|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Commands | No | Commands | No | Commands |
| 1 | **$ ssh username@servername**  command used to login to server | 2 | **$ pwd**  it prints present working directory | 3 | **$ id**  shows current user's UID, username and GID and group name  **$w**  will display more info about the users logged in |
| 4 | **$ ls -l**  listing the files in present directory  **$** l**s –lrth**  Listing the files with time stamp in reverse order & also it will display the file & directory size in human readable format  **$ls –a**  To display the hidden files  **$ls –i <filename >**  Display the inode no.of file  **$ls –id <dirname>**  Display the inode no.of a directory | 5 | **$ cd ..**  takes you to previous Dir from the existing Directory  **$cd ../..**  Takes you back two directories from current location  **$cd –**  Takes you to very recent working directory | 6 | **$ mkdir <directory>**  will create directory  **$ mkdir -p /home/user1/d1/d2/d3**  For creating nexted directories  **$mkdir –m** <permissons><directoryname>  Giving desired permissions to directory at the time of creation itself |
| 7 | **$ vi <file\_name>**  opens file for reading/editing | 8 | **$ cat <file\_name.txt>**  display contents of file  **$cat > <filename.txt>**  Creates a file & you can add text to file  **$cat >> <filename.txte>**  Inorder to add some text to existing file | 9 | **$ touch <file\_name.txt>**  creates a zero/dummy file |
| 10 | **$ cp <file1> <file2>**  Copy a file | 11 | **$ mv <file1> <file2>**  Move/rename a file or folder | 12 | **$ clear**  clears the screen |
| 13 | **$ rm <file\_name>**  will delete file specified  **$ rm \*.txt**  Delete all the files in the present directory (BE CAREFUL WHILE GIVING THIS COMMAND)  **$rm –rf <dirname>**  Will remeove the directory & its subdirectories recursively | 14 | **$ find / -name \*file\_name/dirname\* 2>/dev/null**  To find out the file or directory in / location if you want in some other location need to specify the location of your need  **$find /opt -size +1G 2>/dev/null**  To findout the files & directories with size greater than 1GB  **$find / -size +1024M -size -2048M 2>/dev/null**  To find out the files & directories within range  **$find /opt -perm 775 2>/dev/null**  To find out the files & directories with specific permissions in the above example it will search for 775 permissions in /opt directory you can give directory & permissions of your choice  **$find /tmp -perm /o=t 2>/dev/null**  To find out the directories with sticky bit permissons, you can serach for setuid & set uid files & directories by giving /u=s for set gid & /g=s for set gid | 15 | **$ grep <pattern/word> file\_name**  checks pattern/word in file name specified  & displays the lines that contains the pattern/word  **$grep –i <pattern/word> filename**  Checks the pattern/word by ignoring the case& displays the lines that contains the pattern/word  **$grep –c <pattern/word> filename**  Will display the count of words/pattern in the given file  **$grep –l <pattern/word> \*.txt**  Will display the list of files containing the word in that directory  **$grep –o <pattern/word> filename**  Will display the only list of words not the entire line where the word/pattern is found |
| 16 | **$ chmod 777 <file/dirname>**  Changes file/directory permissions use –R option switch for recursive( i.e chmod –R permissions will populate the permissons to directory & its sub folders)  r=4 w=2 x=1i.e 4+2+1 is full permissons for above ex:  1st 7represents user permission  2nd 7 represents group  3rd 7 represents others | 17 | . **$ chown owner:group <file\_name>**  changes owner & group for the file\_name  **$ chgrp <groupname> <filename>**  use –R for recursive | 18 | **$ gunzip <file\_name>**  unzips file name  **$ gzip <file\_name>**  zips file\_names |
| 19 | . **$tar -cvzf allfile.tar .gz <**directory&files names which you want to archive&compress using gzip compression>  **$ tar - cvjf allfile.tar.bz2**  **<**directory&files names which you want to archive&compress using bzip2 compression>  **$ tar - cvJf allfile.tar.xz**  **<**directory&files names which you want to archive&compress using bzip2 compression> | 20 | **$tar -xvzf allfile.tar .gz**  Will extract all the contents of allfile.tar.gz  **$ tar -xvjf allfile.tar.b2z**  Will extract all the contents of allfile.tar.gz  **$ tar -xvJf allfile.tar.xz**  Will extract all the contents of allfile.tar.xz  Note: if you want to see the content of tar before uncompressing below is the command  **$tar tf allfile.tar.gz/bz2/xz** | 21 | **$scp –r <file/dir>** **username@servername:path** (to which we want to place on the destination server refer example below)  $ scp –r KT u57015@cgvr0018:/tmp will move the KT directory to /tmp folder in destination server  **$sftp <servername>** once sftp connection is established with remote server go to required path where you want to copy the file by using cd & to place the file/get the file from remote server below are the inputs/command to use  **put** to place the file on remote server  **get** to get the file from there |
| 22 | **$ vi <file\_name>**  opens file for reading/editing  by default when you open a file using vi editor it opens in command mode, in order to search for word  **/word you want to search (need to be in cmd mode)**  In order to go to insert mode press i to come out from insert press esc key.  >if you want to replace a word globally in vi editor come to cmd mode  **:%s/word2breplace/newword**  >for save & quit (needs to be in cmd mode use below  **:wq! With out save to quit use :q!**  ( plz refer vi doc shared for more options) | 23 | **$ crontab -l**  Shows the cron jobs running/scheduled for the current user.  **$ crontab -e**  edit/add entries to present user cronjobs | 24 | **$ more <file/logname>**  displays page by page contents of file/log  (Use space bar to navigate page by page through logfile)  **$less <logfilename>**  displays page by page contents of file/log  (Use space bar to navigate page by page through logfile or up & down arrowsin order to search for word  **/word you want to search (n go to next occurance of word N for previous occurance)** |
| 25 | **$ tail –f <file\_name>**  shows last 10 lines of file  use tail -f for continous update of file\_name | 26 | **$ head <file\_name>**  shows first 10 lines of file\_name  $ head –n (no.of lines) <filename>  Shows the top no.of lines specified by you | 27 | **$ top**  **shows all process and memory, cpu etc utilization**  **$ps –ef | grep <process\_name>** (will show the PID of the process in the 2nd field)    **$kill -9 <PID>**  killing an unwanted process      **$ kill -3 <PID>**  Used to take threaddump of java process |
| 28 | **$ ln file1 file2**  creates link of file1 to file2 | 29 | **$ln –s <Dir1> <Dir2>** | 30 | **$du –h** will display the size of directories & Subdirectories in human readable format  **$du –a** will display the size of files  in human readable format  **$du –ah** will display the size of files & subdirectories of a directory |
| 31 | **$ df**  The “df” command displays the information of device name, total blocks, total disk space, used disk space, available disk space and mount points on a file system.  **$df –h**  It same info as above command but displays the sizes in human readable format.  (if you sue df –k info will displayed in kilo bytes df –m will display in mega bytes) | 32 | **$free**  The free command is used to display the amount of free and used system memory along with swap usage.    **$free –h**  used to display the amount of free and used system memory along with swapusage in a human readable format | 33 | **$ mpstat**  shows multi cpu statistics like load on each cpu |
| 40 | . **$ uname –a**  In order to know the bit processor of you system  It also prints all the system information in the following order: Kernel name, network node hostname, kernel release date, kernel version, machine hardware name, hardware platform, operating system some more information | 41 | **$cat /etc/redhat-release**  will display the Redhat version | 42 | . **$ nohup <cmd\_name> &**  nohup is very useful command. it runs the command even the telnet connection is closed/broken.  & is used for running command in background. |
| 43 | **$sed -n '5,15p' <filename.txt>**  To return a range of lines from 5-15 in a file  **$sed -n -e '9,45p' -e '65,85p' <filename.txt>**  To return two ranges 9-45 & 65-85 from samplesed1.txt file  **$sed -e '5,18d' <filename.txt>**  To view entire file except the give range i.e. 5-18 | 44 | **$sed -i 's/Wordtobereplace/new word/g' <filename.txt>**  To replace word in a file globally(use gi instead of g then the case of the word to be replaced will be ignored)  **$sed G <filename.txt>**  To inserst one empty line after each line  **$sed ‘G;G’ <filename.txt>**  To insert two empty lines after each line | 45 | **$sed 's/ \*/ /g' samplesed1.txt**  To replace multiple blank spaces with a single space  **$ sed '/^#\|^$\| \*#/d' <filename.txt>**  To remove empty lines or those beginning with # from the file useful while editing the apache configuration files |
| 46 | **$ hostname**  will give your system name | 47 | **$ ifconfig**  Will show the ip-address of the system | 48 | **$ netstat**  shows network statistics  (ports that are open on the specific server)  **$netstat –alnnp | grep port**  Display the network stats of the port specified by you |
| 49 | **$ ping <hostname>**  will ping and test connectivity between your system and the hostname you give in the ping.  you can also give ping <ip-address> | 50 | **$Nslookup <ip>**  Display the hostname as per the DNS entries by means of reverse look up  **$NSlookup <url>**  Display the server ip as per the DNS entry by performing forwardlookup | 51 | **$ traceroute <ip/hostname>**  In order to trace the network packet information from source until it reaches the destination.  . |
| 52 | **$ last**  Will show the users logged in/out information  last <user\_name> shows particular user logins/logouts  last reboot shows all the system boots | 53 | **$ uptime**  will show how long the system has been up and also shows cpu load, number of users logged in etc | 54 | **$ init 0**  will shutdown the system  **$ init 6**  will reboot the system (other init options are 1, 2, 3, 5 and S  these are also called as run levels in linux  init1-singleusermode  init2-multiusermode with no NFS  init3-multiusermodewithcli &nogui  init4userdefinable  init5-mutiusermodewithgui,standardrunlevel for most of the Linux based systems  runlevel s- maintanence mode |
| 55 | **$ rpm -qa --last > last-rpms.txt**  Gives/shows info about installed packages/software on system | 56 | . **$ lsof -p <pid>**  List the opened files for the process | 57 | **$ dos2unix <filename>**  Convert dos formatted file to unix format |
| 58 | **$ ksh –x <file name.sh>**  Will compile the shell file line by line | 59 | **$ sleep 5**  waits for 5 seconds (useful in shell scripts) |  |  |