LINUX INTERVIEW

Swap Space

Swap space in Linux is used when the amount of physical memory (RAM) is full.
If the system needs more memory resources and the RAM is full, inactive pages in memory are moved to the swap space.
While swap space can help machines with a small amount of RAM, it should not be considered a replacement for more RAM.
Swap space is located on hard drives, which have a slower access time than physical memory.
Swap space file located in /etc/fstab

```
#To view all devices marked as swap in the /etc/fstab file
$ swapon --all
$ dd if=/dev/zero of=/mnt/swapfile bs=1024 count=2097152
$ fallocate --length 2GiB /mnt/swapfile
#And then set the appropriate permissions on the file; make it readable only by root user as follows.
$ chmod 600 /mnt/swapfile
# Now setup the file for swap space with the mkwap command.
$ mkswap /mnt/swapfile
#Next, enable the swap file and add it to the system as a swap file.
$ swapon /mnt/swapfile
#Afterwards, enable the swap file to be mounted at boot time. Edit the /etc/fstab file and add the following line in it.
$ /mnt/swapfile swap swap defaults 0 0
#  verify the swap file was created
$ swapon  -s
or $ cat /proc/swaps
```

************Commands to Check Linux Process Activity******************

```
# Top Command it will show information like tasks, memory, cpu and swap. Press ‘q‘ to quit window.
$ top
# Display Specific User Process
$ top -u tecmint
# atop Command  it will show  processes and also about free and used memory space.
$ atop
$ htop  for interactive mode
```

********** Memory and Disk Spaces***************

```
# Display System Memory
$ free
# Display Memory in Gigabytes
$free -g
# Check File System Disk Space Usage
$ df
# Display Information of all File System Disk Space Usage
$ df -a
# Display Information of File System in GB
$ df -h
# Display File System Type
$ df -T
```

```
# Display File System Inodes
$ df -i
# Display Information of df Command
$ df --help
# disk usage summary of a /home/tecmint directory tree and sub directories.
$ du  /home/tecmint
# disk usage of directory tree along with its subtrees in Megabytes
$ du -mh /home/tecmint
## fdisk stands (for "fixed disk or format disk")

# View all Disk Partitions in Linux
$ fdisk -l
$ fdisk /dev/sda
$ fdisk -s /dev/sda2  ---> Check Size of a Partition in Linux
```

What Are Inodes In Linux?
An inode is a data structure. It defines a file or a directory on the file system and is stored in the directory entry. Inodes point to blocks that make up a file. The inode contains all the administrative data needed to read a file. Every file's metadata is stored in inodes in a table structure.

To check the list of inode numbers, use the following command:

```
$ ls -i
```

***********Hard and Soft Link***********

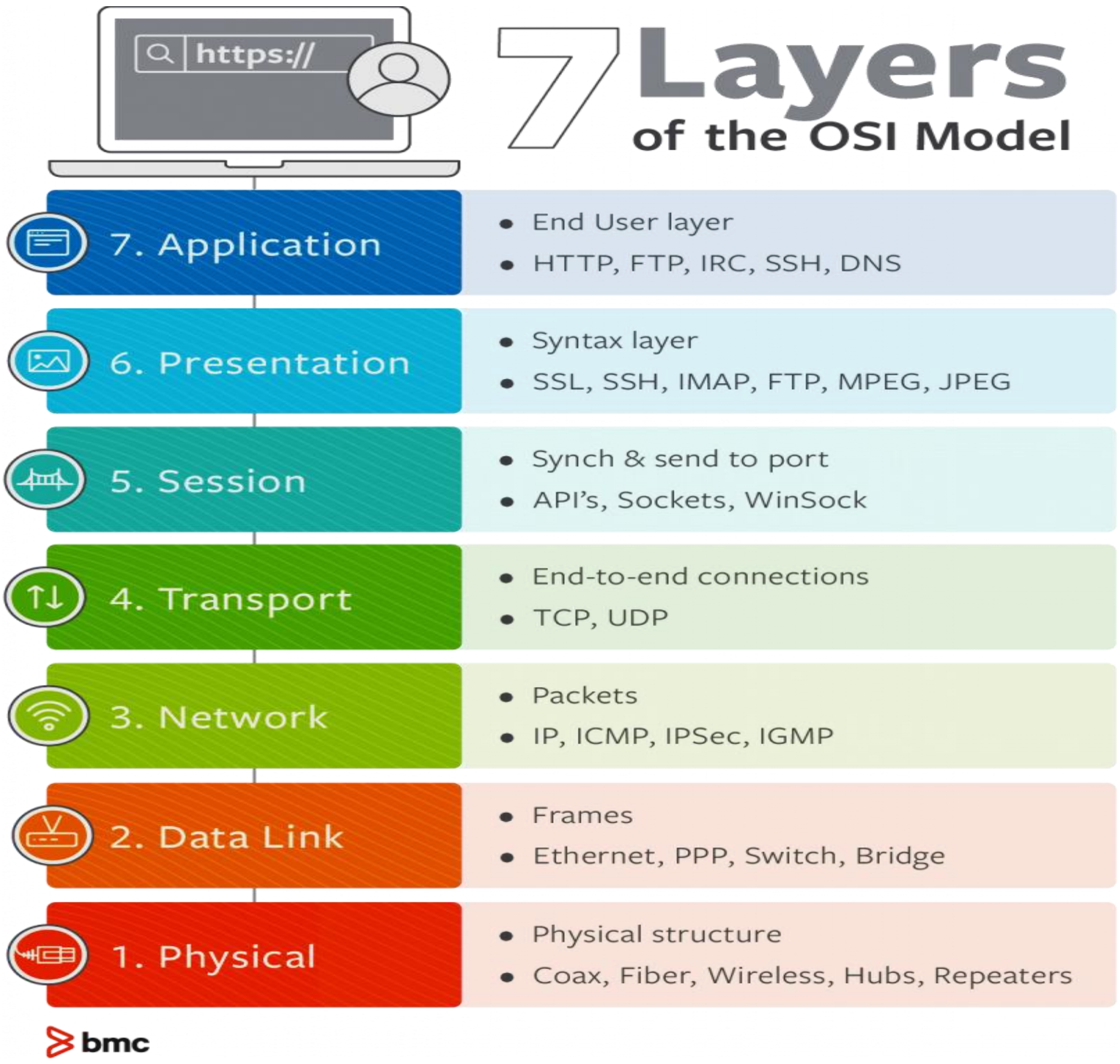
A link in UNIX is a pointer which are pointing to a file or a directory. Creating links is a kind of shortcuts to access a file.
There are two types of links :
1. Soft Link or Symbolic links
2. Hard Links
Ex: For example, if we have a file a.txt. If we create a hard link to the file and then delete the file, we can still access the file using hard link.
But if we create a soft link of the file and then delete the file, we can't access the file

Hard Vs Soft
1. hard linked file is assigned the same Inode value as the original
   soft linked file contains a separate Inode value that points to the original file.
2. Hard Links have actual file contents
    Soft Link contains the path for original file and not the contents
3. We cannot create a hard link for a directory
            A soft link can link to a directory.
4. Hard Link will not be effected even original file is deleted
   Soft link will be effected and become dangle if the original file is deleted

```
#Command to create a hard link is:
$ ln  [original filename] [link name]
Ex:  ln topprocs.sh tp
#Command to create a Soft link is:
$ ln  -s [original filename] [link name]
Ex: ln -s ~/bin/topprocs.sh topps.sh
```

**********OSI layers model ***************

OSI stands for **Open Systems Interconnection**.
It is a 7 layer architecture with each layer having specific functionality to perform.
All these 7 layers work collaboratively to transmit the data from one person to another across the globe.



//Hardware Layers\\

1. **Physical Layer** : The lowest layer of the OSI, responsible for the actual physical connection between the devices responsible for transmitting individual bits from one node to the next.
Ex: Hub, Repeater, Modem, Cables are Physical Layer devices.

2. **Data Link Layer**: Layer is responsible for the node to node delivery of the message.
function of this layer is to make sure data transfer is error-free from one node to another,
Logical Link Control (LLC) & Media Access Control (MAC)
Ex: Switch & Bridge are Data Link Layer devices.

3. **Network Layer**: Network layer for transmission of data from one host to the other in different networks
The sender & receiver's IP address are placed in the header by the network layer.
Routing & Logical Addressing
Ex: Routers
//Hardware Layers\\

//Heart of OSI\\

4. **Transport Layer**: provides services to application layer and takes services from network layer.
It is responsible for the End to End Delivery of the complete message.
Data in the Transport Layer is called as Segments.
** Transport layer is operated by the Operating System.

//Software Layer\\

5. **Session Layer**: This layer is responsible for establishment of connection, maintenance of sessions, authentication and also ensures security.

6. **Presentation Layer**: Presentation layer is also called the Translation layer
The data from the application layer is extracted here and manipulated as per the required format to transmit over the network.
Translation, Encryption/ Decryption, Compression

7. **Application Layer** : At the very top of the OSI. These applications produce the data, which has to be transferred over the network.
Ex: Application – Browsers, Skype Messenger etc.
**Application Layer is also called as Desktop Layer
//Software Layer\\

http://linuxconfig.net/manuals/howto/seven-layers-of-the-osi-model.html

******File system in linux************

Standard Directory Tree
/ ==Root directory—the starting point of the directory tree.

/bin ==Essential binary files, such as commands that are needed by both the system administrator and normal users. Usually also contains the shells, such as Bash.
Contains the basic shell commands that may be used both by root and by other users. These commands include ls, mkdir, cp, mv, rm, and rmdir. /bin also contains Bash,

/boot ==Static files of the boot loader.
Contains data required for booting, such as the boot loader, the kernel
/dev ==Files needed to access host-specific devices.
/etc ==Host-specific system configuration files.
Contains local configuration files that control the operation of programs
/lib ==Essential shared libraries and kernel modules. like DLL files.
/media ==Mount points for removable media.
Contains mount points for removable media, such as CD-ROMs, USB sticks, and digital camera
/mnt ==Mount point for temporarily mounting a file system.
/opt ==Add-on application software packages.
/root ==Home directory for the superuser root.
/sbin ==Essential system binaries.
/srv ==Data for services provided by the system. FTP, HTTP
/tmp ==Temporary files.
/usr ==Secondary hierarchy with read-only data. acronym for UNIX System Resources
This directory contains all application programs. subdirectories, such as /usr/bin, /usr/sbin, /usr/local, and /usr/share/doc.
/usr/bin ## Contains generally accessible programs.
/usr/local ## In this directory, the system administrator can install local, distribution-independent extensions.
/var ==Variable data such as log files

/windows ==Only available if you have both Microsoft Windows* and Linux installed on your system. Contains the Windows data.

/home/username

Holds the private data of every user who has an account on the system. The files located here can only be modified by their owner or by the system administrator.

By default, your e-mail directory and personal desktop configuration are located here.

#wc means word count. Here is the most common usage:

```
$ wc -l  file1  # count lines
$ wc -w file1 # count words
$ wc -c file1 # count characters
$ wc -m file1 # count characters with multibyte support (i.e. emojis count as 1, not as multiple characters)
```

CAT Concatenate files and print to stdout.

```
#>> you can append the content of multiple files into a new file
$ cat file1 file2 >> file3
cat --> display the content of the file
$ cat > <filename> --> to write data into file. after writing press ctrl+data
    ex: cat >1.c
    raja is jsfnganfnfjer.
    ctrl+d
# If file already exists then it get added
$ cat >> <file name>
# copy from source to dest
$ cat <filename> > <filename>
##Every command or exe file should be at /usr/bin/
$ sudo cp <file name> /usr/bin/
## Move a file or rename a file
$ mv <source filename> <dest filename>
## copy a file
$ cp <source filename> <dest filename>
```

```
### remove or delete file or dir ***/*/*/*/
$ rm <file name>  --> remove an empty file
$ rm -f <filename> --> remove non empty file
dir:
$ mkdir <Dir Name>  --> to create a dir
$ rmdir <Dir Name>  --> to remove empty dir
$ rmdir -r <Dir Name>  --> to delete non empty dir
```

1. mnt---> is a default mount dir or copy inode numbers
2. proc---> processors id
3. opt---> 3rd party lib will under this file
4. root---> only super user can enter
5. var---> It contains logs information
6. man ls ---> manual page of ls
7. gcc <filename> -o <filename .exe> ---> to execute a file

**********SSH vs SSL connections************************

SSH means “Secure Shell”. It has a built-in username/password authentication system to establish a connection. It uses Port 22 to perform the negotiation or authentication process for connection.

Read more: Difference between SSH and SSL | Difference Between
http://www.differencebetween.net/technology/difference-between-ssh-and-ssl/#ixzz6PnCTbkFk

SSL means “Secure Sockets Layer”. Many protocols — like HTTP, SMTP, FTP, and SSH ‘were adjusted to include the support of SSL.

Read more: Difference between SSH and SSL | Difference Between
http://www.differencebetween.net/technology/difference-between-ssh-and-ssl/#ixzz6PnCIJsAv

1.Normally, SSH uses port 22 while SSL uses port 443.
2. SSL is used predominantly for securely transmitting critical information like in credit cards and banking. On the other hand, SSH is for securely executing commands across the internet.
3. SSH uses a username/password authentication system to establish a secure connection while SSL does not really bother with it.
4. SSH is more about network tunneling while SSL is more about certificates.

SSH Security Best Practices:
1. Change the Default SSH Port:

```
    Default port 22
  $ vi /etc/ssh/sshd_config
    port 22  --> Change the port number
```

2. Enhance Linux SSH Security Using Key Pairs
To set up SSH keys, you will need to generate a key pair on the client computer
#SSH keygen utility, available with the standard OpenSSH tool.

```
$ ssh-keygen –t rsa
```

Key will be save in default location ~/.ssh with id_rsa and id_rsa.pub
# creating larger 4096-bit keys instead of the default 2048 bits

```
$ ssh-keygen -t rsa -b 4096
```

# Copying a Public Key

```
$ ssh-copy-id username@your_host_address
```

Above command will copy the contents of ~/.ssh/ id_rsa.pub key to the authorized_keys file under the ~/.ssh home directory on the server.
# To test if the authentication with the keys is working

```
$ ssh username@your_host_address
```

3. Disable Server SSH Root Login and Password-Based Logins
Make sure that you have added an account that can gain root access.
#log into the server as root and goto ssh file

```
$ vi /etc/ssh/sshd_config
```

 >>   line that says “PermitRootLogin_yes“ and change to "PermitRootLogin_no"
 >> PasswordAuthentication and change to PasswordAuthentication_no
 >> Add user line like AllowUsers your_username_here
 >> Restart the SSH service

```
$ sudo service ssh restart
```

Now try to login to the server.

Ports and Network ip

# To list all open ports or currently running ports including TCP and UDP in Linux

```
$ netstat -lntu
```

-l – prints only listening sockets   -n – shows port number -t – enables listing of tcp ports
-u – enables listing of udp ports
# Listing TCP Ports connections

$ netstat -at
# Listing UDP Ports connections
$ netstat -au

PORT NUMBER AND NAME
20, 21 -- File Transfer Protocol (FTP)
22 -- Secure Shell (SSH)
23 -- Telnet
25 -- Simple Mail Transfer Protocol (SMTP)
50, 51 -- IPSec
53 -- Domain Name System (DNS)
67, 68 -- Dynamic Host Configuration Protocol (DHCP)
69 -- Trivial File Transfer Protocol (TFTP)
80 -- HyperText Transfer Protocol (HTTP)
110 -- Post Office Protocol (POP3)
119 -- Network News Transport Protocol (NNTP)
123 -- Network Time Protocol (NTP)
135-139 -- NetBIOS
143 -- Internet Message Access Protocol (IMAP4)
161, 162 -- Simple Network Management Protocol (SNMP)
389 -- Lightweight Directory Access Protocol
443 --HTTPS
1433,1434 -- Ms SQL Database
3389 -- Remote Desktop Protocol
5985, 5986 -- WinRM-HTTP, WinRM-HTTPS (Windows Remote Management Service)
8080 -- HTTP

USER MANAGEMENT

Basically in Linux three types of user:
1.**ROOT USER**: Once you install Linux, the Linux administrative root account is created
2.**NORMAL USER**: Regular users have therights to perform basic activities
3.**SERVICE USER**: Services such as Apache, Squid, Mail, databases.
When a user is created in Linux its entry is recorded in the following files.
--> /home:- user default home directory except for root user.
--> /etc/passwd: User account details are contained in this file and have 7 sections.
$ cat /etc/passwd ==> for All Details of User
$ cat /etc/shadow Within this file protected user account details contains the password related info for the user.
*********To create/add a user*************
$ useradd username or $ adduser username
**********To set a password for new and existing users***************
$ passwd username
**********To create a group****************
$ groupadd groupname
Note: While creating a user it also creates a group for users.
***********To add an existing user to a group*******************************
$ usermod -g groupname username
*-g for the primary group
*****************To add new users to the secondary group*****************
$ usermod -G groupname username
***************To set a password for the group**************************

```
$ gpasswd groupname
***********To delete a user.*******************************************
$ userdel username
*******************To delete a user along with its files forcefully.*************
$ userdel -rf username
*************************To delete a group.**************************
$ groupdel -f groupname
***************************To lock user.******************************
$ usermod -L username
$ passwd -l username
*******************To unlock the user.********************************
$ usermod -U username
$ passwd -u username
*******************To change or rename the username.*******************
$ usermod -l login-name old-name
*******************To change user UID.*******************
$ usermod -u UID username
*******************To change or rename the group name.*******************
$ groupmod -n new-group old-group
*******************To change group GID.*****************************
$ groupmod -u GID groupname
*******************To change the user home directory location.****************
$ usermod -m -d /new/path/of/home/directory username
#-m move the content of the old home directory to a new home directory. #-d path of new home directory.
*******************Users must change the password as a logon*******************
$ chage -d 0 username   OR  $ passwd -e username
To set the Minimum Password Age for User.
$ chage -m 7 username
```

CHMOD:

chmod ---> change file mode or permission   X   X   X

1. Group-->g   User   Group   Others
2. User-->u   1   1   1
3. Other-->o   Read   Write Execute
4. All-->a

4 stands for "read",   2 stands for "write",   1 stands for "execute", and   0 stands for "no permission."

Syntax ***: chmod <options> <permissions> <file name>***

Ex: chmod a+x <filename> ---> all have exe permission

    chmod u+w <filename> ---> user have write permission

chown---> used to change owner of a file

ex: chown raja 1.can

Syntax: chown [Options]... NewOwner File...

    chown [Options]... :Group File...

    chown [OWNER] [[:.] [GROUP]]

Find---> To search for a file in a directory

Syntax: find <directory> -name <file type>

Ex: find ./ -name "* .txt" (or)

```
    find ~ -name '*txt'
        ----> this means that search for all txt files in current dir
```

/*/*/*/*/****** **Process Status** *****/*/*/*/*/*/*/

ps --> Process Status
syntax: ps [options]
$ ps au --> to print all active process
$ ps -ef --> To perform a full-format listing
$ ps -ef | more to get page wise search

ps -ef | grep <keyword>
Tail: tail command, as the name implies, print the last N number of data of the given input
Syntax: tail [OPTION]... [FILE]...
EX: $ tail -3 state.txt -->prints last 3 lines
   $ tail +25 state.txt --> prints from 25th line till end

**********whoami/logname/finger*******
$ finger ---> finger looks up and displays information about system users

************Process kill***********
$ pkill <exe file>
$ kill -9 <file id>
**********restart and shutdown***********
shutdown --> init 0
restart --> init 6
$ uname -a ---> To know the linux version

Best Shell Scripting Interview

Q #1) What is Shell?
Ans: Shell is a command interpreter, which interprets the command which the user gives to the kernel. It can also be defined as an interface between a user and operating system.

Q #2) What is Shell Scripting?
Ans: Shell scripting is nothing but series or sequence of UNIX commands written in a plain text file. Instead of specifying one job/command at a time, in shell scripting we give a list of UNIX commands like a to-do list in a file to execute it.

Q #3) What is the Importance of writing Shell Scripts?
Ans: The points given below explain the importance of writing shell scripts.

Shell script takes input from the user, file and displays it on the screen.
Shell scripting is very useful in creating your own commands.
It is helpful in automating some tasks of the day to day life.
It is useful for automating system administration tasks.
Mainly it saves time.

Q #4) List some of the common and most widely used UNIX commands.
Ans: Given below is a list of widely used UNIX Commands.
Command	Example/Usage of Command	Description
ls	1. $ ls
2. $ ls –lrt or $ ls -ltr

1. It lists files in the current directory.
2. It lists files in the long format.

cd 1. $ cd

2. $ cd test
3. $ cd .. (after cd space needs to be given before entering two dots.)
1. It changes directory to your home directory.
2. It changes directory to test.
3. It moves back to one directory or to the parent directory of your current directory.

mkdir $ mkdir test It creates a directory called test.

rmdir $ rmdir test1

CAUTION: Be careful while using this command.

It removes directory test1.

cp 1. $ cp file1 test

2. $ cp file1 file1.bak
1. It copies file1 to test directory.
2. It takes backup of file1.

rm $ rm file1

CAUTION: Be careful while using this command.

It removes or deletes a file1.

mv $ mv file1 file2 It moves or renames file1 to file2.

more $ more It checks or display one page at a time.

touch $ touch test It creates an empty file called test.

cat 1. $ cat File1

2. $ cat test1 > test2
1. It displays contents of File1.
2. It creates a new file test2 with the contents of test1.

compress $ compress file1 It reduces the size of file1 and creates a compressed file called file1.z and deletes file1.

date $ date

e.g. Output:

Tuesday, September 12, 2017 06:58:06 AM MDT

It displays current date and time.

diff $ diff file1 file2 It displays line by line difference between file1 and file2.

find $ find . –name ‘*.t’ -print It searches in the current directory and in all its subdirectories for files ending with .t, and writes their

names in the output.

finger $ finger It displays information about user.

who $ who It lists the users those who are logged in on the machine.

grep 1.$ grep Hello file1

2.$ grep –c Hello file1
1. It searches for the lines containing Hello in file1.
2. It gives count or number of lines that contains Hello in file1.

kill kill

$ kill 1498

It kills the process which is having PID as 1498.

lpr 1.$ lpr –Pprinter1 test

2.$ lp file1
1. It sends file test to print it on printer1.
2. It prints file1.

man $ man ls It displays online manual or help about ls command.

passwd $ passwd It is used to change the password.

pwd $ pwd

e.g. Output: /u/user1/Shell_Scripts_2017

It displays present working directory.

ps $ ps

e.g. Output:

```
PID TTY TIME COMMAND
1498 3b 0:10 sh
1500 3b 0:05 sh
```

It displays the list of processes which are currently running on the machine.

talk $ talk user1 It is used to talk to the user1 who is currently logged into the same machine.

wc $ wc file1

e.g. Output:

```
4 6 42 file1
```

It counts the number of lines, words and characters in file1.

chmod $ chmod 744 file1 It changes the permissions of file1 & assigns this permission rwxr--r--

gzip $ gzip file1 It compresses the file1. After compression file1 should look like this, file1.gz

gunzip $ gunzip file1.gzIt uncompresses the file1.gz. After uncompression file1.gz should look like this, file1

history $ history It lists all the commands which are recently used.

logname $ logname

e.g. Output:

```
user1
```

It prints log name of the user.

uname $ uname

e.g. Output:

```
SunOS
```

It gives information about unix system which you are using.

tty $ tty

e.g. Output:

```
/dev/pts/1
```

It displays the device name of your terminal.

sort $ sort file1 This will sort the contents of file1 and displays sorted output on the screen.

head $ head -15 file1 It displays first 15 lines of the file.

tail $ tail -15 file1 It displays last 15 lines of the file.

Q #5) Shell programs are stored in which file?

Ans: Shell programs are stored in a file called sh.

Q #6) What are the different Types of Shells available?

Ans: There are mainly 4 important types of shells that are widely used.

And they include:

Bourne Shell (sh)

C Shell (csh)

Korn Shell (ksh)

Bourne Again Shell (bash)

Q #7) What are the Advantages of C Shell over Bourne Shell?

Ans: The advantages of C Shell over Bourne Shell are:

C shell allows aliasing of commands i.e. a user can give any name of his choice to the command. This feature is mainly useful when a user has to type the lengthy command again and again. At that point of time, instead of typing a lengthy command a user can type the name that he has given.

C shell provides command history feature. C shell remembers the previously typed command. Thus, it avoids typing the command again and again.

Q #8) In a typical UNIX environment how many kernels and shells are available?

Ans: In a typical UNIX environment, only one kernel and many shells are available.

Q #9) Is separate compiler required for executing a shell program?
Ans: A separate compiler is not required to execute a shell program. The shell itself interprets the command in the shell program and executes them.

Q #10) How many shell scripts come with UNIX operating system?
Ans: There are approximately 280 shell scripts that come with the UNIX operating system.

Q #11) When should shell programming/scripting not be used?
Ans: Generally, shell programming/scripting should not be used in the below instances.
When the task is very much complex like writing the entire payroll processing system.
Where there is a high degree of productivity required.
When it needs or involves different software tools.

Q #12) Basis of shell program relies on what fact?
Ans: The basis of shell programming relies on the fact that UNIX shell can accept commands not just only from the keyboard but also from a file.

Q #13) What are the default permissions of a file when it is created?
Ans: 666 i.e. rw-rw-rw- is the default permission of a file when it is created.

Q #14) What can be used to modify File permissions?
Ans: File permissions can be modified using umask.

Q #15) How to accomplish any task via shell script?
Ans: Any task can be accomplished via shell script at the dollar ($) prompt and vice versa.

Q #16) What are Shell Variables?
Ans: Shell variables are the main part of shell programming or scripting. They mainly provide the ability to store and manipulate information within a shell program.

Q #17) What are the two types of Shell Variables? Explain in brief.
Ans: The two types of shell variables are:

#1) Unix Defined Variables or System Variables – These are standard or shell defined variables. Generally, they are defined in CAPITAL letters.
Example: SHELL – This is a Unix Defined or System Variable, which defines the name of the default working shell.

#2) User Defined Variables – These are defined by users. Generally, they are defined in lower letters
Example: $ a=10 –Here the user has defined a variable called 'a' and assigned value to it as 10.

Q #18) How are shell variables stored? Explain with a simple example.
Ans: Shell variables are stored as string variables.
Example: $ a=10
In the above statement a=10, the 10 stored in 'a' is not treated as a number, but as a string of characters 1 and 0.

Q #19) What is the lifespan of a variable inside a shell script?
Ans: The lifespan of a variable inside shell script is only until the end of execution.

Q #20) How to make variables as unchangeable?

Ans: Variables can be made unchangeable using readonly. For instance, if we want variable a value to remain as 10 and not to be changed then we can achieve this using readonly.
Example:

```
$ a=10
$ readonly a
```

Q #21) How variables can be wiped out?
Ans: Variables can be wiped out or erased using the unset command.
Example:

```
$ a =20
$ unset a
```

Upon using the above command the variable ‘a’ and its value 20 get erased from shell’s memory.
CAUTION: Be careful while using this unset command.

Q #22) What are positional parameters? Explain with an example.
Ans: Positional parameters are the variables defined by a shell. And they are used whenever we need to convey information to the program. And this can be done by specifying arguments at the command line.
There are totally 9 positional parameters present i.e. from $1 to $9.
Example: $ Test Indian IT Industry has grown very much faster
In the above statement, positional parameters are assigned like this.
$0 -> Test (Name of a shell program/script)
$1 ->Indian
$2 -> IT and so on.

Q #23) What does the. (dot) indicate at the beginning of a file name and how should it be listed?
Ans: A file name which begins with a .(dot) is called as a hidden file. Whenever we try to list the files it will list all the files except hidden file.
But it will be present in the directory. And to list the hidden file we need to use –a option of ls. i.e. $ ls –a.

Q #24) Generally, each block in UNIX is how many bytes?
Ans: Generally, each block in UNIX is of 1024 bytes.

Q #25) By default, a new file and a new directory which is being created will have how many links?
Ans: New file contains one link. And a new directory contains two links.

Q #26) Explain about file permissions.
Ans: There are 3 types of file permissions as shown below:

Permissions	Weight
r – read	4
w – write	2
x - execute	1

The above permissions are mainly assigned to owner, group and to others i.e. outside the group. Out of 9 characters first set of 3 characters decides/indicates the permissions which are held by the owner of a file. The next set of 3 characters indicates the permissions for the other users in the group to which the file owner belongs to.
And the last 3 set of characters indicate the permissions for the users who are outside the group. Out of the 3 characters belonging to each set, the first character indicates the “read” permission, the second character indicates “write” permission and the last character indicates “execute” permission.
Example: $ chmod 744 file1
This will assign the permission rwxr–r–to file1.

Q #27) What is a file system?
Ans: The file system is a collection of files which contain related information of the files.

Q #28) What are the different blocks of a file system? Explain in brief.

Ans: Given below are the main 4 different blocks available on a file system.

File System

Block No.	Name of the Block
1st Block	Boot Block
2nd Block	Super Block
3rd Block	Inode Table
4th Block	Data Block

Super Block: This block mainly tells about a state of the file system like how big it is, maximum how many files can be accommodated etc.

Boot Block: This represents the beginning of a file system. It contains bootstrap loader program, which gets executed when we boot the host machine.

Inode Table: As we know all the entities in a UNIX are treated as files. So, the information related to these files are stored in an Inode table.

Data Block: This block contains the actual file contents.

Q #29) What are the three different security provisions provided by UNIX for a file or data?

Ans: Three different security provisions provided by UNIX for a file or data are:

It provides a unique user id and password to the user, so that unknown or unauthorized person should not be able to access it.

At file level, it provides security by providing read, write & execute permissions for accessing the files.

Lastly, it provides security using file encryption. This method allows encoding a file in an unreadable format. Even if someone succeeds in opening a file, but they cannot read its contents until and unless it is decrypted

Q #30) What are the three editors available in almost all the versions of UNIX?

Ans: The three editors are ed, ex & vi.

Q #31) What are the three modes of operation of vi editor? Explain in brief.

Ans: The three modes of operation of vi editors are,

(i) Command Mode: In this mode, all the keys pressed by a user are interpreted as editor commands.

(ii) Insert Mode: This mode allows for insertion of a new text and editing of an existing text etc.

(iii) The ex-command Mode: This mode allows a user to enter the commands at a command line.

Q #32) What is the alternative command available to echo and what does it do?

Ans: tput is an alternative command to echo.

Using this, we can control the way in which the output is displayed on the screen.

Q #33) How to find out the number of arguments passed to the script?

Ans: The number of arguments passed to the script can be found as shown below.

```
echo $ #
```

Q #34) What are control instructions and how many types of control instructions are available in a shell? Explain in brief.

Ans: Control Instructions are the ones, which enable us to specify the order in which the various instructions in a program/script are to be executed by the computer. Basically, they determine a flow of control in a program.

There are 4 types of control instructions that are available in a shell.

Sequence Control Instruction – This ensures that the instructions are executed in the same order in which they appear in the program.

Selection or Decision Control Instruction – It allows the computer to take a decision as to which instruction is to be executed next.

Repetition or Loop Control Instruction – It helps a computer to execute a group of statements repeatedly.

Case-Control Instruction – This is used when we need to select from several alternatives.

Q #35) What are Loops and explain three different methods of loops in brief?
Ans: Loops are the ones, which involve repeating some portion of the program/script either a specified number of times or until a particular condition is being satisfied.
3 methods of loops are:
For loop – This is the most commonly used loop. For loop allows specifying a list of values which the control variable in the loop can take. The loop is then executed for each value mentioned in the list.
While loop – This is used in a program when we want to do something for a fixed number of times. While loop gets executed till it returns a zero value.
Until loop – This is similar to while loop except that the loop executes until the condition is true. Until loop gets executed at least once till it returns a non-zero value.

Q #36) What is IFS?
Ans: IFS stands for Internal Field Separator. And it is one of the system variables. By default, its value is space, tab, and a new line. It signifies that in a line where one field or word ends and another begins.

Q #37) What is a Break statement and what is it used for?
Ans: The break is a keyword and is used whenever we want to jump out of a loop instantly without waiting to get back to the control command.
When the keyword break is encountered inside any loop in the program, control will get passed automatically to the first statement after a loop. A break is generally associated with an if.

Q #38) What is Continue statement and what is it used for?
Ans: Continue is a keyword and is used whenever we want to take the control to the beginning of the loop, by passing the statements inside the loop which have not yet been executed.
When the keyword continue is encountered inside any loop in the program, control automatically passes to the beginning of a loop. Continue is generally associated with an if.

Q #39) What are Metacharacters in a shell? Explain with some examples.
Ans: Metacharacters are special characters in a program or data field which provides information about other characters. They are also called as, regular expressions in a shell.
Example:
ls s* – It lists all the files beginning with character 's'.
Execution over Shell Interpreter/Editor
Meta Characters Script
Output:
Meta Characters Output
$ cat script1 > script2 – Here output of cat command or script1 will go to a script2.
Execution over Shell Interpreter/Editor
Meta Character Script
Output:
Meta Character Output
$ ls; who – This will execute ls first and then who.

Execution over Shell Interpreter/Editor
Meta Character Scripts
Output:
Meta Character Scripts Output
Multiple script

Q #40) How to execute multiple scripts? Explain with an example.
Ans: In a shell, we can easily execute multiple scripts i.e. one script can be called from the other. What we have to do is, we need to mention the name of a script to be called when we want to invoke it.

Example: In the below program/script upon executing the first two echo statements of script1, shell script executes script2. Once after executing script2, the control comes back to script1 which executes a pwd command and then terminates.
Code for script1
Multiple script Code
Code for script2
Multiple Script
Execution of script1 over Shell Interpreter/Editor
Multiple Script Code
Output displayed on the Editor upon executing script1
Multiple Script Code Output

Q #41) Which command needs to be used to know how long the system has been running?
Ans: uptime command needs to be used to know how long the system has been running.
Example: $ uptime
Upon entering the above command at shell prompt i.e. $ uptime, the output should look something like this.

```
9:21am  up 86 day(s), 11:46,  3 users,  load average: 2.24, 2.18, 2.16
```

Execution over Shell Interpreter/Editor
System Runnning Code
Output:
System Runnning output

Q #42) How to find the current shell which you are using?
Ans: We can find the current shell what we are using with echo $SHELL.
Example: $ echo $SHELL
Execution over Shell Interpreter/Editor
Current Shell Script
Output:
Current Shell output

Q #43) How to find all the available shells in your system?
Ans: We can find all the available shells in our system with $ cat /etc/shells.
Example: $ cat /etc/shells
Execution over Shell Interpreter/Editor
Available Shell Script
Output:
Available Shell output

Q #44) How to read keyboard inputs in shell scripts?
Ans: Keyboard inputs can be read in shell scripts as shown below,
Script/Code
Keyword Input Script
Execution over Shell Interpreter/Editor
Keyword Input Script
Output:
Keyword Input output

Q #45) How many fields are present in a crontab file and what does each field specify?
Ans: The crontab file has six fields. The first five fields tell cron when to execute the command: minute(0-59), hour(0-23), day(1-31), month(1-12), and day of the week(0-6, Sunday = 0).
And the sixth field contains the command to be executed.

Q #46) What are the two files of crontab command?

Ans: Two files of crontab command are:

cron.allow – It decides which users need to be permitted from using crontab command.

cron.deny – It decides which users need to be prevented from using crontab command.

Q #47) What command needs to be used to take the backup?

Ans: tar is the command which needs to be used to take the backup. It stands for tape archive. The tar command is mainly used to save and restore files to and from an archive medium like tape.

Q #48) What are the different commands available to check the disk usage?

Ans: There are three different commands available to check the disk usage.

And they are:

df – This command is used to check the free disk space.

du – This command is used to check the directory wise disk usage.

dfspace – This command is used to check the free disk space in terms of MB.

Q #49) What are the different communication commands available in Unix/shell?

Ans: Basically, there are 4 different communication commands available in Unix/shell. And they are mail, news, wall & motd.

Q #50) How to find out the total disk space used by a specific user, say for example username is John?

Ans: The total disk space used by John can be found out as shown below.

```
du –s/home/John
```

Q #51) What is Shebang in a shell script?

Ans: Shebang is a # sign followed by an exclamation i.e. !. Generally, this can be seen at the beginning or top of the script/program. Usually, a developer uses this to avoid repetitive work. Shebang mainly determines the location of the engine which is to be used in order to execute the script.

Here '#' symbol is called as hash and '!' is called a bang.

Example: #!/bin/bash

The above line also tells which shell to use.

Q #52) What is the command to be used to display the shell's environment variables?

Ans: Command to be used to display the shell's environment variables is env or printenv.

Q #53) How to debug the problems encountered in shell script/program?

Ans: Though generally it depends on the type of problem encountered. Given below are some common methods used to debug the problems in the script.

Debug statements can be inserted in the shell script to output/display the information which helps to identify the problem.

Using "set -x" we can enable debugging in the script.

Q #54) How to know the variable length?

Ans: Variable length can be checked as shown below

```
$ {#variable}
```

Q #55) What is the difference between = and ==?

Ans: = -> This is used for assigning value to the variable.

== -> This is used for string comparison.

Q #56) How to open a read-only file in Unix/shell?

Ans: Read-only file can be opened as shown below:

```
vi –R <File Name>
```

Q #57) How can the contents of a file inside jar be read without extracting in a shell script?
Ans: The contents of the file inside a jar can be read without extracting in a shell script as shown below.
tar –tvf <File Name>.tar

Q #58) What is the difference between diff and cmp commands?
Ans: diff – Basically, it tells about the changes which need to be made to make files identical.
cmp – Basically it compares two files byte by byte and displays the very first mismatch.

Q #59) Explain in brief about sed command with an example.
Ans: sed stands for stream editor. And it is used for editing a file without using an editor. It is used to edit a given stream i.e. a file or input from a pipeline.
Syntax: sed options file
Here 's' command present in sed will replace string Hello with Hi.

Q #60) Explain in brief about awk command with an example.
Ans: awk is a data manipulation utility or command. Hence, it is used for data manipulation.
Syntax: awk options File Name
$0 -> For whole line (e.g. Hello John)
$1 -> For the first field i.e. Hello
$2 -> For the second field
Execution over Shell Interpreter/Editor

----

http://www.linux-tutorial.info/modules.php?name=MContent&obj=topics

1) What is Linux?
Linux is an operating system based on UNIX and was first introduced by Linus Torvalds. It is based on the Linux Kernel and can run on different hardware platforms manufactured by Intel, MIPS, HP, IBM, SPARC, and Motorola. Another popular element in Linux is its mascot, a penguin figure named Tux.

2) What is the difference between UNIX and LINUX?
Unix originally began as a propriety operating system from Bell Laboratories, which later on spawned into different commercial versions. On the other hand, Linux is free, open source and intended as a non-propriety operating system for the masses.

3) What is BASH?
BASH is short for Bourne Again SHell. It was written by Steve Bourne as a replacement to the original Bourne Shell (represented by /bin/sh). It combines all the features from the original version of Bourne Shell, plus additional functions to make it easier and more convenient to use. It has since been adapted as the default shell for most systems running Linux.

4) What is Linux Kernel?
The Linux Kernel is a low-level systems software whose main role is to manage hardware resources for the user. It is also used to provide an interface for user-level interaction.

5) What is LILO?

LILO is a boot loader for Linux. It is used mainly to load the Linux operating system into main memory so that it can begin its operations.

6) What is a swap space?
Swap space is a certain amount of space used by Linux to temporarily hold some programs that are running concurrently. This happens when RAM does not have enough memory to hold all programs that are executing.

7) What is the advantage of open source?
Open source allows you to distribute your software, including source codes freely to anyone who is interested. People would then be able to add features and even debug and correct errors that are in the source code. They can even make it run better and then redistribute these enhanced source code freely again. This eventually benefits everyone in the community.

8) What are the basic components of Linux?
Just like any other typical operating system, Linux has all of these components: kernel, shells and GUIs, system utilities, and an application program. What makes Linux advantageous over other operating system is that every aspect comes with additional features and all codes for these are downloadable for free.

9) Does it help for a Linux system to have multiple desktop environments installed?
In general, one desktop environment, like KDE or Gnome, is good enough to operate without issues. It's all a matter of preference for the user, although the system allows switching from one environment to another. Some programs will work in one environment and not work on the other, so it could also be considered a factor in selecting which environment to use.

10) What is the basic difference between BASH and DOS?
The key differences between the BASH and DOS console lie in 3 areas:
- BASH commands are case sensitive while DOS commands are not;
- Under BASH, / character is a directory separator and \ acts as an escape character. Under DOS, / serves as a command argument delimiter and \ is the directory separator
- DOS follows a convention in naming files, which is 8 character file name followed by a dot and 3 characters for the extension. BASH follows no such convention.

11) What is the importance of the GNU project?
This so-called Free software movement allows several advantages, such as the freedom to run programs for any purpose and freedom to study and modify a program to your needs. It also allows you to redistribute copies of software to other people, as well as the freedom to improve software and have it released for the public.

12) Describe the root account.
The root account is like a systems administrator account and allows you full control of the system. Here you can create and maintain user accounts, assigning different permissions for each account. It is the default account every time you install Linux.

13) What is CLI?
CLI is short for Command Line Interface. This interface allows the user to type declarative commands to instruct the computer to perform operations. CLI offers greater flexibility. However, other users who are already accustomed to using GUI find it difficult to remember commands including attributes that come with it.

14) What is GUI?
GUI, or Graphical User Interface, make use of images and icons that users click and manipulate as a way of communicating with the computer. Instead of having to remember and type commands, the use of graphical elements makes it easier to interact with the system, as well as adding more attraction through images, icons, and colors.

15) How do you open a command prompt when issuing a command?

To open the default shell (which is where the command prompt can be found), press Ctrl-Alt-F1. This will provide a command line interface (CLI) from which you can run commands as needed.

16) How can you find out how much memory Linux is using?

From a command shell, use the "concatenate" command: cat /proc/meminfo for memory usage information. You should see a line starting something like Mem: 64655360, etc. This is the total memory Linux thinks it has available to use.

You can also use commands
free - m
vmstat
top
htop
to find current memory usage

17) What is a typical size for a swap partition under a Linux system?

The preferred size for a swap partition is twice the amount of physical memory available on the system. If this is not possible, then the minimum size should be the same as the amount of memory installed.

18) What are symbolic links?

Symbolic links act similarly to shortcuts in Windows. Such links point to programs, files or directories. It also allows you instant access to it without having to go directly to the entire pathname.

19) Does the Ctrl+Alt+Del key combination work on Linux?

Yes, it does. Just like Windows, you can use this key combination to perform a system restart. One difference is that you won't be getting any confirmation message and therefore, a reboot is immediate.

20) How do you refer to the parallel port where devices such as printers are connected?

Whereas under Windows you refer to the parallel port as the LPT port, under Linux you refer to it as /dev/lp . LPT1, LPT2 and LPT3 would therefore be referred to as /dev/lp0, /dev/lp1, or /dev/lp2 under Linux.

21) Are drives such as hard drive and floppy drives represented with drive letters?

No. In Linux, each drive and device have different designations. For example, floppy drives are referred to as /dev/fd0 and /dev/fd1. IDE/EIDE hard drives are referred to as /dev/hda, /dev/hdb, /dev/hdc, and so forth.

22) How do you change permissions under Linux?

Assuming you are the system administrator or the owner of a file or directory, you can grant permission using the chmod command. Use + symbol to add permission or – symbol to deny permission, along with any of the following letters: u (user), g (group), o (others), a (all), r (read), w (write) and x (execute). For example, the command chmod go+rw FILE1.TXT grants read and write access to the file FILE1.TXT, which is assigned to groups and others.

23) In Linux, what names are assigned to the different serial ports?

Serial ports are identified as /dev/ttyS0 to /dev/ttyS7. These are the equivalent names of COM1 to COM8 in Windows.

24) How do you access partitions under Linux?

Linux assigns numbers at the end of the drive identifier. For example, if the first IDE hard drive had three primary partitions, they would be named/numbered, /dev/hda1, /dev/hda2 and /dev/hda3.

25) What are hard links?

Hard links point directly to the physical file on disk, and not on the pathname. This means that if you rename or move the original file, the link will not break since the link is for the file itself, not the path where the file is located.

26) What is the maximum length for a filename under Linux?

Any filename can have a maximum of 255 characters. This limit does not include the path name, so therefore the entire pathname and filename could well exceed 255 characters.

27)What are filenames that are preceded by a dot?
In general, filenames that are preceded by a dot are hidden files. These files can be configuration files that hold important data or setup info. Setting these files as hidden makes it less likely to be accidentally deleted.

28) Explain virtual desktop.
This serves as an alternative to minimizing and maximizing different windows on the current desktop. Using virtual desktops can clear the desktop when you can open one or more programs. Rather than minimizing/restoring all those programs as needed, you can simply shuffle between virtual desktops with programs intact in each one.

29) How do you share a program across different virtual desktops under Linux?
To share a program across different virtual desktops, in the upper left-hand corner of a program window look for an icon that looks like a pushpin. Pressing this button will "pin" that application in place, making it appear in all virtual desktops, in the same position onscreen.

30) What does a nameless (empty) directory represent?
This empty directory name serves as the nameless base of the Linux file system. This serves as an attachment for all other directories, files, drives, and devices.

31) What is the pwd command?
The pwd command is short for print working directory command.
Example:

```
pwd
Output: /home/guru99/myDir
```

32) What are daemons?
Daemons are services that provide several functions that may not be available under the base operating system. Its main task is to listen for service request and at the same time to act on these requests. After the service is done, it is then disconnected and waits for further requests.

33) How do you switch from one desktop environment to another, such as switching from KDE to Gnome?
Assuming you have these two environments installed, just log out from the graphical interface. Then at the login screen, type your login ID and password and choose which session type you wish to load. This choice will remain your default until you change it to something else.

34) What are the kinds of permissions under Linux?
There are 3 kinds of permissions under Linux:- Read: users may read the files or list the directory- Write: users may write to the file of new files to the directory- Execute: users may run the file or lookup a specific file within a directory

35) How does case sensitivity affect the way you use commands?
When we talk about case sensitivity, commands are considered identical only if every character is encoded as is, including lowercase and uppercase letters. This means that CD, cd, and Cd are three different commands. Entering a command using uppercase letters, where it should be in lowercase, will produce different outputs.

36) What are environmental variables?
Environmental variables are global settings that control the shell's function as well as that of other Linux programs. Another common term for environmental variables is global shell variables.

37) What are the different modes when using vi editor?
There are 3 modes under vi:- Command mode – this is the mode where you start in- Edit mode – this is the mode that allows you to do text editing- Ex mode – this is the mode wherein you interact with vi with instructions to process a file

38) Is it possible to use shortcuts for a long pathname?
Yes, there is. A feature known as filename expansion allows you do this using the TAB key. For example, if you have a path named /home/iceman/assignments directory, you would type as follows: /ho[tab]/ice[tab]/assi[tab] . This, however, assumes that the path is unique and that the shell you're using supports this feature.

39) What is redirection?
Redirection is the process of directing data from one output to another. It can also be used to direct an output as an input to another process.

40) What is grep command?
grep a search command that makes use of pattern-based searching. It makes use of options and parameters that are specified along with the command line and applies this pattern in searching the required file output.

41) What could be the problem when a command that was issued gave a different result from the last time it was used?
One highly possible reason for getting different results from what seems to be the same command has something to do with case sensitivity issues. Since Linux is case sensitive, a command that was previously used might have been entered in a different format from the present one. For example, to lists all files in the directory, you should type the command ls, and not LS. Typing LS will either result in an error message if there is no program by that exact name exist or may produce a different output if there is a program named LS that performs another function.

42) What are the contents of /usr/local?
It contains locally installed files. This directory matters in environments where files are stored on the network. Specifically, locally-installed files go to /usr/local/bin, /usr/local/lib, etc.). Another application of this directory is that it is used for software packages installed from source, or software not officially shipped with the distribution.

43) How do you terminate an ongoing process?
Every process in the system is identified by a unique process id or pid. Use the kill command followed by the pid to terminate that process. To terminate all process at once, use kill 0.

44) How do you insert comments in the command line prompt?
Comments are created by typing the # symbol before the actual comment text. This tells the shell to completely ignore what follows. For example "# This is just a comment that the shell will ignore."

45) What is command grouping and how does it work?
You can use parentheses to group commands. For example, if you want to send the current date and time along with the contents of a file named OUTPUT to a second file named MYDATES, you can apply command grouping as follows:

```
(date cat OUTPUT) > MYDATES
```

46) How do you execute more than one command or program from a single command line entry?
You can combine several commands by separating each command or program using a semicolon symbol. For example, you can issue such a series of commands in a single entry:
ls –l cd .. ls –a MYWORK which is equivalent to 3 commands: ls -l cd.. ls -a MYWORK
**Note that this will be executed one after the other, in the order specified.

47) Write a command that will look for files with an extension "c", and has the occurrence of the string "apple" in it.
Answer: Find ./ -name "*.c" | xargs grep –i "apple"

48) Write a command that will display all .txt files, including its individual permission.
Answer: ls -al *.txt

49) Write a command that will do the following:

-look for all files in the current and subsequent directories with an extension c,v
-strip the,v from the result (you can use sed command)
-use the result and use a grep command to search for all occurrences of the word ORANGE in the files.

```
Find ./ -name "*.c,v" | sed 's/,v//g' | xargs grep "ORANGE"
```

50) What, if anything, is wrong with each of the following commands?

a) ls -l-s
b) cat file1, file2
c) ls - s Factdir
Answers:
a) there should be space between the 2 options: ls -l -s
b) do not use commas to separate arguments: cat file1 file2
c) there should be no space between hyphen and option label: ls –s Factdir

51) What is the command to calculate the size of a folder?

To calculate the size of a folder uses the command du –sh folder1.

52) How can you find the status of a process?

Use the command $ ***ps ux***

53) How can you check the memory status?

You can use the command
free -m to display output in MB
free -g to display output in GB

54) Explain how to color the Git console?

To color the Git console, you can use the command git config—global color.ui auto. In the command, the color.ui variable sets the default value for a variable such as color.diff and color.grep.

55) How can you append one file to another in Linux?

To append one file to another in Linux you can use command cat file2 >> file 1. The operator >> appends the output of the named file or creates the file if it is not created. While another command cat file 1 file 2 > file 3 appends two or more files to one.

56) Explain how you can find a file using Terminal?

To find a file you have to use a command, find . –name "process.txt" . It will look for the current directory for a file called process.txt.

57) Explain how you can create a folder using Terminal?

To create a folder, you have to use command mkdir. It will be something like these: ~$ mkdir Guru99

58) Explain how you can view the text file using Terminal?

To view the text file, go to the specific folder where the text files are located by using the command cd and then type less filename.txt.

59) Explain how to enable curl on Ubuntu LAMP stack?

To enable curl on Ubuntu, first, install libcurl, once done use following command sudo/etc/init .d /apache2 restart or sudo service apache2 restart.

60) Explain how to enable root logging in Ubuntu?

The command which enables root logging is

```
#sudo sh-c 'echo "greater-show-manual-login=true" >>/etc/lightdm/lightdm.conf'
```

61) How can you run a Linux program in the background simultaneously when you start your Linux Server?
By using nohup. It will stop the process receiving the NOHUP signal and thus terminating it you log out of the program which was invoked with. & runs the process in the background.

62) Explain how to uninstall the libraries in Linux?
To uninstall the libraries in Linux, you can use command sudo apt-get remove library_name