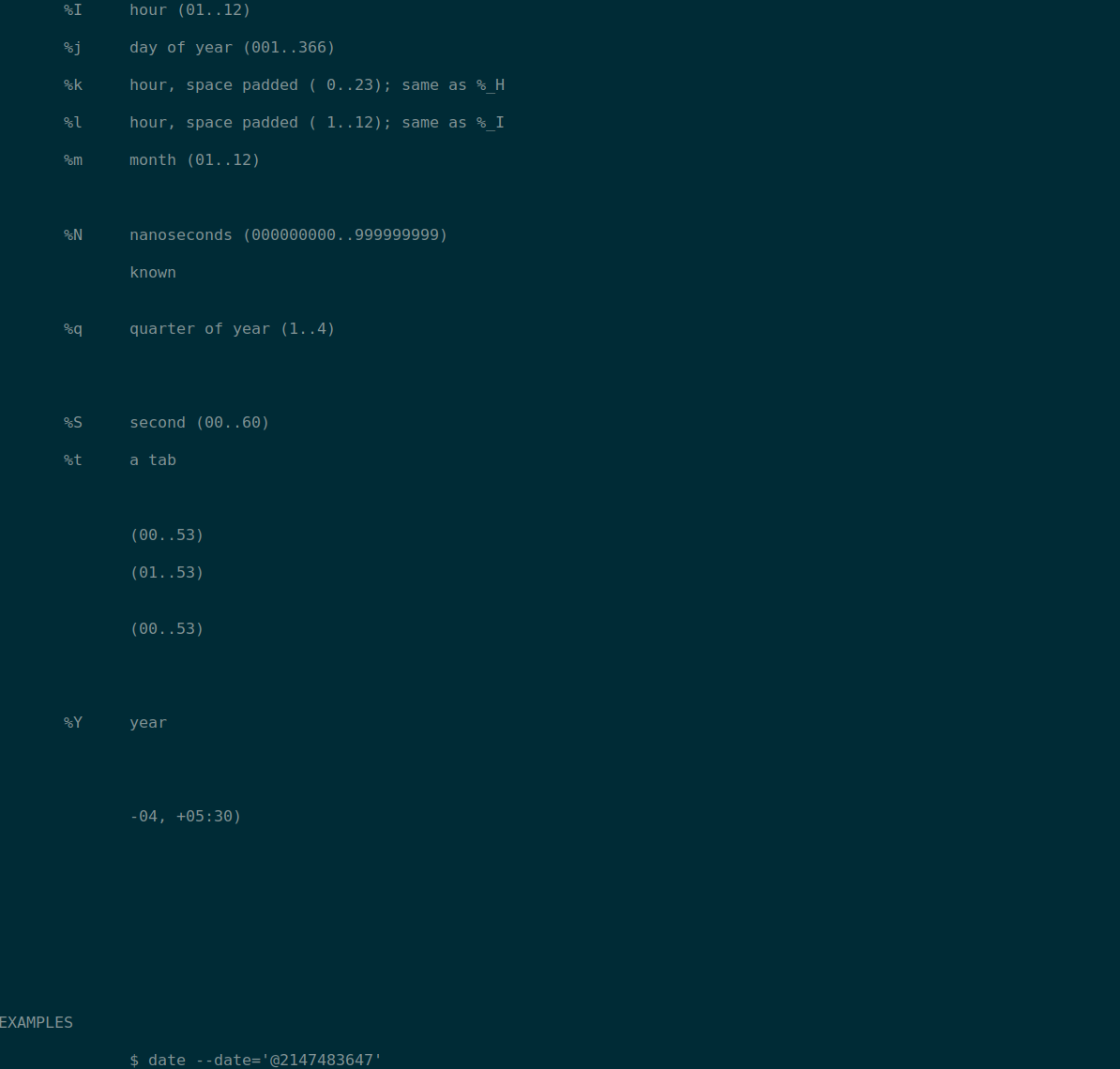
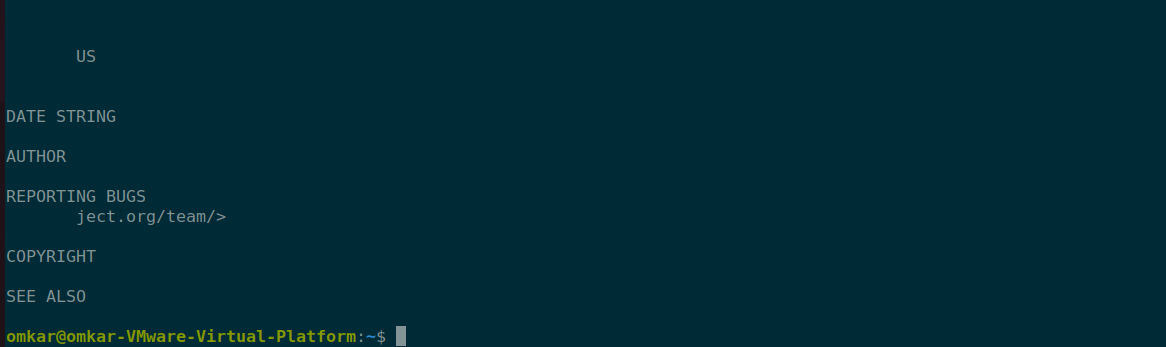
****

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

****

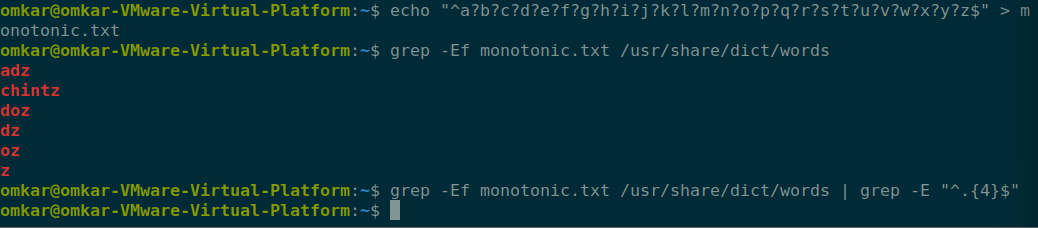
1. **Find out the line numbers on which the string “date” exists in *mydate*.**

****

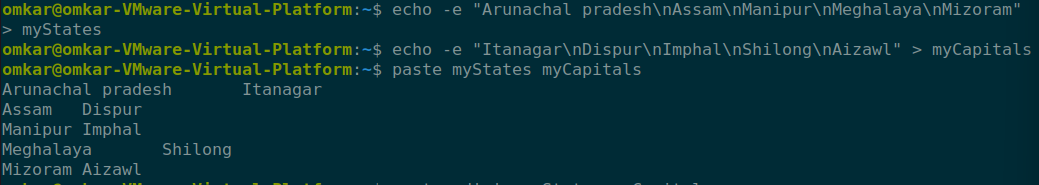
1. **Print all lines of *mydate* except those that have the letter “i” in them.**   
2. **Create the file *monotonic* as follows:**

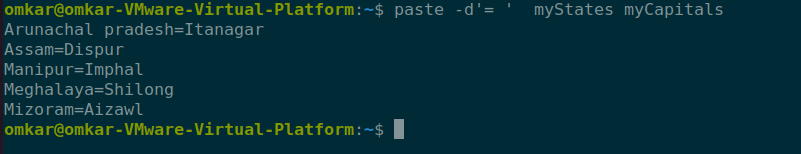
**^a?b?b?c?…………..x?y?z$**

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

****

1. **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.**

****

****

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

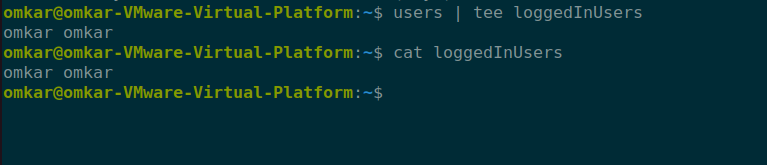
**Column 1**

****

**Column 3**

****

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

****