**INTRODUCTION**

**How will you check either the remote server is up or not**

* Syn: Ping hostname/ip

Ex: ping amazon.in

**What is the Command to get the IP**

* ipconfig (windows)
* ifconfig (linux)

**what is the command used for get the ip of the FQDN(Fully qualified domain name)**

* syn: nslookup hostname

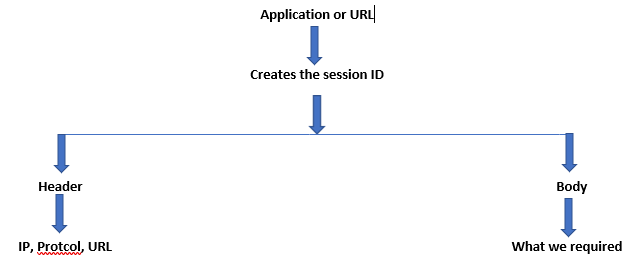
ex : nslookup amazon.in

**How to check the no of hopes to reach the destination**

* syn: tracert hostname

ex: tracert amazon.in

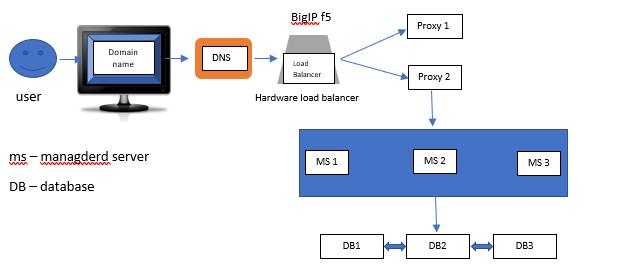
**what contains in url**



**why requests are not mismatch, even multiple users try to access the same application**

* Because each session will be having unique session id.

**How ip will resolve**



**Advantages of unix over windows or why we will use unix?**

1. Unix is more flexible and can be installed on many different types of machines, including main-frame computers, supercomputers and micro-computers.
2. Unix is more stable and does not go down as often as Windows does, therefore requires less administration and maintenance.
3. Unix has greater built-in security and permissions features than Windows.
4. Unix possesses much greater processing power than Windows.
5. Unix is the leader in serving the Web applications. About 90% of the Internet relies on Unix operating systems running Apache, the world's most widely used Web server.
6. Software upgrades from Microsoft often require the user to purchase new or more hardware or prerequisite software. That is not the case with Unix.

**Difference between Unix and Windows**

1. UNIX is more stables and doesn’t go down very often as windows does. Hence unix requires less administration and maintenance.
2. UNIX has greater built in security and permission features comparing to Windows.
3. Comparing to UNIX and Windows, Windows user friendly. And that is why all users prefer windows.
4. UNIX process much greater processing power than Windows. Since UNIX is more powerful and secure, 90% of projects use UNIX operating system.
5. Software upgrades from Microsoft often requires the user to purchase new or more hardware or pre-requisite software, which is not in the case of UNIX.
6. Windows uses ascii, writing the virous is easy. UNIX uses binary format so writing the virous is difficult.

**What is the OS (Operating System)?**

An Operating System (OS) is a software component of a computer system, which:

* 1. Shares the limited resources of the computer
  2. Coordinates and manages the various activities
  3. Offers a number of services to application programs and users
  4. Acts as a host for Applications that are run on the machine

Printer

Monitor

Applications

Disk Drive

Keyboard

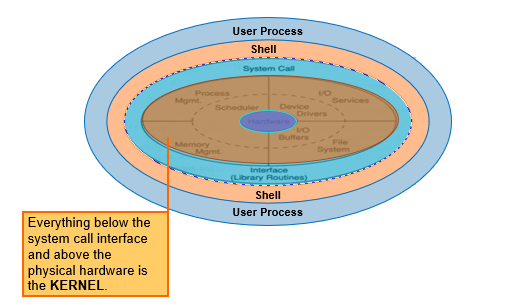
***Operating System***

**What are the features of unix OS?**

1. Multi-user, Multi-tasking, Time sharing system
2. File structure
3. Modularity
4. Input/Output Redirection and Pipes
5. Consistent Peripheral Interface
6. Security
7. Communication and Networking
8. UNIX Utilities and Software Development Tools

**What is UNIX Onion Architecture?**

1. system call -- The system call interface layer converts a process running in user mode to a protected kernel mode process. System call provides a programming interface that allows user programs to access kernel functions.
2. Kernel − The kernel is the heart of the operating system. It interacts with the hardware and most of the tasks like memory management, task scheduling and file management.
3. Shell − shell is the utility that processes your requests. When you type in a command at your terminal, the shell interprets the command and calls the program that you want. The shell uses standard syntax for all commands. C Shell, Bourne Shell and Korn Shell are the most famous shells which are available with most of the Unix variants.
4. Commands and Utilities − There are various commands and utilities which you can make use of in your day to day activities. cp, mv, cat and grep, etc. are few examples of commands and utilities. There are over 250 standard commands plus numerous others provided through 3rd party software. All the commands come along with various options.



**What is Program?**

Program is a program is a set of instructions written to perform a specific task.

**What is process?**

Process is an operation which takes the instructions given and does the manipulation or anything that is instructed in the code itself and provides the output.

**What is file?**

fileis a collection of data

**What is Directory?**

Directory is a folder which contains files, sub-directories and parent directories.

**What is Filesystem?**

File system is a directory, which contains multiple files and directories. It is mounted on disk as a separate partition.

**What are the diff types of files in unix?**

1. Ordinary files: These files contain general data. (.doc and .txt)

2. Directory: it stores both special and ordinary files.

3. Special files: used to represent a real physical device such as a printer, tape drive or terminal.

4. Character special files: A character special file is a file that provides access to an input/output device.

5. Block special files: provide buffered access to hardware devices, and provide some abstraction from their specifics.

**How you will create the empty file?**

Syn: Touch “filename”

Ex: touch abc (to create abc file)

Ex: touch a1 b1 c1 (to create the multiple files at a time like a1, b1 and c1)

**How you will create the directory?**

Syn: mkdir “directoryname”

Ex: mkdir xyz (to create xyz directory)

Ex: mkdir x1 y1 z1 (to create the multiple directory at a time like x1, y1 and z1)

**How will you check the present working directory?**

By using command ‘pwd’ (**p**resent **w**orking **d**irectory)

**How will you check the file and directories in the present working directory?**

By using command ‘ls’ (list)

**How you will check the hidden files in present working directory?**

By using command ‘ls -la’

**What . and .. and .filename?**

. 🡪 represents present working directory

.. 🡪 parent working directory

.filename 🡪 any file name start with . mean hidden file

**How will you list the files/directory ‘s with their creation time at the bottom?**

With the command ‘ls -ltr’

**How will check what are all the process running?**

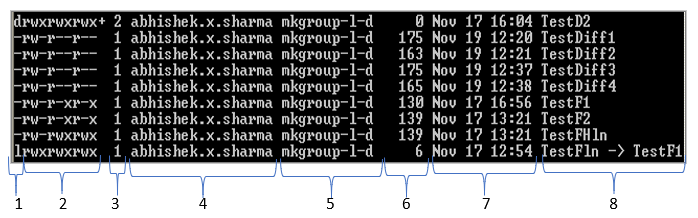
With ‘ps’ command, we can check prosses related to current shell (**p**rocess **s**tatus)

With ‘ps -ef’ command, we can check all the process running

Ps =Process status

e=show all processes, environment variables  
**-f**=full-format

**What are the columns will show in ls -ltr?**



1. l 🡪 link file

- 🡪 ordinary file

d 🡪 directory

1. permissions (chmod command to change the permissions)
2. no of links
3. owner (chown to change the ownership)
4. group
5. size
6. month date and time
7. file/directory name

**ps :** process status

**ps -ef** : it will display all the process of the perticular system

**process** : process is a job which carries given instructions and diplay the output...

**mkdir** : it is used to create directory

mkdir t1

mkdir t2 t3 t4

**ls -la** : it is used to display all files and directories along with that it displays all hidden files

ex: hidden files will be prefixed with .(dot)

**pwd** : this command is used to know the present working directroy

**cd** : it is used to change the directory

cd directory name

**cd** .. : to exit from the directroy

**mkdir** -p dir1/dir2/dir3 : to create parent directories recursively

**which** : it is used to locate the location of the command

syn: which ls

**PATH** : It is a system environment variable which helps to locate pertuclar command to execute

To display PATH environment values

**echo $PATH**

**top** : it is used to display System load average, CPU utilization by users, system, idle time, total memory allocation, utilization, availability and SWAP memory allocation, utilzation, availability along with that it will each and individual process id , owner, Cpu, memory utilization along with that we can find total number of processes, sleeping process, running processes, zombie process

**free -m/-g** : it is used to display physical memory and swap memory allocation, utilization, availability and also shared and buffer size

**logname** : it is used to display logname

**hostname** : it is used to display the hostname

**Redirection**

> - this is for over written

>> - this is for appending the data at the end of the line

**practice:**

echo "where are you going" > v1

**ls -ltr**

**cat :** This command is used to read the file

cat filename

**ln -s** : this command will be used to create softlink

ex:- ln -s sourcefile targetlinkfile

**ln :** this command is used to create hardlink for a file

ln filename hardlinkfilename

**rm -** this command is used to remove the file

rm filename

**what is the differece between softlink and Hard links**

**Symbolic link/softlink** is just a short for a orginal file which can be access and manipulate the original file by using softlink file and the softlink can be span across mulitiple system along with that it will occupy the space in the disk based up the softlink filename. But if you remove the original file you cannot access the data.

Softlink/sybolic link will have its own inode number that is why it will occupy the space in the disk

In **Hardlink t**he original and hadlink file shares same inode number that is the reason hardlink will not occupy the space in the disk and also if you remove the orginal file you can still access the data by using hardlink file since both shares common inode number. But you cannot space hardlink across mulitiple filesystem.

**what is inode?**

whever you creat a file automatically one inode number will created and associated to that perticular file that is uniq for each file. The inode number reprents the location of the file, owner of the file, size, when the file was created, accessed modified all these this it will represent.

Command to find the inode number “ls -i filename”

**getconf LONG\_BIT** : this command is used to know the cpu is 64bit or 32bit

**uname**: this command is used to know the operating system name

**uname** -a: this display os name, hostname, os kernel version and architecture whether it is 32 bit or 64bit

**uname** -m: this shows only architecture also shows whether it is 32 bit or 64bit

**uname -r:** it shows OS kernel version

**who:** This command is used to display how many users logged in to the system with the usernames, their terminals and from which Ip they logged in it will display

**who -b:** this command is used to know when was the system booted

**uptime:** this command will display since how long the system is up and running, how many users are logged in and load average it will display

**whoami:** this command will display from which user you logged in

**su - username:** this command is used to switch to another user

to come out use exit command

**finger :** it is used to display similar to who command but it will display additional information than who command, it displays logged users, idle time, terimal used by the users, login time, user address details and from which ipaddress the user logged in.

**date** : it is used to display current date and time

**netstat -an |grep port(22)** : it is used to know whether the port listening or not along with that we can see socket status such as Listen, established, connecting, connected, waiting, closed, waiting closed

here an refers to alphanumeric

**nslookup :** it is used to resolve from hostname to ip address and from ip address to hostname

**forward lookup :** hostname to ipaddress

**reverse lookup :** ip address to hostname

**where the user information will be stored after creating the user?**

it will be stored in /etc/passwd

**where will the user password store in ?**

/etc/shadow

**filepwconv :** this command is used to synchnorize between passwd file and shadow file ( syn : pwconv (with root user)

**cat /etc/redhat-release** : it is used to know the red hat release version

**cat /proc/meminfo** : it is used to know the complete memory information such as allocated, available, free, cached, buffer, hugepage details,swap memory details if top or free command not available in that case we can use this command as an alternative

**cat /proc/swaps** : it is used to know the swcat ap memory allocation and on which filesystem this perticular swap memory is mounted/

**cat /proc/cpuinfo:** it is used to find out the cpu information

**swapon -s :** it is used to display the partition of swap and allocated size and priority

**swapon -v :** it will display verbose with size in human readable format

**swapon -a :** it is used add the swap space and this will be done through the root user

**cp :** it is used to copy a file or take backup of a file

syn: cp sourcefile(existing file) targetfile(non existing file)

**cp -r :** to copy files and directories recursively

**cp -p :** to copy files with preservative that means while copying the backup file date, time, permissions will not be changed

**cp -i :** it is used to copy or backup a file with interactive mode

Note: you can use as "cp -rpfi" ( here f is for forcefull)

**ls -lc -** This command is used to know file creation date and time

Note:- pratice this command after done cp -p so that you get the required output)

**ls -lSr -** This command is used to display the largest files at the end that means the files size will display ascending order

**ls -lhr :** this command is used to display files and directories size in human readable format.

**cat :** this command is used to read a file

syn :- cat filename

============

**Vi editor**

**insert commands**

**esc + i** - this is used to insert data before the cursor

**esc + a -** this is used to append the data after the cursor

**esc + o -** this is used to insert an empty line below the cursor

**esc + O -** this is used to insert an empty line above the cursor

**esc + :w -** it is used to save the file

**esc + :wq! -** it is used to save and quit forcefully

**esc + :q! -** it is used to quit forcefully without saving the data

**esc ZZ :- this** also is similar to :wq which will be used to save and come out

**Cursor motion commands in vi editor**

**esc + $ :** it is used to move the cursor to end of the line

**esc + ^ :** it is used to move the cursor to starting of the line

**esc + l :** it is used to move the cursor one character forward

**esc + h :** it is used to move the cursor one character backward

**esc + b :** it is used to move the cursor one word backward

**esc + w :** it is used to move the cursor one word forward

**esc + j :** it is used to move the cursor one line downward

**esc + k :** it is used to move the cursor one line upward

**esc + M :** it is used to move the cursor to middle of the page

**esc + G :** it is used to move the cursor to end of the file

**esc + 1G :** it is used to move the cursor to the first line of the file

**esc + J :** it is used to join the below line to the current line as single line

**esc + dd :** it is used to delete a line

esc + ndd

esc + 10dd

**esc + u :** it is used to undo the changes

**esc + yy :** it is used to copy a line (yanking a line or lines)

**esc + nyy :** it is used to copy n number of lines from the cursor

esc + 10yy

**esc + p :** it is used to paste the copies thorugh esc + yy

**esc + :r** filename : it is used to insert the file data to current file

**esc + :sh :** it is used to give you the shell prompt to execute unix commands

**to return to vi editior just type as "exit"**

**esc + :set number :** it is used to set the line numbers

**esc + : set nonumber :** to unset the numbers

**esc + /stringname :** it is used to forward search a given string

syn: esc + /are

**esc + ?stringname :** it is used to backward search a given string

syn: esc + ?are

**esc + :set ic :** it is used to ignore the case

**esc + :%s/source string/target string/g :** it is used to seach and replace the target string in a file globally

syn : esc + :%s/are/were/g

**esc + :%s/.$// :** it is used to remove the ^M characters for a file

or else you can use dos2unix command also to convert file to unix format which helps to remove ^M characters

**more :** it is used to read the data page wise

**less :** it is simlar to more command to read the data page wise but additionally you can move forward and backward to read the data

Note: if you need to comeout from less command you should press "q" character

**view :** it is used to read the data in read only format (it is similar to vi editor but file will open in readble format)

**Note :** To comeout from view command use command esc + :q! (similar to vi command)

**cd directroyname :** to enter to a directroy

**cd .. :** to exit from a directory

**cd ../../../ :** to comeout three parent directories you can use in this way and more you need to comeout use the more "/.."

**cd - :** it is used to bring you to the previous location from the present location

**cd ~ :** it is used to bring you to the user home directroy from wherever the directroy you are

**for cd ~** alternative you can just do the command "cd" and enter

**rm :** it is used to remove a file

rm filename

**rmdir :** it is used to remove empty directroy

rmdir emptydirectory

**rm -r :** it used to remove directroy and files recursively

rm -r directroy ( it removes either empty or non empty directroy recursively along with files)

**rm -rf :** it used to remove directroy and files recursively and forcefully

rm -rf directroy ( it removes either empty or non empty directroy recursively along with files forcefully)

**rm -rfi :** it used to remove directroy and files recursively and forcefully and interactively

**rm -rf directroy** ( it removes either empty or non empty directroy recursively along with files forcefully, interactively)

**mv :** this command is used to move from one location to another location or rename a file or directroy

mv sourcefile/directroy targetfile/directory

ex: mv abc xyz ( to rename)

mv abc /tmp

**Permissions:**

**-rw-(owner) rw- (group) r-- (others)**

**r :** read

**w :** write

**x :** execute

r -4

w -2

x -1

---

7

**rwx rwx rwx**

7 7 7

**rw- rw- rw-**

6 6 6

**umask : 022**

666

022

---

644

**umask :** umask value by default it will be 022, it plays the important role in getting file permissions by default

while creating the file it will get the permissions as **666** but after substracting from umask value 022 it end up with 644,

this umask value we can change with help of system administrator if needed.

Note: whenever we create a directroy by default it will get **777** but it will be substracted from umask value 022 and finally the directroy permission we get 755

**Note :**if you give 777 permissions to a file or to directroy then it is security issue and this type of permssion not recommended and it also called world writable permission

if you want to change globally then change in **/etc/login.def** file with root user else you need to add in to your profile if it is bash shell then the profile name is .bash\_profile

**chmod :** this command is used to change the permissios for a file or a directroy

syn:- chmod 777 file/directory

chmod -R 777 directory

**wild cards**

**\* -** represents all the characters

**? -** it represents single character

To change the ownership , you should login with root user and execute chown command, normal user cannot execute chown command

**chown :** this command is used to change the ownership for a file/files or for a directroy/directroies

syn: chown username:group file/directroy

chown -R username:group file/directroy

chown pooja:oracle filname

chown -R pooja:oracle directoryname

**grep :** it is used to search given string in a file

grep stringname filename

**grep -i :** it is used to ingore the case senstive and search the string in a file

grep -i stringname filename

**grep -v :** it is used to ignore the lines those are containing the given string and display remaining lines

grep -v stringname filename

**grep -c :** it is used to display the count that means the number of times the given string is repeated in a file

grep -c stringname filename

**grep -n :** it is used to display the line number of the occurence given string

grep -n stringname filename

**grep -l :** it is used to display the filenames those are containing the given string

grep -l stringname \*

**1- standard output**

**2- standard error**

**2>/dev/null : here the error will nullified**

**grep -w :** this command is used to search a string which should match with given string and display on the screen

syn:- grep -w stringname filename

ex:- grep -w is filename

**egrep :** this command is used to search multiple strings from a file

syn: egrep 'string1|string2|string3' filename

ex:- egrep 'is|are|were' filename

**grep -f** : it is a file grep which is used to search each line pattern from one file to another file and print if matches

grep -f file1 file2

**df :** it is used to display the disk file system size ( it displays in kilobytes)

**df -k :** it is used to display the file system size in kilo bytes

the output will display as filesystem name , allocated size, utilization size, available size, percentage of utilization, and mount point directories

syn: df -k

**df -h :** it is used to display the file system size in Human readable format meaning is if file size is in GB, TB or kb it display as it is

the output will display as filesystem name , allocated size, utilization size, available size, percentage of utilization, and mount point directories

syn: df -h

**df -h / :** this command it used to display the root (/) filesystem allocation size, utilization size, available size, percentage of utilization and directrory mounted on

you can use for different filesystem as

df -h /

df -h /tmp

df -h /home

**du -** this is command is used to get the disk usage by each file and directory recursively

syn:- du

**du -k** this is command is used to get the disk usage by each file and directory recursively in kilo bytes

syn:- du -k

**du -sk** this is command is used to display total size of file and directory and display in kilo bytes

syn :- du -sk \*

or

du -sk directory or filename

**du -sh** this is command is used to display total size of file and directory and display in human readable format

syn :- du -sh \*

or

du -sh directory or filename

**Compressing the Data**

**zip :** This command is used to compress files and folders

zip filename.zip file

**unzip :** this command is used to extract the zip file

unzip filename.zip

**zipinfo** : to know what are all the files and directories contain inside the zip file

zipinfo filename.zip

**zip -ry :** it is used to compress the files and directories recursively and also it takes symbolic link files to zip file

zip -ry filename.zip files directories files

r - recursively

y - this option used to store symbolic link files

**zip -d :** to delete an entry from zip file

zip -d filename.zip file

**zip -u :** to update zip file with new entry

zip -u filename.zip file

**gzip :** it is used to compres a file it should no work for a directory

gzip filename

zip command works on file/files/directories and recursively it will work

zip file extension name will be filename.zip

gzip will work only on single file

gzip extention name will be filename.gz

**syn: gzip filename**

**gunzip :** it is used to extract the gzip file

**syn: gunzip filename.gz**

how to tansfer files from one server to another server

**SCP : SECURE COPY WHICH IS USED TO TRANSFER FILES FROM ONE SERVER TO ANOTHER SERVER**

1) to copy from source to remote server use the following syntax

scp source\_filename remote\_server\_username@remote\_server\_ip:/remote\_server\_location\_to\_copy

2) to fetch the file from remote server to source server use the following syntax

**scp remote\_server\_username@remote\_server\_ip:/locatation\_of\_the\_file/filename**

**How to archive and take the back of files and directories**

**tar :-** it is used arachive all the files and directories

tar -cvf filename.tar files/directory/\*

c - create

v - verbose

f - file

--------------------------

How to practice this

tar -cvf backup.tar \*

cp backup.tar /tmp

rm -rf \*

ls -ltr

cp /tmp/backup.tar .

tar -xvf backup.tar

ls -ltr

practice end

-------------------------------

To extract the tar file you need to execute the command

tar -xvf backup.tar

To archive files, directories and compress with gz format

tar -zcvf filename.tar \*

tar -zxvf filename.tar ( extract the compressed tar file)

To display the content of the tar file

tar -tvf filename.tar

t - table of content

It is used to delete a file or directory from a tar file

tar -vf backup.tar --delete ab1

**Generate SSH keys to establish trust authentication that means without prompting for password it will login**

it mainly used for automation ( for steps use the soft copy txt file which i shared)

1) generate keys on both the servers

**a) ssh-keygen -t rsa**

this command generate the public key and private key under

user's home directory .ssh in my case it is under /u01/app/orace/.ssh

id\_rsa.pub

id\_rsa

2) once generated the keys on both server create a file "authorized\_keys" under home\_dir/.ssh/ folder

vi authorized\_keys

now add other sever public key in this file

2) exchange the public keys between the servers

3) once done the exchange you can login without password

**How to sort the data by using sort command**

**sort :** this is command is used to sort the data and display the data on the screen into ascending order

sort abc1

**sort -r :** this "r" option will be used to display the data in to decending order

sort -r abc1

**sort -u :** this command with option "u" helps to eliminate duplicate entries and print unique data

sort -u abc1

**sort file1 file2 > newfile :** this command is used to sort the data of file1 and file2 then redirects the data to a new file

sort file1 file2 > newfile

**sort file -o newfile :** this command is used with option "o" to send the output to a new file

sort file -o newfile

**sort -M :**this command is used to display the jumbled months in a way they come in an year

**sort -k 2 filename :** this command is used to sort the file based on second string and display the output with second string alphabetical order like wise if need to sort another just change string number

sort -k 2 filename

uniq command

**uniq :** this command helps to eliminate duplicate entries and print unique data

**sort abc1 |uniq :** this command is similar to sort -u meaning is alternative to sort -u

**uniq -d :** this command is used to display only repeated data but the data should be in ascending order

usage can be done with the combination of sort command

sort filename | uniq -data

if the file already sorted then directly use **uniq -data**

uniq -d filename

**uniq -c :** this command is used to count the number of times a string is repeated but the data has to be sorted in ascending order

**Sed command**

**sed: This command is used to** edit the string in desired and specific lines in a file

**sed :** It is stream editor which is used replace the string in a file

syn: sed 's/are/were/g' filename

{or}

cat filename | sed 's/are/were/g'

**sed -i :** it is used to replace the existing string with new string in a file

sed -i 's/are/were/g' filename

{or}

cat filename | sed -i 's/are/were/g'

**sed -n '1,4p' filename :** to print range of lines from a file

**sed '1d' filename :** to delete 1st line from a file

**sed '1,3d' filename :** it is used to delete range of lines .. with this command it deletes from 1 to 3 lines and print remaining line

syn: sed '1,5d' filename

**sed '1d;2d;5d;6d' filename :** to delete the desired lines

**sed '/^$/d' :** to remove the empty lines from a file

sed '/^$/d' filename

to remove directly in file

sed -i '/^$/d' filename

**sed '1s/sales/manager/' filename :** this command replace the sales string with manager in the first line

**sed '1,5s/sales/manager/' filename:** this command replaces the sales string with manager from 1 to 5 lines range if sales string occurrence

**sed 's/sales/manager/2' filename:** this command is used to replace 2nd time repeated string sales with manager if any lines are having sales string twice

**sed 's/sales/&/' filename:** it is used to replace sales string with sales only

**awk : it is used to scan and print with desired column**

**examples:**

**1) ls -ltr | awk '{print $3 $9}' :** this command print 3rd and 9th column from the ls -ltr output

**2) ls -ltr | awk '{print $3 " " $9}'** this command print 3rd and 9th column from the ls -ltr output and also print the space between the columns

**3) ls -ltr | awk '{print $3 "\t" $9}'** this command print 3rd and 9th column from the ls -ltr output and also print the one tab space(that means 5 characters space) between the columns.

**4) awk '{print $1 "\t" $3}' filename :** this command print 1st and 3rd column from the file

**5) awk -F: '{print $1 "\t" $3}' filename:** this command remove colon delemeter and print print 1st and 3 column from the file

**6) awk -F: '/sales/{print NR,$1, $2}' emp :** this command is used to print 1st and 2nd column from emp file and display those lines containing sales string

**7) awk -F: '/sales/{OFS="-"; print NR,$1,$2}' emp :** to remove colon between the string with -F: and print the out put with hyphen field separator between the columns and also prints line number

NR : number for current records

**grep -A10 -B10 "ERROR" filename :** this command is used to print 10 line before ERROR string occurrence and after ERROR string occurrence

**stat :** this cmmand is used to display status of the file such as owner, group of the file, size, links, inode numner, when it has created, accessed, modified such information it will display

**syn : stat filename**

**head :** it is used to display top 10 lines of the file

**syn: head filename**

**tail :** it is used to display bottom 10 lines of the file

**syn: tail filename**

**tail -f :** it is used to wait for the update at the end of file

**tail -f filename**

Do exercise as follows:

open to terminals (2 puttys) for the same ip and cd to the location where the file is available

in first putty execute the command as "tail -f abc1"

in second putty execute the command as echo "this is for testing" >> abc1

like that you can monitor the footer/updates for that particular file

**paste :** it is used to merge two files

**paste file1 file2**

**find :** it is used to search for a file or for a directory

find

**a) find . -name "filename/directoryname"** ( recommended to use): it is used to search give file or directory from present working directory

**b) find / -name "filename/directoryname" :** it is used to search give file or directory from root directory

**c) find . -name "\*.log" -mtime +30 :** it is used to display 30 days old log files

**find . -type f -name "\*.zip" -mtime +30 :**it is used to search 30 days old zip files

**d) find . -type f -name "\*.log" :** it search only given file from present working directory

**e) find . -type d -nmae "directoryname"** it seraches only given directroy from present working directory

**f) find . -type f -name "\*.log" -mtime +30 -exec rm -rf {} \; :** this command is used to search 30days old log files and remove those log files

**g) find . -type f -name "\*.log" -mtime +30 -exec zip -ry backup.zip {} \;**

**h) find . -exec grep -l "error" {} \;**

**i) find . -mmin -60 :** whatever files are modified 60min.. back it will display

**j) find . -type f -name "\*.log" -maxdepth**

**k) find . -type f -name "\*.log" -mindepth**

**l) find . -type f -name "\*" -perm -g=rw**

**m) find . -type f -name "\*" -perm -g=r**

**n) find . -type f -name "\*" -perm -o=r**

**o) find . -type f -name "\*" -perm -o=x**

**p) find . -type f -name "\*" -perm -g=x**

**find / -maxdepth 3** -name passwd

**maxdepth levels :** Descend at most levels (a non-negative integer) levels of directories below the starting-points. -maxdepth 0 means only apply the tests and actions to the starting-points themselves.

**mindepth levels :** Do not apply any tests or actions at levels less than levels (a non-negative integer). -mindepth 1 means process all files except the starting-points.

**Wc command :** This command is used to display the content of the file in numerical expressions

1) wc test1

2) wc -l test1

3) wc -w test1

4) wc -c test1

5) cat test1 |wc -l

6) cat test1 |wc -w

7) cat test1|wc -c

8) ls -ltr |wc -l

**tr command**

**tr command :** this is used to capitalise the content in the file it just buffer the data and print that to get the out we need to redirect buffered data into new file

**{syn : tr 'a-z' 'A-Z' < filename > new file}**

**9)tr 'a-z' 'A-Z' < filename**

or

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

or

**cat filename|tr "[:lower:]" "[:upper:]"**

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

or

**cat filename|tr "[:upper:]" "[:lower:]"**

**cat filename |tr -cd [:digit:] :** this command removes the characters and prints digits

**cat file name |tr -cd [:alpha:] :** this command removes the numeric and print the characters

**cat filenmae |tr -d [a-v] :** all the characters from a to v range will be deleted from the given file and print on the screen

**diff :** it is used to find the difference between two files

diff file1 file2

**sum :** checksum and count the blocks in a file

syn : sum filename

**cksum :** checksum and count the bytes in a file

syn : cksum filename

PRACTICE -5

#!/bin/bashecho "Enter filename to find exists are not"read fileFPATH="/u01/app/oracle/scripts/Oracle/Day1/operators/Conditional"cd $FPATHif [ -f "$file" ]thenecho "The $file exists but it is a regular file"elif [ -d "$file" ]thenecho "The given $file exists but it is a directory file"

else

echo "$file does not exist"

fi