What is operating system?

- An operating system is software programs which acts as an interface between the user and the hardware.
- It is responsible for execution of all processes,resource allocation,CPU management,file management and many other tasks.
- The purpose of os is providing the environment for the user to execute all his programs in a convenient and efficient manner.

What is a kernel?

- Kernel is the core of the operating system.it establishes the communication between the software and hardware of the system.
- It is also responsible for maintaining system resources.
- Basically it has 4 responsibilities:

- Device Management:

The system is connected to various devices such as CPU,graphics card,etc.These devices data are stored in device drivers present inside the kernel(without kernel we cannot manage these devices)

- Memory Management:

Kernel keeps track of used and unused memory and makes sure the processes should not manipulate the data of each other using virtual memory addresses.

- Process Management:

In process management kernel assigns enough time and makes priorities for processes before handling CPU to other processes.It also deals with security and ownership information.

- Handling system calls:

Handling system call means the programmer can ask the kernel to perform a task.

Linux commands:

date	displays both date and time
cal	displays the calendar with current month
cal [month] [year]	displays the calendar with passed month and year
who	It tells the users who used the system

clear	It clears the terminal screen
ps	It displays the processes
uname -a	It displays all the details of kernel version
pwd	It displays the present working directory
ls	It lists the files present in the directory
cd	It is used to change directory to home directory
cd ..	It is used to go back one directory
ls -l	List the files with its attributes
ls > [file]	Here the greater than symbol is used to store the output to a file
ls >> [file]	Here the greater than symbol is used to store the output to a file by adding in newline(i.e) it appends into a file.
wc	Counting number of lines,words,characters in a file
echo “viki”	It prints the value in console
x=5 echo $x	It displays the value assigned in variable x o/p: 5
echo “$x”	It displays the value assigned in variable x o/p: 5
echo ‘$x’	It displays the value passed in quotes since it is single quotes
mkdir [name]	Creates a directory
rmdir [name]	Removes a directory
cd /	It is used to change directory to root directory
cd De [tab] It takes as cd Desktop	It shows the predicted words present in directory

ls -a	It lists all the files including hidden files in a directory
mkdir [file1] [file2] [file3]	Create a multiple directory
rmdir [file]	It is used to remove an empty directory
rm -rf [file]	Remove the directory if files is present in directory
file [file]	Determines the file type
rm [file]	To remove a file
cp [source] [target]	To copy a file from source to destination
mv [source] [target]	To rename (or) move from source to target file
touch [file]	To create a file
rm *[extension] (i.e) rm *.o Note: * refers to whatever names before .o.it can be implemented whenever based on our requirement If abc.obj Acb.ojb *.o*	To remove all selected extension in a directory
cp -r [source] [target]	It copies the directories along with all sub-directories
cp [file1/dir1] [file2/dir2] [directory]	To copy multiple files/directories in to a directories
cat [file]	Displays the content of file in console
uname -r	Displays kernel version
head [file]	Displays the beginning of file
head -n [file] (i.e) head -3 file.txt head -n3 file.txt head -n 3 file.txt	Displays the no of line you have from the beginning Displays the first 3 lines in a file
head -c [no] [file]	Displays the count of number of

(i.e) head -c 3 file.txt eg:Hello	character in a file o/p:hel

tail [file]	Displays the content of file from last
tail -n [file] (i.e) tail -3 file.txt tail -n3 file.txt tail -n 3 file.txt	Displays the no of line you have from the last Displays the first 3 lines in a file
tail -c [no] [file] (i.e) tail -c 3 file.txt eg:hello	Displays the count of the number of characters in a file. o/p:lo
head -n [no] | tail -n [no] (i.e)head -n 22 | tail -n 12	Display the middle of the text file. (i.e)displays the content between 12 to 22
cut -b n1,n2,n3 [file] (i.e) cut -b 1,3,5 file.txt eg:Tamilnadu	It is used to cut byte by byte from file. It slices specified byte position in file o/p:Tml
cut -b n1-n3,n5-n7 [file] (i.e) cut -b 1-3,5-7 [file] eg:Tamilnadu	It is used to cut bytes at a specified position range o/p:Tamlna
cut -d “ “ -f [no] [file] (i.e)cut -d “ “ -f 1 file.txt eg:Andhra Pradesh	It is used to cut using delimiter with fields (i.e)it is separated with tabs and field represent the lines o/p:Andhra
cut -d “ “ -f 1-3 [file]	It is used to cut using delimiter with fields (i.e)it is separated with tabs and field represent the lines with range between 1 to 3
cut -c [no] [file] (i.e)cut -c 2,3 file.txt	It is used to cut character at column o/p:am

eg:Tamilnadu	
cut -c n1-n3,n5-n7 [file] (i.e) cut -c 1-3,5-7 [file] eg:Tamilnadu	It is used to cut character at column at a specified position range o/p:Tamlna
cut --complement -d “ “ -f [no] [file]	It is the opposite of usual delimiter
cut --complement -c [no] [file]	It is opposite of opposite character
wc -l [file]	Counts only no of lines in a file
wc -w [file]	Counts only no of words in a file
wc -c [file]	Counts only no of characters in a file
Kill -9 [pid]	Kills the process forcefully that is running
whatis	Displays one-line manual page description
tar cvzf [file].tar.gz *.[extension] (i.e) tar cvzf hello.tar.gz *.c It archives every .c file into hello.tar.gz	To create archive a collection of file into a compressed tar file c -creates archive file f- for file name v-displays verbose information z-zip,tells gzip to create tar file
tar xvzf [file].tar.gz *.[extension] (i.e) tar xvzf hello.tar.gz *.c It uzip the archives every .c file into directory from hello.tar.gz	To extract a archive a collection of file from a compressed tar file x- extracts archive file f- for file name v-displays verbose information z-zip,tells gzip to create tar file
crontab -e [Min] [hr] [date of month] [month] [day of month] cmd Eg: * * * * * ls >> pth/file.txt It prints list to file.txt every minute in every hour in every month in every day of month	Running job periodically First by giving this command it asks for editor to implement automation Min - 00-59 Hr- 00-24(24hr format) Date of month-1-31 Month - 1-12(jan-dec) Day of month-0-6 (sun-sat)

Eg: 10 14 3 3 4 ls >> pth/file.txt It prints every ten minutes for 2pm 3rd march wednesday and stores that value to file.txt	

crontab -r	It removes the cron file
crontab -l	Displays the content present in console
find . -name “ ” (i.e) find . -name “*.jar” It finds all files with .jar extension in current directory	It is used to find specific string which is passed with -name flag in current directory(.)
find /home/ -name “*.jar” -print	It is used to find the all .jar extension file in home directory and it print it in console by using -print flag
find . -name “*.jar” -print -exec [cmd] {} \;	It is used to execute linux commands along with find command followed by {} \;
find . -name “*.jar” -mtime -5	It is used to find every file with .jar extension with -mtime says the processes done for past 5 days
sort [file]	It sorts the file content in ascending order
sort -r [file]	It sorts the content of file in reverse order
sort -k [no] [file]	It sorts the content of file in column wise you give
sort -n -k [no] [file]	It sorts the numbers in ascending order in column wise
sort -M [file]	It sorts the content of file month wise
sort -t $’\t’ [file] Here -t is a delimiter	It removes tab separated value before sorting
tr ‘old’ ‘new’ eg:tr ‘[a-z]’ ‘[A-z]’ (i.e) it translates every small letter to capital letter in file	It translates old string content to new string content

tr -s ‘[letter]’ Eg:tr -s ‘l’ Hello o/p:helo	It is used to squeeze the character which we are passing with -s flag inside a file
tr -d ‘[letter]’ Eg:tr -d ‘H’ Hello o/p:ello	It is used to delete a specified letter with the -d flag
grep “string” [file] Eg: grep “unix” readme.txt It displays the unix word that is present in readme.txt file in console	This is used to search the string inside a file
grep -i “string” [file] Eg: grep -i “unix” readme.txt It displays the unix word that is present in readme.txt file in console but it does not care about case of string	This -i flag is used for case-sensitive while searching a file
grep -c “string” [file] Eg: grep -c “unix” readme.txt It displays count of the unix word that is present in readme.txt file in console	The -c flag is used to print the count of string present in a file.
grep -w “string” [file] Eg:grep -w “unix” readme.txt It displays every string and substring unix present in a file	It displays whole string and substring present in a file
grep -l “string” [file/path] Eg:grep -l “string” * o/p:readme.txt It displays every filename that contains unix string in it.	The -l flag is used to find the filename with matching string/pattern

grep -o “string” [file] Eg: grep -o “unix” readme.txt o/p:unix It displays only the string matching in a file on screen.	It displays only the string present in a file that is matching.it does not display entire line as grep command
grep -n “string” [file] Eg: grep -n “unix” readme.txt o/p:1:unix It displays the no of line in which unix is present in readme.txt file	The -n flag displays the no of lines the string matches in a file.
grep -v “string” [file] Eg:grep -v “unix” readme.txt It displays the line where unix string is not matching in readme.txt file	The -v flag is used to find the line that is matching with the string that we pass.
grep -e “string1” -e “string2” [file] Eg: grep -e “unix” -e “Unix” readme.txt It displays multiple input string that matches in a file readme.txt	The -e flag is to check multiple string input that matches in file
grep -r “string” [. , *(current or all)] Eg:grep -r “unix” . This search the pattern that is matching recursively (i.e)it searches current directory and the sub-directory Opposite of grep -l which returns only the file that matches the current directory	It searches the string that matches the current directory and sub-directory recursively
sed ‘s/[old]/[new]/’ [file] Eg:sed ‘s/unix/linux/’ readme.txt S-substitution operation Here it converts word unix to linux inside readme.txt file	It is used to replace old string to new string (i.e) substitution
sed ‘s/[old]/[new]/[no]’ [file] Eg:sed ‘s/unix/linux/2’ readme.txt S-substitution operation	It is used to replace old string to new string (i.e) substitution but it converts the string based on occurence no you give

Here it converts the word unix to linux inside readme.txt file with 2nd occurence no you give	
sed ‘s/[old]/[new]/g’ [file] Eg:sed ‘s/unix/linux/g’ readme.txt S-substitution operation Here it converts the word unix to linux inside readme.txt file with all the occurence of a file	It is used to replace old string to new string (i.e) substitution but it converts the string on all occurrence in file
sed ‘s/[old]/[new]/[no]g’ [file] Eg:sed ‘s/unix/linux/3g’ readme.txt S-substitution operation Here it converts the word unix to linux inside readme.txt file with the 3rd occurence to all other occurence of a file	It is used to replace the nth occurence to all other occurence of string in a file.
sed ‘[no] unix/linux/’ [file] Eg:sed ‘3 s/unix/linux/’ readme.txt It replaces the occurrence of unix to linux in third line of file	It replaces the string in specific line in a file
sed -i ‘s/unix/linux/’ [file]	The -i flag replaces the string in file and makes changes in file also.
sed ‘[no],[no1] s/unix/linux/’ [file] Eg:sed ‘1,3 s/unix/linux/’ readme.txt It replaces the occurrence of unix to linux in range of no of line of file	The range of no of line is replaced in a file
sed ‘[n]d’ [file] Eg: sed ‘3d’ readme.txt It delete the 3rd line in readme.txt file	This command is used to delete the given line in a file.
chmod a+x [file]	It is used to change file permission for all the users (i.e)into executable file
chmod u+x [file]	It is used to change file permission for the current users(i.e)into executable file

chmod o+x [file]	It is used to change file permission for other users(i.e)into executable file
zip [file.zip] [file] Eg: zip file.txt *.txt It zips all the .txt file into file.zip file	It is used to zip the collection of file
unzip [file.zip] Eg:unzip file.zip It unzips the file.zip into a directory	It is used to unzip the file
zip -u [file.zip] [file] Eg:zip -u file.zip test.txt It is used to update the test.txt file in file.zip file	The -u flag updates the file to existing zip file
zip -r [file.zip] [dir] Eg:zip -r file.zip test It is used to recursively add the files to zip files(i.e) it is used to zip file that are present as sub-directory	This -r flag is used to zip files recursively From the sub-directory present in parent directory

unzip [file.zip] -d path/to unzip/ Eg: unzip file.zip -d /home The file is unzipped to home directory	The -d flag helps to unzip the zip file content to another directory.

wget [url]	It is used to download files from the internet. It runs on background without affecting current process
curl [url]	curl is a command line tool to transfer data to or from a server, using any of the supported protocols