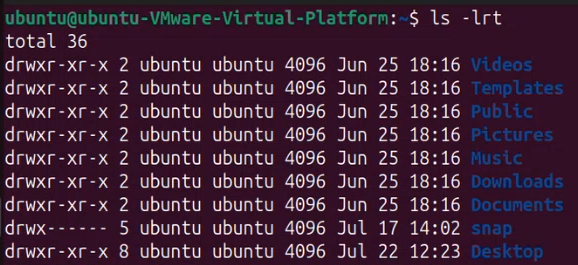
**20THCLASS-UNIX NOTES:**

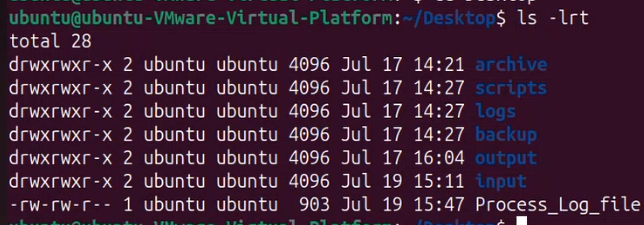
|  |  |
| --- | --- |
| Listing command |  |
| 1**. ls -lrt** dr—directory | --detailed mode/ listFiles |



|  |  |
| --- | --- |
| 2. ls | -It provides valuable information about files, directories, and their attributes. |

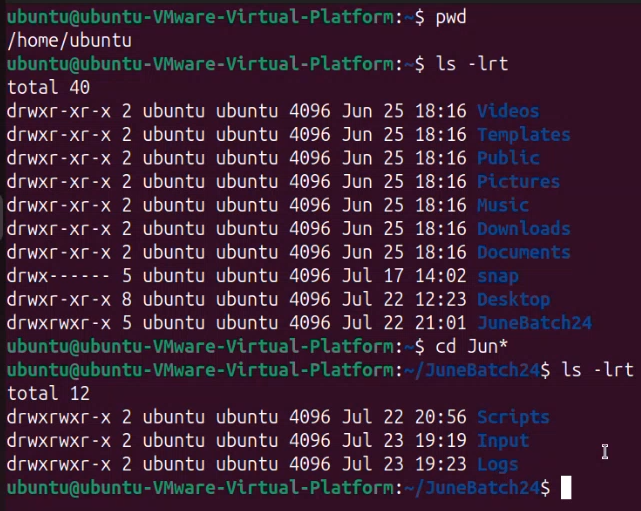
Output: Desktop Documents Downloads Music Pictures Public snap Templates Videos

|  |  |
| --- | --- |
| 3. cd (change directory) | **cd Desktop** |
| 4.listing in desktop | **ls -lrt** |

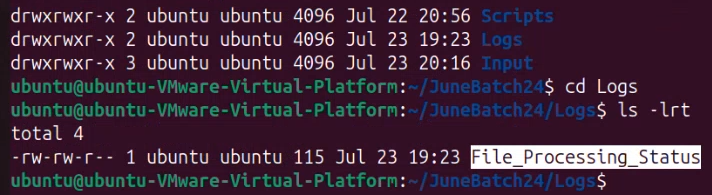


|  |  |
| --- | --- |
| Input dir  Output dir  logs | Input dir -it is dir which is created to take feed from other organisation  O-output dir is a dir wherein you are pushing the files to the other system or want to push it  Script- where is the process invocation details are stored,ur input scripts and output scripts  Logs-dir which helps you to understand whether my process is running or not ,it generated log for that process taking the feed from the input |
| 4. cd.. | to come out of directory and cd works for only folder not file |
| 5.pwd | present working directory  cd Downloads  pwd  /home/ubuntu/Downloads |
| If it is folders it is denoted by d  s -system files  l-log files  If it is file it |  |
|  | create a new folder |
| 6.mkdir - make directory  mk-make  dir-directory | ex-  mkdir "foldername"  mkdir JuneBatch24  mkdir Scripts  mkdir Logs  mkdir Input  mkdir Output |
| 8. cd June\* | is used in a Unix-like shell (e.g., bash) to change the current directory to one that matches the pattern "June\*" |
| 9. cd D\*{TAB} –  TO list the directories that start with D | Examples: PICTURES/PUBLIC  Examples: Desktop/ Documents/ Downloads/ |
| 7.mkdir [dir1] [dir2] ......  -for creating multiple directories | Examples- mkdir Backup Archive |
| 10. clear | clear everything |
| 11. history | -view history of all commands |
| 12. date | -view date and time  Ex:Mon Jul 22 08:53:18 PM IST 2024 |
| 13. rmdir -remove directory | Ex: rmdir "Foldername" rmdir Archive |
| 14**. rmdir** [dir1] [dir2] ......  - remove multiple directory: | Ex:  **rmdir** Output Backup |
|  | 1.mkdir Scripts mkdir Logs  mkdir Input  mkdir Output  2.ls -lrt  3.pwd  /home/ubuntu/JuneBatch24  4.mkdir Backup Archive |

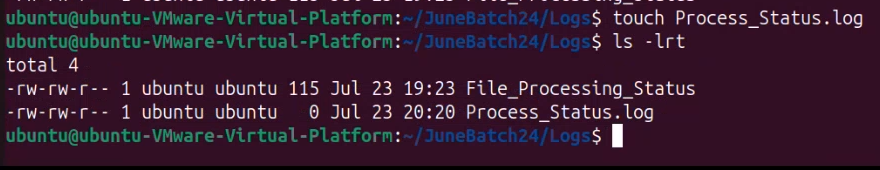
**21THCLASS** 23-07-2024



|  |  |
| --- | --- |
| 1. cd Input   ls -lrt | In input dir you should have backup in backup dir you should have archive |
| 1. mkdir -p Backup/Archive | ls -lrt |
|  | I get “Backup” directory |
| 1. cd Backup | ls -lrt |
|  | I get “Archive” directory |
| 1. creating test1 and test2 in “backup directory” | mkdir test1 test2 |
| 1. rmdir Archive test1 test2 | Removing multiple directories |
| 1. rmdir Backup | Removing backup directory |
| 1. pwd | /home/ubuntu/JuneBatch24/Input |
|  |  |
| In JuneBatch24 we have created 3 directories  -Scripts  -Logs  -Input |  |
| 1. cd Logs | go to logs folders |



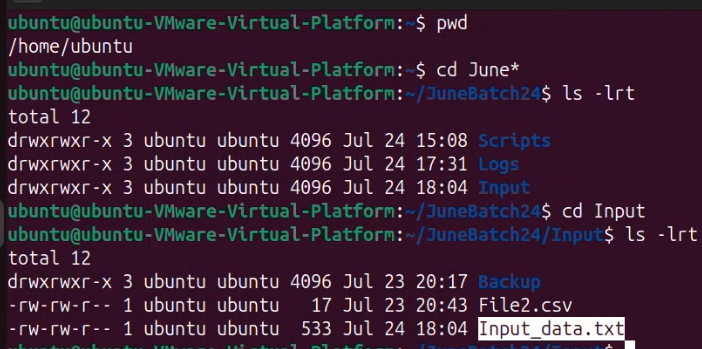
|  |  |
| --- | --- |
| 1. **touch** Process\_Status.log | with touch cmd you’ll b able to create **0 byteFile** |
| 1. ls -lrt | Listing the content inside the file |
| 1. **view** Process\_Status.log | view command |
|  | Ex: “Process\_Status.log”[readonly] 0L, 0B  The view cmd helps you to view the file in readonly mode  -to c the content of file |
| 1. **view** File\_Processing\_Status | Ex: File initiation has started  File loading process is successfully completed  File processing completed successfully  “File\_Processing\_Status”[readonly] 4L, 115B |



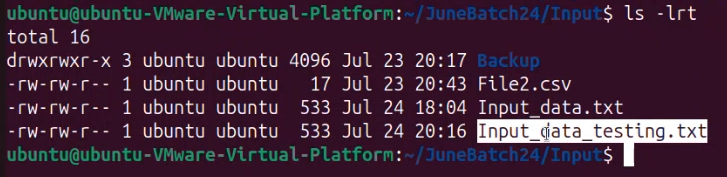
|  |  |
| --- | --- |
| 1. vi File\_Processing\_Status |  |
| 1. **Esc+i** | It will help you to insert the data |
| 1. **Esc + dd**(two times) | Delete a particular line |
| 1. **Esc+x** | erase one character at a time or char by char |
| 1. **Esc+yy** | Copy the text/text |
| 1. Esc+p | Paste the text |
| 1. Esc button+ :wqs | Save the changes |
| 1. :q! | I dont want to save the file , quit after writing |
| 1. :wq or ctrl + s | Save and exit |
| 1. ctrl + s | Save in normal notepad and exit |
|  | **vi File\_Name** Esc key is important  **esc i** - insert text file  **esc dd** - delete a single line at a time  **esc x** - erase one character at a time  **esc yy** - copy (before executing this cmd, copy the test you want o paste  **esc p** - Paste |
| 1. **cat -concatenation** |  |
| 1. **touch** file1 file2 file3 | At one go 3 files creates in Logs directory |
| 1. **vi file1** | Esc+i  File One Details  esc :wq! |
| 1. **cat file1** | cat cmd to view the details |
| 1. **vi file2** | Esc+i  File Two Details  esc :wq! |
| 1. **vi file3** | Esc+i  File Three Details  esc + wq! |
| 1. **cat file1 file2 file3> file4** | put this into file4 |
| 1. **cat file4** | File One Details File Two Details  File Three Details |
| **touch file5 file6**  **vi file5=** File five details  **vi file6**= File six details **cat file5 file6 > file4**  **cat file4(it will overwriting previous file and creating a new file)** | **cat file4**  output:  File five details  File six details |
| **cat** file1 file2 file3> file4 | **cat file4**  File One Details File Two Details  File Three Details |
| **cat file5 file6 >> file4**  **> - this will overwrite the data**  **>>- this will append the data** | **cat file4**  File One Details File Two Details  File Three Details  File five details  File six details |

**22NDCLASS**

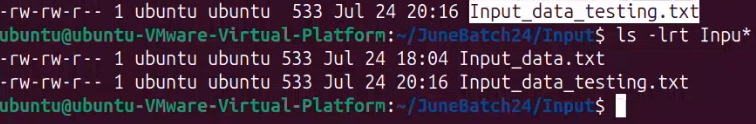
|  |  |
| --- | --- |
| pwd | **/home/ubuntu** |
| cd june\* |  |
| ls-lrt |  |

****

|  |  |
| --- | --- |
| cat **Input\_data.txt** | output:  Line Number One Details  Line Number Two Details  Line Number Three Details  Line Number Four Details  Line Number Five Details  Line Number Six Details  Line Number Seven Details  Line Number Eight Details  Line Number Nine Details  Line Number Ten Details  Line Number Eleven Details  Line Number Twelve Details  Line Number Thirteen Details  Line Number Fourteen Details  Line Number Fifteen Details  Line Number Sixteen Details  Line Number Seventeen Details  Line Number Eighteen Details  Line Number Twenty Details |
| copy-creating the copy of the file | **cp Input\_data.txt Input\_data\_testing.txt** |

****

|  |  |
| --- | --- |
| ls -lrt Inpu\* |  |

****

|  |  |
| --- | --- |
| mkdir testing | creating testing directory |
| cd Testing |  |
| ls-lrt | output:  **-rw-rw-r--1 ubuntu ubuntu 533 Jul 24 20:19** input\_data\_testing.txt |
| cp Input\_data.txt Testing/Input\_data\_testing.txt | cp input\_data.txt directory\_name/ new\_name |
| cd Testing |  |
| ls-lrt | **List files in testing directory** |
| Copy multiple files | cp File2.csv Input\_Data.txt Testing  ex:  Input\_data\_testing.txt  File2.csv  Input\_data.txt |
| vendor to organization dir  Homework Ftp and smtp | **After the processing is done I need to remove from the dir.** |
| touch file1 file2 |  |
| mv- move command | -mv file2 file2\_processed  Actual file is moved to another dir  N copied file stays in same dir  -mv cmd renames the existingFile to newName  -originalFail was retained,  -file2 has moved to file2\_processed |
| mv file1 Backup/file1\_processed | move file1  -It has moved to backup  **copy**  **move** |

|  |  |
| --- | --- |
| In Input directory |  |
| **cat Input\_data.txt**  ex: Number - **bill number**  details - **amount** | **Line Number One Details**  **Line Number Two Details**  **Line Number Three Details**  **Line Number Four Details**  **Line Number Five Details**  **Line Number Six Details**  **Line Number Seven Details**  **Line Number Eight Details**  **Line Number Nine Details**  **Line Number Ten Details**  **Line Number Eleven Details**  **Line Number Twelve Details**  **Line Number Thirteen Details**  **Line Number Fourteen Details**  **Line Number Fifteen Details**  **Line Number Sixteen Details**  **Line Number Seventeen Details**  **Line Number Eighteen Details**  **Line Number Nineteen Details**  **Line Number Twenty Details** |
| **cut -cuts file in vertical manner**  **“-d” -delimiter**  **“-f “ -field**  **cut -d ’ ’-f 2,3 Input\_data.txt** | Number One  Number Two  Number Three  Number Four  Number Five  Number Six  Number Seven  Number Eight  Number Nine  Number Ten  Number Eleven  Number Twelve  Number Thirteen  Number Fourteen  Number Fifteen  Number Sixteen  Number Seventeen  Number Eighteen  Number Nineteen  Number Twenty |
| **cut command** | **cut -d ’ ’-f 2,3 Input\_data.txt>Input\_req\_data.txt** |
| **cat input\_req\_data.txt** | Number One  Number Two  Number Three  Number Four  Number Five  Number Six  Number Seven  Number Eight  Number Nine  Number Ten  Number Eleven  Number Twelve  Number Thirteen  Number Fourteen  Number Fifteen  Number Sixteen  Number Seventeen  Number Eighteen  Number Nineteen  Number Twenty |
| horizontal cutting |  |
| **head -10 Input\_data.txt** | It gives top 10 values from top  Line Number One Details  Line Number Two Details  Line Number Three Details  Line Number Four Details  Line Number Five Details  Line Number Six Details  Line Number Seven Details  Line Number Eight Details  Line Number Nine Details  Line Number Ten Details |
| **tail -10 Input\_data.txt** | Line Number Twelve Details  Line Number Thirteen Details  Line Number Fourteen Details  Line Number Fifteen Details  Line Number Sixteen Details  Line Number Seventeen Details  Line Number Eighteen Details  Line Number Nineteen Details  Line Number Twenty Details \n new line |
| **wc -l Input\_data.txt**  **-cmd to count the no of lines in file** | Ex:21 Input\_data.txt |
| **vi Input\_data.txt** |  |
| **tail -1 Input\_data.txt** | 20 |
| **wc -l Input\_data.txt** | 21 Input\_data.txt |
| **Print**  **echo** | printf ”Hello World”  echo “Hello World”  echo “File processing has started”>Log\_file.log |
| **cat Log\_file.log** | “File processing has started |
| **Printf ”1st file is picked for loading”>>** Log\_file.log |  |
| **cat Log\_file.log** | File processing has started  1st file is picked for loading |
| **Printf ”2nd file is picked for loading”>>** Log\_file.log |  |
| **cat Log\_file.log** | File processing has started  1st file is picked for loading**2nd file is picked for loading** |
| **echo “3rd file is picked for loading”>>Log\_file.log** |  |
| **cat Log\_file.log** | File processing has started  1st file is picked for loading**2nd file is picked for loading3rd file is picked for loading** |
| **echo “4th file is picked for loading”>>Log\_file.log** |  |
| **cat Log\_file.log** | --File processing has started  1st file is picked for loading**2nd file is picked for loading3rd file is picked for loading**  --4th file is picked for loading |
| **printf”\n5th file is picked for loading\n”>> Log\_file.log** |  |
| **cat Log\_file.log** | --File processing has started  1st file is picked for loading**2nd file is picked for loading3rd file is picked for loading**  --4th file is picked for loading  --5th file is picked for loading |
| **printf”6th file is picked for loading\n”>> Log\_file.log** |  |
| **cat Log\_file.log** | --File processing has started  1st file is picked for loading2nd file is picked for loading3rd file is picked for loading  --4th file is picked for loading  --5th file is picked for loading  --6th file is picked for loading |

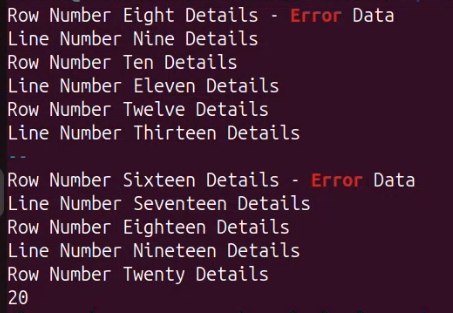
**Revision:**

|  |  |
| --- | --- |
| **Copy- copy the file in the same directory and copying a file in other directory**  **why copy is used, so that you impact the org file ,you can create a copy of it and play around with copied file** |  |
| **copy and move difference** | copy cmd- copy cmd ll copy org file without making any changes into it  New file and org file will b there |
| **Move –** | **it’ll rename the org file**  **It will move from source dir to destination dir**  **original file will not be there** |
| **How to cut the file vertically?** | demiliter ‘ ’ [columnno] |
| **How to cut the file horizontally?** |  |
| **How can I count the no of lines in File?** | wc -l filename |

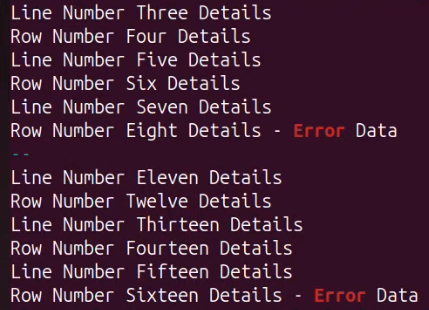
**23RDCLASS**

|  |  |
| --- | --- |
| grep “Line” Input\_data.txt |  |
|  | **Line Number One Details**  **Line Number Three Details**  **Line Number Five Details**  **Line Number Seven Details**  **Line Number Nine Details**  **Line Number Eleven Details**  **Line Number Thirteen Details**  **Line Number Fifteen Details**  **Line Number Seventeen Details**  **Line Number Nineteen Details** |
| grep “LINE” Input\_data.txt |  |
| grep -i“LINE” Input\_data.txt  “-i” case insensitive | Line Number One Details  Line Number Three Details  Line Number Five Details  Line Number Seven Details  Line Number Nine Details  Line Number Eleven Details  Line Number Thirteen Details  Line Number Fifteen Details  Line Number Seventeen Details  Line Number Nineteen Details |
| grep -i “lInE” Input\_data.txt | **Same output as above** |
| cat Input\_data.txt | Line Number One Details  Row Number Two Details  Line Number Three Details  Row Number Four Details  Line Number Five Details  Row Number Six Details  Line Number Seven Details  Row Number Eight Details - Error data  Line Number Nine Details  Row Number Ten Details  Line Number Eleven  Row Number Twelve Details  Line Number Thirteen Details  Row Number Fourteen Details  Line Number Fifteen Details  Row Number Sixteen Details  Line Number Seventeen Details-Error data  Row Number Eighteen Details  Line Number Nineteen Details  Row Number Twenty Details  20 |
| grep -v Seven Input\_data.txt (void) | Line Number One Details  Row Number Two Details  Line Number Three Details  Row Number Four Details  Line Number Five Details  Row Number Six Details  Row Number Eight Details - Error data  Line Number Nine Details  Row Number Ten Details  Line Number Eleven  Row Number Twelve Details  Line Number Thirteen Details  Row Number Fourteen Details  Line Number Fifteen Details  Row Number Sixteen Details-Error data  Row Number Eighteen Details  Line Number Nineteen Details  Row Number Twenty Details  20 |

|  |  |
| --- | --- |
| grep -A 5 Error Input\_data.txt | After the error |

****

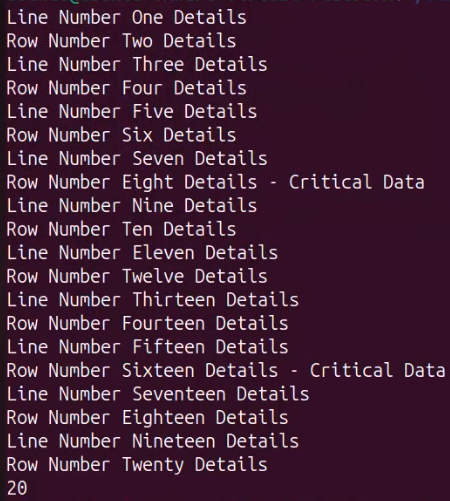
|  |  |
| --- | --- |
| grep -B 5 Error Input\_data.txt | Before the error message |

****

|  |  |
| --- | --- |
| grep -C 5 Error Input\_data.txt | C |

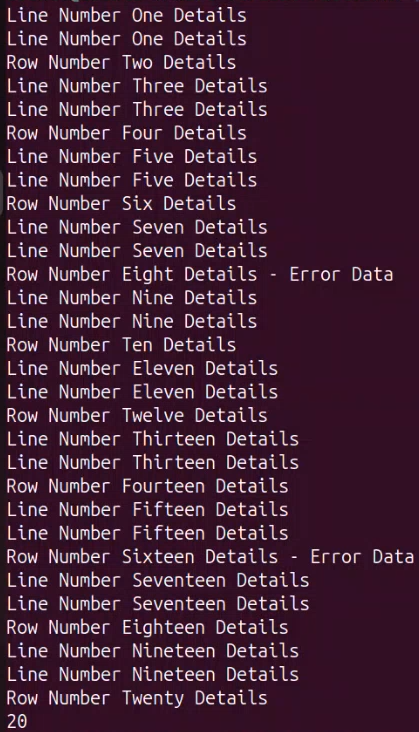
****

|  |  |
| --- | --- |
| sed 's/Error/Critical/' Input\_data.txt  sed 's/Error/Critical/' Input\_data.txt>Input\_data\_sed.txt | stream editor ,s-substitute |

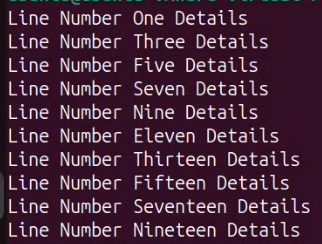
****

|  |  |
| --- | --- |
| -sed cmd Substitute the word with another word  Which the grep cmd unable to do it  -sed can also do functionalities of grep cmd to search  So sed has capability of doing the functionalities of gerp and replacing the word and substituting the word |  |

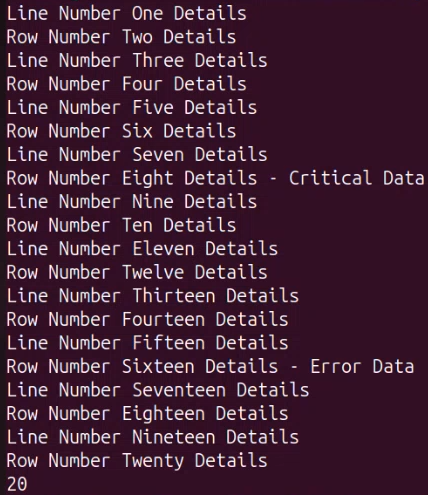
|  |  |
| --- | --- |
| **sed ‘/Line/p’ Input\_data.txt** |  |

****

|  |  |
| --- | --- |
| **sed -n ‘/Line/p’ Input\_data.txt** | **“-n”** |

****

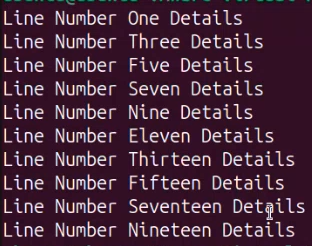
|  |  |
| --- | --- |
| **sed ‘s/Error/Critical’ Input\_data.txt>Input\_data\_sed.txt^C** | **cat Input\_data.txt sed ’8 s/Error/Critical’ Input\_data.txt** |

****

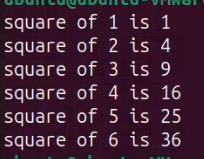
|  |  |
| --- | --- |
| **sed ’8,16 s/Error/Critical’ Input\_data.txt** | 18th and 16th line error to critical |

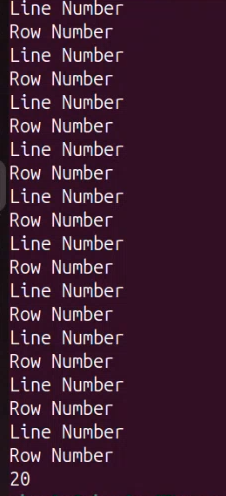
|  |  |
| --- | --- |
| grep "Line" Input\_data.txt  - to print every line in Input\_data.txt, those match the pattern  grep -i line Input\_data.txt  -Case insensitive search  grep -v Seven Input\_data.txt  - to print every line in Input\_data.txt, that do not match the pattern  grep -A 5 Error Input\_data.txt  - To print the five lines after a match  grep -B 5 Seven Input\_data.txt  - to print the five lines before a match  grep -C 5 Seven Input\_data.txt  - to print the five lines before after a match | grep |
| sed 's/Error/Critical/' Input\_data.txt  -s => Substitute  sed '8 s/Error/Critica/' Input\_data.txt  -script runs only on line 8 of Input\_data.txt.  sed '8,16 s/Error/Critical/' Input\_data.txt  - script runs on lines 8 to 16 of Input\_data.txt | sed |
|  |  |

|  |  |
| --- | --- |
| awk | awk ‘/Line/{print}’ Input\_data.txt |

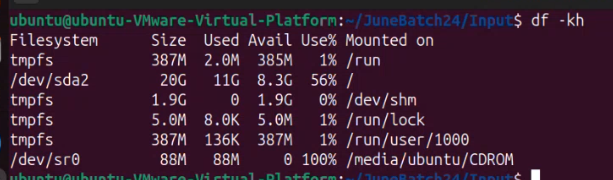


|  |  |
| --- | --- |
| awk “gsub -global substitution” | awk ‘{ gsub (/Error/, Critical )} {print}’ Input\_data.txt |
|  | awk ‘{ gsub (/Error/, “Critical” )} {print}’ Input\_data.txt |
| cut cmd -cutting vertically |  |
| awk ’{print $1, $2}’ Input\_data.txt  cut -d’ ’ -f 1,2 Input\_data.txt | awk -F “ , ” ’{print $1, $2}’Input\_data.txt  awk ’{print $3, $2}’ Input\_data.txt |
| awk 'BEGIN { for(i=1;i<=6;i++) print "Square of", i, "is", i\*i; }' |  |

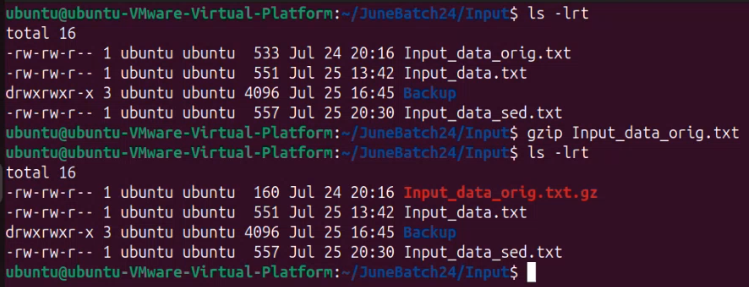




|  |  |
| --- | --- |
| df -kh  df |  |



|  |  |
| --- | --- |
| To zip the file | gzip Input\_data\_orig.txt |



|  |  |
| --- | --- |
| To keep original file and zip file | gzip -k Input\_data\_sed.txt |
| Remove | rm Input\_data\_set.txt.gz |
| How to unzip | gunzip Input\_data\_orig.txt.gz |

---

awk 'BEGIN { for(i=1;i<=6;i++) print "square of", i, "is",i\*i; }'

awk '{print $1, $2}' Input\_data.txt (cut cmd functionality) 6 of 11

awk '{print $3, $2}' Input\_data.txt - sequence can change

cat Input\_data.txt

awk '{gsub(/Error/, Critical)}{print}' Input\_data.txt

cat Input\_data.txt awk '/Line/{print}' Input\_data.txt

awk '/Line/{print}' Input\_data.txt

—

df -kh Check the space consumed by the Fie Systems

gz

gzip

gunzip

gzip Input\_data\_orig.txt

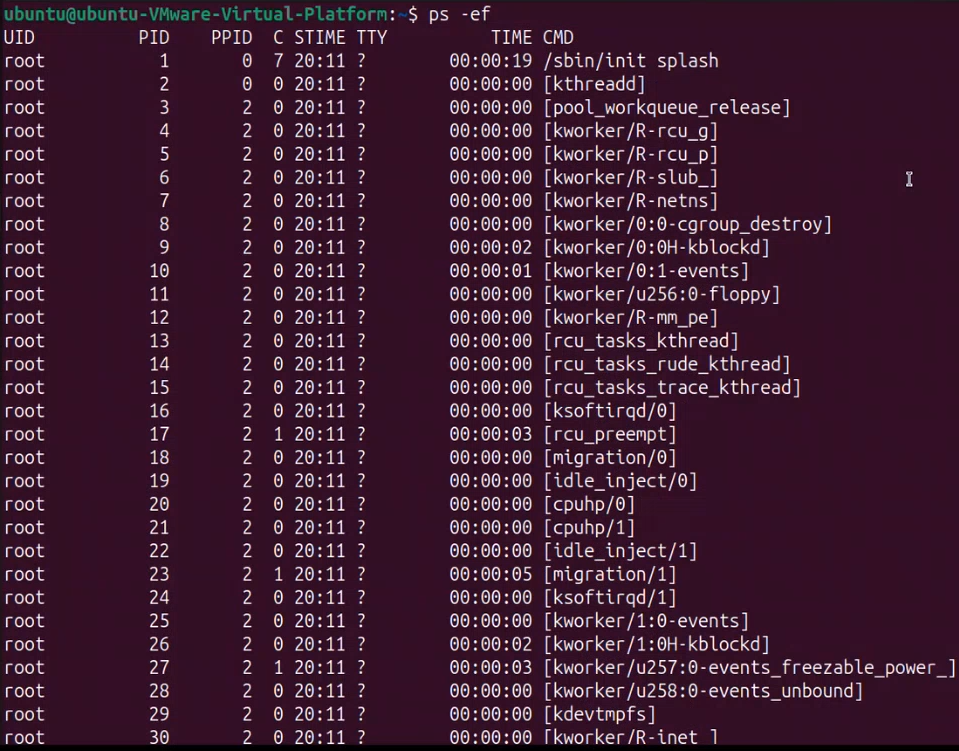
gzip -k Input\_data.txt

gunzip Input\_data\_orig.txt.gz

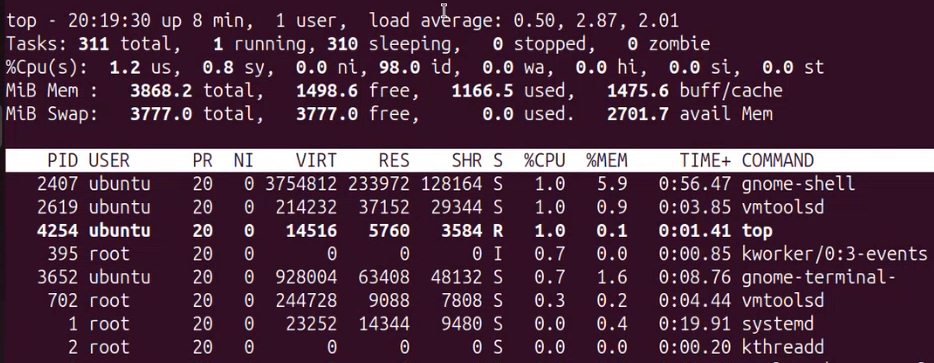
--

**24THCLASS**

|  |  |
| --- | --- |
| uses of ps-ef | -They are static |

****

|  |  |
| --- | --- |
| **Top-** | ongoing variations on task manager on runtime bass |

****