The following are the 6 high level stages of a typical Linux boot process.

1. BIOS

BIOS stands for Basic Input/Output System

Performs some system integrity checks

Searches, loads, and executes the boot loader program.

It looks for boot loader in floppy, cd-rom, or hard drive. You can press a key (typically F12 of F2, but it depends on your system) during the BIOS startup to change the boot sequence.

Once the boot loader program is detected and loaded into the memory, BIOS gives the control to it.

So, in simple terms BIOS loads and executes the MBR boot loader.

2. MBR

MBR stands for Master Boot Record.

It is located in the 1st sector of the bootable disk. Typically /dev/hda, or /dev/sda

MBR is less than 512 bytes in size. This has three components 1) primary boot loader info in 1st 446 bytes 2) partition table info in next 64 bytes 3) mbr validation check in last 2 bytes.

It contains information about GRUB (or LILO in old systems).

So, in simple terms MBR loads and executes the GRUB boot loader.

3. GRUB

GRUB stands for Grand Unified Bootloader.

If you have multiple kernel images installed on your system, you can choose which one to be executed.

GRUB displays a splash screen, waits for few seconds, if you don’t enter anything, it loads the default kernel image as specified in the grub configuration file.

GRUB has the knowledge of the filesystem (the older Linux loader LILO didn’t understand filesystem).

Grub configuration file is /boot/grub/grub.conf (/etc/grub.conf is a link to this). The following is sample grub.conf of CentOS.

#boot=/dev/sda

default=0

timeout=5

splashimage=(hd0,0)/boot/grub/splash.xpm.gz

hiddenmenu

title CentOS (2.6.18-194.el5PAE)

root (hd0,0)

kernel /boot/vmlinuz-2.6.18-194.el5PAE ro root=LABEL=/

initrd /boot/initrd-2.6.18-194.el5PAE.img

As you notice from the above info, it contains kernel and initrd image.

So, in simple terms GRUB just loads and executes Kernel and initrd images.

4. Kernel

Mounts the root file system as specified in the “root=” in grub.conf

Kernel executes the /sbin/init program

Since init was the 1st program to be executed by Linux Kernel, it has the process id (PID) of 1. Do a ‘ps -ef | grep init’ and check the pid.

initrd stands for Initial RAM Disk.

initrd is used by kernel as temporary root file system until kernel is booted and the real root file system is mounted. It also contains necessary drivers compiled inside, which helps it to access the hard drive partitions, and other hardware.

5. Init

Looks at the /etc/inittab file to decide the Linux run level.

Following are the available run levels

0 – halt

1 – Single user mode

2 – Multiuser, without NFS

3 – Full multiuser mode

4 – unused

5 – X11

6 – reboot

Init identifies the default initlevel from /etc/inittab and uses that to load all appropriate program.

Execute ‘grep initdefault /etc/inittab’ on your system to identify the default run level

If you want to get into trouble, you can set the default run level to 0 or 6. Since you know what 0 and 6 means, probably you might not do that.

Typically, you would set the default run level to either 3 or 5.

6. Runlevel programs

When the Linux system is booting up, you might see various services getting started. For example, it might say “starting sendmail …. OK”. Those are the runlevel programs, executed from the run level directory as defined by your run level.

Depending on your default init level setting, the system will execute the programs from one of the following directories.

Run level 0 – /etc/rc.d/rc0.d/

Run level 1 – /etc/rc.d/rc1.d/

Run level 2 – /etc/rc.d/rc2.d/

Run level 3 – /etc/rc.d/rc3.d/

Run level 4 – /etc/rc.d/rc4.d/

Run level 5 – /etc/rc.d/rc5.d/

Run level 6 – /etc/rc.d/rc6.d/

Please note that there are also symbolic links available for these directory under /etc directly. So, /etc/rc0.d is linked to /etc/rc.d/rc0.d.

Under the /etc/rc.d/rc\*.d/ directories, you would see programs that start with S and K.

Programs starts with S are used during startup. S for startup.

Programs starts with K are used during shutdown. K for kill.

There are numbers right next to S and K in the program names. Those are the sequence number in which the programs should be started or killed.

For example, S12syslog is to start the syslog deamon, which has the sequence number of 12. S80sendmail is to start the sendmail daemon, which has the sequence number of 80. So, syslog program will be started before sendmail.

Bash Bang (!) Commands

In the final part of this article, we will explain some useful ! (bang) operations:

!! – execute last command.

!top – execute the most recent command that starts with ‘top’ (e.g. !).

!top:p – displays the command that !top would run (also adds it as the latest command in the command history).

!$ – execute the last word of the previous command (same as Alt +., e.g. if last command is ‘cat tecmint.txt’, then !$ would try to run ‘tecmint.txt’).

!$:p – displays the word that !$ would execute.

!\* – displays the last word of the previous command.

!\*:p – displays the last word that !\* would substitute.

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

1

pwd

Output:

/home/guru99/myDir

1

/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:

MS DOS

ls –l cd .. ls –a MYWORK which is equivalent to 3 commands: ls -l cd.. ls -a MYWORK

1

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:

MS DOS

Find ./ -name “\*.c” | xargs grep –i “apple”

1

Find ./ -name “\*.c” | xargs grep –i “apple”

48) Write a command that will display all .txt files, including its individual permission.

Answer:

MS DOS

ls -al \*.txt

1

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.

MS DOS

Find ./ -name “\*.c,v” | sed ‘s/,v//g’ | xargs grep “ORANGE”

1

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

63) What is Swap space?

Ans: Swap space is the amount of physical memory that is allocated for use by Linux to hold some concurrent running programs temporarily. This condition usually occurs when Ram does not have enough memory to support all concurrent running programs. This memory management involves the swapping of memory to and from physical storage.

There are different commands and tools available to manage the Swap space usage.

64) What do you understand by Root account?

Ans: As the name suggests, it is like a system administrator account which gives you the ability to fully control the system. Root account serves as the default account whenever Linux is installed.Below mentioned functions can be performed by Root account

Below mentioned functions can be performed by Root account

Create user accounts

Maintain user accounts

Assign different permissions to each account created and so on..

Que 9) Explain virtual desktop?

Ans: When there are multiple windows available on the current desktop and there appears the problem of minimizing and maximizing windows or restoring all the current programs, there ‘Virtual Desktop’ serves as an alternative. It allows you to open one or more programs on a clean slate. Virtual desktops are basically stored on a remote server and serve the following benefits:

Virtual desktops are basically stored on a remote server and serve the following benefits

Cost savings as the resources can be shared and allocated as and when required.

Resources and energy are more efficiently used.

Data integrity is improved.

Centralized administration

Less compatibility issues.

Que 10) Differentiate between BASH and DOS?

Ans: The basic differences between BASH and DOS can be understood from the below table

Show entriesSearch:

BASH DOS

BASH commands are case sensitive. DOS commands are not case sensitive.

‘/’ character used as a directory separator.

‘\’ character acts as an escape character. ‘/’ character: serves as a command argument delimiter.

‘\’ character: serves as a directory separator.

File naming convention includes: 8 character file name followed by a dot and 3 characters for the extension. No file naming convention is followed in DOS.

Showing 1 to 3 of 3 entriesPreviousNext

Que 11) Explain the term GUI?

Ans: GUI stands for the Graphical user interface. GUI is considered as the most attractive and user-friendly because it consists of the usage of images and icons. These images and icons are clicked and being manipulated by the users for the purpose of communication with the system.

Advantages of GUI:

Allows users to navigate and operate the software with the help of visual elements.

More intuitive and rich interface is possible to be created.

Less chances of occurrence of errors as complex, multi-step, dependent tasks are easily grouped together.

Productivity is enhanced with the means of multitasking as with a simple click of the mouse, the user is able to maintain multiple open applications and transitions between them.

Disadvantages of GUI:

End-users have less control over the operating system and file systems.

Although it is easier to use mouse and keyboard for navigation and controlling operating system, the whole process is a bit slow.

It requires more resources because of the elements that need to be loaded such as icons, fonts, etc.

Que 12) Explain the term CLI?

Ans: CLI stands for Command Line Interface. It is a way for humans to interact with computers and is also known as Command line user interface. It relies on textual request and response transaction process where user types declarative commands to instruct the computer to perform operations.

Advantages of CLI

Very flexible

Can easily access commands

Much faster and easier to use by expert

Does not use much CPU processing time.

Disadvantages of CLI

Learning and remembering type commands is hard.

Have to be typed precisely.

Can be very confusing.

Surfing web, graphics, etc are few tasks which are hard or impossible to do on the command line.

Que 13) Enlist some Linux distributors (Distros) along with its usage?

Ans: Different parts of LINUX say kernel, system environment, graphical programs, etc are developed by different organizations. LINUX Distributions (Distros) assemble all these different parts of Linux and give us a compiled operating system to be installed and used.

There are around six hundred Linux distributors. Let us see some of the important ones

UBuntu: It is a well known Linux Distribution with a lot of pre-installed apps and easy to use repositories libraries. It is very easy to use and works like MAC operating system.

Linux Mint: It uses cinnamon and mate desktop. It works on windows and should be used by newcomers.

Debian: It is the most stable, quicker and user-friendly Linux Distributors.

Fedora: It is less stable but provides the latest version of the software. It has GNOME3 desktop environment by default.

Red Hat Enterprise: It is to be used commercially and to be well tested before release. It usually provides the stable platform for a long time.

Arch Linux: Every package is to be installed by you and is not suitable for the beginners.

Que 14) How can you determine the total memory used by LINUX?

Ans: It is always required to keep a check on the memory usage in order to find out whether the user is able to access the server or the resources are adequate. There are roughly 5 methods that determine the total memory used by the Linux.

This is explained as below

Free command: This is the most simple and easy to use the command to check memory usage. For example: ‘$ free –m’, the option ‘m’ displays all the data in MBs.

/proc/meminfo: The next way to determine the memory usage is to read /proc/meminfo file. For example: ‘$ cat /proc/meminfo’

Vmstat: This command basically lays out the memory usage statistics. For example: ‘$ vmstat –s’

Top command: This command determines the total memory usage as well as also monitors the RAM usage.

Htop: This command also displays the memory usage along with other details.

Que 15) Explain the 3 kinds of file permissions under LINUX?

Ans: Every file and directory in Linux are assigned three types of owners namely ‘User’, ‘Group’, and ‘Others’. Find the three kinds of permissions defined for all the three owners

Read: This permission allows you to open and read the file as well as list the contents of the directory.

Write: This permission allows you to modify the contents of the file as well as allows adding, removing and renaming files stored in the directories.

Execute: User can access and run the file in the directory. You cannot run a file unless the execute permission is set.

Que 16) What is the maximum length for any file name under LINUX?

Ans: The maximum length for any filename under Linux is 255 characters.

Que 17) How permissions are granted under LINUX?

Ans: A system administrator or the owner of the file can grant permissions using the ‘chmod’ command. Following symbols are used while writing permissions

Is it legal to edit Linux Kernel?

Yes. You can edit Linux Kernel because it is released under General Public License (GPL) and any one can edit it. It comes under the category of free and open source software.

What is the basic difference between BASH and DOS?

BASH commands are case sensitive while DOS commands are not case sensitive.

DOS follows a convention in naming files. In DOS, 8 character file name is followed by a dot and 3 characters for the extension. BASH doesn't follow such convention.

Suppose your company is recently switched from Microsoft to Linux and you have some MS Word document to save and work in Linux, what will you do?

Install Open Office Suite on Linux. It facilitates you to work with Microsoft documents.

What is Samba? Why is it used?

Samba service is used to connect Linux machines to Microsoft network resources by providing Microsoft SMB supp

What is the maximum length for a file name in Linux?

255 characters.

You wish to print a file ‘draft’ with 60 lines on a page. What command would you use?

The command used : pr -l60 draft

The default page length when using pr is 66 lines.

The -l option specifies a different length. What is LD\_LIBRARY\_PATH?

LD\_LIBRARY\_PATH is an environment variable. It is used for debugging a new library or a non-standard library. It is also used to identify what directories need to be searched. Path to search for directories needs to given.

The variable can be set using setenv—LD\_LIBRARY\_PATH–$PATH.

It is used to search for the shared objects / dynamic libraries by the operating system for extendable functionality at runtime.

Prepare yourself for the Linux Certification with comprehensive Linux online training.

5. Name a service that you should disable (which acts both as Web and FTP Server) on a Linux Server.

The finger service should be disabled because a remote user can get important information about the system by using that command.

Learn all about Linux through this online training course.

6. What does Sar provide? Where are Sar logs stored?

Sar collects, reports, or saves system activity information, sar serves to log and evaluate a variety of information regarding system activity. With performance problems, sar also permits retroactive analysis of the load values for various sub-systems (CPUs, memory, disks, interrupts, network interfaces and so forth) and limitation of problems in this manner. If CPU utilization is near 100 % (user + nice + system), the workload sampled is CPU-bound.

By default, log files of Sar command is located at /var/log/sa/sadd file, where dd parameter indicates the current day.

7. How to check Memory stats and CPU stats as a Linux admin?

Using ‘free’ & ‘vmstat’ command, we can display the physical and virtual memory statistics, respectively. With the help of ‘sar’ command, we can see the CPU utilization & other stats.

8. How to reduce or shrink the size of LVM partition?

Below are the logical steps to reduce size of LVM partition :

Unmount the filesystem using unmount command,

Use resize2fs command , e.g resiz2fs /dev/mapper/myvg-mylv 10G

Now use the lvreduce command as lvreduce -L 10G /dev/mapper/myvg-mylv

The above Command will shrink the size LVM partition and fixes the filesystem size to 10GB.Most in-depth, industry-led curriculum in linux.

9. What are the different modes of Network bonding in Linux?

Below are list of modes used in Network Bonding :

balance-rr or 0 – round-robin mode for fault tolerance and load balancing.

active-backup or 1 – Sets active-backup mode for fault tolerance.

balance-xor or 2 – Sets an XOR (exclusive-or) mode for fault tolerance and load balancing.

broadcast or 3 – Sets a broadcast mode for fault tolerance. All transmissions are sent on all slave interfaces.

802.3ad or 4 – Sets an IEEE 802.3ad dynamic link aggregation mode. Creates aggregation groups that share the same speed & duplex settings.

balance-tlb or 5 – Sets a Transmit Load Balancing (TLB) mode for fault tolerance & load balancing.

balance-alb or 6 – Sets an Active Load Balancing (ALB) mode for fault tolerance & load balancing.

Now that you are aware of network bonding in Linux, Check Linux online training.

10. How to check and verify the status of the bond interface?

Using the command ‘cat /proc/net/bonding/bond0,’ we can check what mode is enabled and what LAN cards are used in this bond. In this example, we have one one bond interface. However, we can have multiple bond interfaces like bond1, bond2 and so on.

11. How can you enhance the security of password file?

Linux keep user account information in a text file called /etc/passwd. This file also stores one way encrypted password. It is accessed by several tools to get user information, which is a security risk, so file need to ‘Word Readable.’ To minimize the security risk, you can use shadow password format. This method saves account information in regular file /etc/passwd.

However, the password is stored as a single “x” character (not actually stored in this file). A second file, called “/etc/shadow” contains encrypted password as well as other information such as account or password expiration values, etc. The /etc/shadow file is readable only by the root account and is therefore less of a security risk

12. What command can you use to make a tape archive file of a /home directory and send it to the /dev/tape device?

The correct command is tar -cvf /dev/tape /home.

The -xvf option is used to extract files from an archive.Expert Linux Professionals are in much demand.

Take this linux administration online certification & join the big league.

13. Suppose your FTP Server is hacked and the entire server needs to be restored. How should you restore the original kernel system files?

You cannot restore the entire operating system from tape backup device. So you should reinstall the core operating system, and then restore system configuration files and user data from tape backup device.

14. Why should you avoid Telnet to administer a Linux system remotely?

Telnet uses most insecure method for communication. It sends data across the network in plain text format and anybody can easily find out the password using the network tool. In the case of Telnet, these include the passing of login credentials in plain text, which means anyone running a sniffer on your network can find the information he needs to take control of a device in a few seconds by eavesdropping on a Telnet login session.

15. What is Puppet Server?

Puppet is an open-source & enterprise software for configuration management toll in UNIX like operating system. Puppet is an IT automation software that is used to push configuration to its clients (puppet agents) using code. Puppet code can do multiple tasks from installing new software to checking file permissions to updating user accounts.

16. What is the difference between Cron and Anacron?

Minimum granularity with Cron is minute while it is in days with Anacron.

Cron job can be scheduled by any normal user while Anacron can be scheduled only by the super user (the superuser is a special user account used for system administration. Depending on the operating system (OS), the actual name of this account might be root,administrator, admin or supervisor)

Cron expects the system to be up and running while the Anacron doesn’t expect the system to be up and running all the time. In case of Anacron, if a job is scheduled and the system is down that time, it will execute the job as soon as the system is up and running.

Cron is ideal for servers while Anacron is ideal for desktops and laptops.

Cron should be used when you want a job to be executed at a particular hour and minute while Anacron should be used in when the job can be executed irrespective of the hour and minute.

17. What command is used to check the number of files, disk space and each user’s defined quota?

repquota command is used to check the status of the user’s quota along with the disk space and number of files used.

This command gives a summary of the user’s quota that how much space and files are left for the user. Each user has a defined quota in Linux. This is done mainly for the security as some users have only limited access to files. This provides a security to the files from unwanted access. The quota can be given to a single user or to a group of users.

18. What is the name and path of the main system log?

By default, the main system log is ‘/var/log/messages’. This file contains all the messages and the script written by the user. By default, all scripts are saved in this file. This is the standard system log file, which contains messages from all system software, non-kernel boot issues, and messages that go to ‘dmesg’. dmesg is a system file that is written upon system boot.

19. Can Linux computer be made a router so that several machines may share a single Internet connection? How?

Yes, a Linux machine can be made a router. This is called “IP Masquerade.” IP Masquerade is a networking function in Linux similar to the one-to-many (1: Many) NAT (Network Address Translation) servers found in many commercial firewalls and network routers.

The IP Masquerade feature allows other “internal” computers connected to this Linux box (via PPP, Ethernet, etc.) to also reach the Internet as well. Linux IP Masquerading allows this functionality even if the internal computers do not have IP addresses.

The IP masquerading can be done by the following steps :

The Linux PC must have an internet connection and a connection to LAN. Typically, the Linux PC has two network interfaces-an Ethernet card for the LAN and a dial-up PPP connection to the Internet (through an ISP).

All other systems on your LAN use the Linux PC as the default gateway for TCP/IP networking. Use the same ISP-provided DNS addresses on all systems.

Enable IP forwarding in the kernel. By default the IP forwarding is not enabled. To ensure that IP forwarding is enabled when you reboot your system, place this command in the /etc/rc.d/rc.local file.

Run /sbin/iptables-the IP packet filter administration program-to set up the rules that enable the Linux PC to masquerade for your LAN.

20. How to change window manager by editing your home directory?

“/.xinitrc file” allows changing the window manager we want to use when logging into X from that account. The dot in the file name shows you that the file is a hidden file and doesn’t show when you do a normal directory listing. For setting a window manager we have to save a command in this file.

The syntax of command is : exec window manager. After this, save the file. Next time when you run a startx a new window manager will open and become default.

The commands for starting some popular window managers and desktop environments are :

KDE = startkde

Gnome = gnome-session.

Blackbox = blackbox.

FVWM = fvwm

Window Maker = wmaker.

IceWM = icewm

21. How shadow passwords are given by in Linux?

pwconv command is used for giving shadow passwords. Shadow passwords are given for better system security. The pwconv command creates the file /etc/shadow and changes all passwords to ‘x’ in the /etc/passwd file.

First, entries in the shadowed file which don’t exist in the main file are removed. Then, shadowed entries which don’t have `x’ as the password in the main file are updated. Any missing shadowed entries are added. Finally, passwords in the main file are replaced with `x’. These programs can be used for initial conversion as well to update the shadowed file if the main file is edited by hand.

22. What daemon is used for scheduling of the commands?

The crontab command is used for scheduling of the commands to run at a later time.

SYNTAX

crontab [ -u user ] file

crontab [ -u user ] { -l | -r | -e }

Options

l List – display the current crontab entries.

r Remove the current crontab.

e Edit the current crontab using the editor specified by the VISUAL or EDITOR environment variables.

When user exits from the editor, the modified crontab will be installed automatically. Each user can have their own crontab, and though these are files in /var, they are not intended to be edited directly.

If the –u option is given than the crontab gives the name of the user whose crontab is to be tweaked. If it is given without this then it will display the crontab of the user who is executing the command.

23. What shell does a Linux Administrator assign to a POP3 mail-only account?

POP3 mail only account is assigned to the /bin/false shell. However, assigning bash shell to a POP3 mail only gives user login access, which is avoided. /bin/nologin can also be used. This shell is provided to the user when we don’t want to give shell access to the user. The user cannot access the shell and it rejects shell login on the server as in Telnet. It is mainly for the security of the shells.

POP3 is basically used for downloading mail to mail program. So for illegal downloading of emails on the shell, this account is assigned to the /bin/false shell or /bin/nologin. These both shells are same as they both do the same work of rejecting the user login to the shell.

The main difference between these two shells is that false shell shows the incorrect code and any unusual coding when user login to it. But the nologin shell simply tells that no such account is available. So nologin shell is used often in Linux.

24. If a volume group named VG0 already exists and we need to extend this volume group up to 4GB. How will you do it?

Firstly create Physical volume (/dev/sda7) of size 4GB.

Now run following command.

vgextend VG0 /dev/sda7

25. Is there any relation between modprobe.conf file and network devices?

Yes, this file assigns a kernel module to each network device.

For Example :-

[root@localhost ~]# cat /etc/modprobe.conf

alias eth0 b44

Here, b44 is the kernel module for network device eth0.

We can confirm whether this module “b44” is present or not by the following command

[root@localhost ~]# lsmod |grep b44

b44 29005 0

Learn Linux from Experts! Enrol Today

26. What is YUM?

YUM stands for Yellow dog Updater, Modified because it is based on YUP, the Yellow dog Updater. Where does the name Yellow dog come from? Yellow Dog is a version of Linux for the Power Architecture hardware and is RPM-based, just like Red Hat Enterprise Linux and Fedora. YUP, and later YUM, were written by the Linux community as a way to maintain an RPM-based system

27. What is the role of Kudzu?

Kudzu is used to detect new Hardware. RedHat Linux runs a hardware discoverer, named kudzu. When attempting to identify a serial port Kudzu resets the serial port. This stops the serial console. Kudzu is configured from the file/etc/sysconfig/kudzu.

Kudzu can be prevented from resetting hardware, by setting the configuration parameter SAFE to yes.

28. What is the difference between ext2 and ext3 file systems?

The ext3 file system is an enhanced version of the ext2 file system.

The most important difference between Ext2 and Ext3 is that Ext3 supports journaling.

After an unexpected power failure or system crash (also called an unclean system shutdown), each mounted ext2 file system on the machine must be checked for consistency by the e2fsck program. This is a time-consuming process and during this time, any data on the volumes is unreachable.

The journaling provided by the ext3 file system means that this sort of file system check is no longer necessary after an unclean system shutdown. The only time a consistency check occurs using ext3 is in certain rare hardware failure cases, such as hard drive failures. The time to recover an ext3 file system after an unclean system shutdown does not depend on the size of the file system or the number of files. Rather, it depends on the size of the journal used to maintain consistency. The default journal size takes almost a second to recover, depending on the speed of the hardware.

29. Explain /proc filesystem?

/proc is a virtual file system that provides detailed information about Linux kernel, hardware and running processes. Files under /proc directory named as Virtual files.

Since /proc contains virtual files, it is called virtual file system. These virtual files have unique qualities. Most of them are listed as zero bytes in size.

Virtual files such as /proc/interrupts, /proc/meminfo, /proc/mounts, and /proc/partitions provide an up-to-the-moment glimpse of the system’s hardware. Others: /proc/filesystems file and the /proc/sys/ directory provide system configuration information and interfaces.

30. How do you create ext4 file system?

# mke2fs -t ext4 /dev/DEV

31. How to Enable ACLs for /home partition?

Add following entry in /etc/fstab

LABEL=/home /home ext3 acl 1 2

Now remount /home partition with acl option.

mount -t ext3 -o acl /dev/sda3 /home

**Linux/Unix user management commands**

Here is a list of linux user management commands

|  |  |
| --- | --- |
| **Command** | **Description** |
| sudo adduser username | Adds a user |
| sudo passwd -l 'username' | Disable a user |
| sudo userdel -r 'username' | Delete a user |
| sudo usermod -a -G GROUPNAME USERNAME | Add user a to a usergroup |
| sudo deluser USER GROUPNAME | Remove user from a user group |
| finger | Gives information on all logged in user |
| finger username | Gives information of a particular user |

### **Q1. What is Jenkins?**

My suggestion is to start this answer by giving a definition of Jenkins.

Jenkins is an open source automation tool written in Java with plugins built for Continuous Integration purpose. Jenkins is used to build and test your software projects continuously making it easier for developers to integrate changes to the project, and making it easier for users to obtain a fresh build. It also allows you to continuously deliver your software by integrating with a large number of testing and deployment technologies.

Once you have defined Jenkins give an example, you can refer the below mentioned use case:

* First, a developer commits the code to the source code repository. Meanwhile, the Jenkins server checks the repository at regular intervals for changes.
* Soon after a commit occurs, the Jenkins server detects the changes that have occurred in the source code repository. Jenkins will pull those changes and will start preparing a new build.
* If the build fails, then the concerned team will be notified.
* If built is successful, then Jenkins deploys the built in the test server.
* After testing, Jenkins generates a feedback and then notifies the developers about the build and test results.
* It will continue to check the  source code repository for changes made in the source code and the whole process keeps on repeating.



Interviewer now knows what is Jenkins but why we use it, there are many other CI tools as well, so why Jenkins?, the next question in this Jenkins interview questions will deal with that answer.

### **Q2. What are the benefits of using Jenkins?**

I will suggest you to include the following benefits of Jenkins, if you can recall any other benefit apart from the below mentioned points you can include that as well.

* At integration stage, build failures are cached.
* For each change in the source code an automatic build report notification is generated.
* To notify developers about build report success or failure, it is integrated with LDAP mail server.
* Achieves continuous integration agile development and test driven development.
* With simple steps, maven release project is automated.
* Easy tracking of bugs at early stage in development environment than production.

Interviewer: Okay Jenkins looks like a really cool tool, but what are the requirements for using Jenkins?

### **Q3. What are the pre-requisites for using Jenkins?**

Answer to this is pretty straightforward To use Jenkins you require:

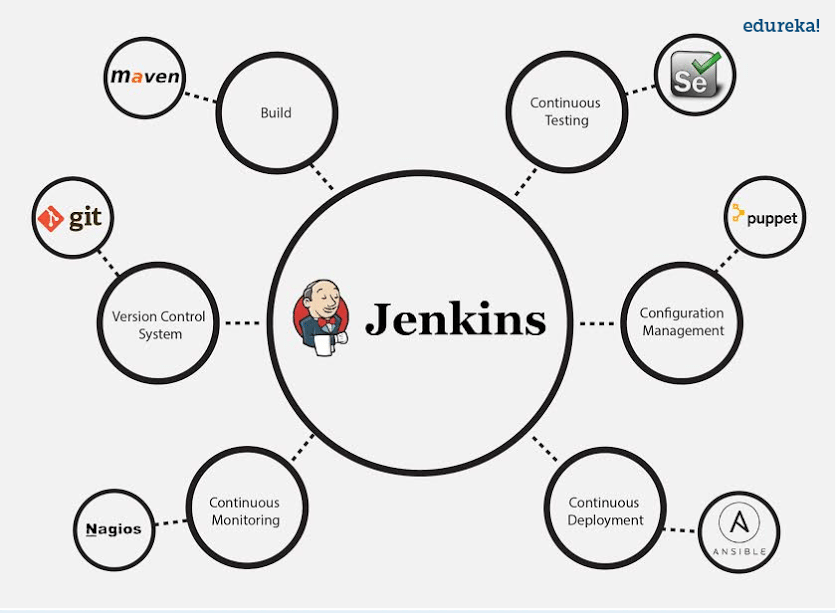
* A source code repository which is accessible, for instance, a Git repository.
* A working build script, e.g., a Maven script, checked into the repository.

Remember, you have mentioned Plugins in your previous answer, so next question in this Jenkins interview questions blog will be regarding Plugins.

### **Q4. Mention some of the useful plugins in Jenkins?**

Below I have mentioned some important Plugins:

* Maven 2 project
* Git
* Amazon EC2
* HTML publisher
* Copy artifact
* Join
* Green Balls



These Plugins I feel are the most useful plugins, if you want to include any other Plugin that is not mentioned above, you can add that as well, but make sure you first mention the above stated plugins and then add your own.

### **Q5. Mention what are the commands you can use to start Jenkins manually?**

For this answer I will suggest you to go with the below mentioned flow:

To start Jenkins manually open Console/Command line, then go to your Jenkins installation directory. Over there you can use the below commands:

To start Jenkins: **jenkins.exe start**  
To stop Jenkins: **jenkins.exe stop**  
To restart Jenkins: **jenkins.exe restart**

### **Q6. Explain how you can set up Jenkins job?**

My approach to this answer will be to first mention how to create Jenkins job.

Go to Jenkins top page, select “New Job”, then choose “Build a free-style software project”.

Now you can tell the elements of this freestyle job:

* Optional SCM, such as CVS or Subversion where your source code resides.
* Optional triggers to control when Jenkins will perform builds.
* Some sort of build script that performs the build (ant, maven, shell script, batch file, etc.) where the real work happens.
* Optional steps to collect information out of the build, such as archiving the artifacts and/or recording javadoc and test results.
* Optional steps to notify other people/systems with the build result, such as sending e-mails, IMs, updating issue tracker, etc..

### **Q7. Explain how to create a backup and copy files in Jenkins?**

Answer to this question is really direct.

To create a backup all you need to do is to periodically back up your JENKINS\_HOME directory. This contains all of your build jobs configurations, your slave node configurations, and your build history. To create a back-up of your Jenkins setup, just copy this directory. You can also copy a job directory to clone or replicate a job or rename the directory.

[**Learn Jenkins With DevOps Now**](https://www.edureka.co/devops)

### **Q8. How will you secure Jenkins?**

The way I secure Jenkins is mentioned below, if you have any other way to do it than mention that:

* Ensure global security is on.
* Ensure that Jenkins is integrated with my company’s user directory with appropriate plugin.
* Ensure that matrix/Project matrix is enabled to fine tune access.
* Automate the process of setting rights/privileges in Jenkins with custom version controlled script.
* Limit physical access to Jenkins data/folders.
* Periodically run security audits on same.

I hope you have enjoyed the above set of Jenkins interview questions, the next set of questions will be more challenging, so be prepared.

### **Q9 Explain how you can deploy a custom build of a core plugin?**

Below are the steps to deploy a custom build of a core plugin:

* Stop Jenkins.
* Copy the custom HPI to **$Jenkins\_Home/plugins**.
* Delete the previously expanded plugin directory.
* Make an empty file called **<plugin>.hpi.pinned**.
* Start Jenkins.

### **Q10. What is the relation between Hudson and Jenkins?**

You can just say Hudson was the earlier name and version of current Jenkins. After some issue, the project name was changed from Hudson to Jenkins.

### **Q11. What you do when you see a broken build for your project in Jenkins?**

There can be multiple answers to this question I will approach this task in the following way:

I will open the console output for the broken build and try to see if any file changes were missed. If I am unable to find the issue that way, then I will clean and update my local workspace to replicate the problem on my local and try to solve it.

If you do it in a different way then just mention that in your answer.

### **Q12. Explain how you can move or copy Jenkins from one server to another?**

I will approach this task by copying the jobs directory from the old server to the new one. There are multiple ways to do that, I have mentioned it below:

You can:

* Move a job from one installation of Jenkins to another by simply copying the corresponding job directory.
* Make a copy of an existing job by making a clone of a job directory by a different name.
* Rename an existing job by renaming a directory. Note that if you change a job name you will need to change any other job that tries to call the renamed job.

### **Q13. What are the various ways in which build can be scheduled in Jenkins?**

You can schedule a build in Jenkins in the following ways:

* By source code management commits
* After completion of other builds
* Can be scheduled to run at specified time ( crons )
* Manual Build Requests

### **Q14. What is the difference between Maven, Ant and Jenkins?**

Maven and Ant are Build Technologies whereas Jenkins is a continuous integration tool.

### **Q15. Which SCM tools Jenkins supports?**

Below are Source code management tools supported by Jenkins:

* AccuRev
* CVS,
* Subversion,
* Git,
* Mercurial,
* Perforce,
* Clearcase
* RTC

Now, the next set of Jenkins interview questions will test your experience with Jenkins.

### **Q16. What are the two components Jenkins is mainly integrated with?**

According to me Jenkins is mainly integrated with the following:

* Version Control system like GIT,SVN.
* Build tools like Apache Maven.

If you have anything else in your mind then mention that as well but make sure you include the above two components in your answer.

Maven pom.xml file

POM is an acronym for Project Object Model. The pom.xml file contains information of project and configuration information for the maven to build the project such as dependencies, build directory, source directory, test source directory, plugin, goals etc.

Maven reads the pom.xml file, then executes the goal.

Before maven 2, it was named as project.xml file. But, since maven 2 (also in maven 3), it is renamed as pom.xml.

Elements of maven pom.xml file

For creating the simple pom.xml file, you need to have following elements:

Element Description

project It is the root element of pom.xml file.

modelVersion It is the sub element of project. It specifies the modelVersion. It should be set to 4.0.0.

groupId It is the sub element of project. It specifies the id for the project group.

artifactId It is the sub element of project. It specifies the id for the artifact (project). An artifact is something that is either produced or used by a project. Examples of artifacts produced by Maven for a project include: JARs, source and binary distributions, and WARs.

version It is the sub element of project. It specifies the version of the artifact under given group.

File: pom.xml

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0

http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.javatpoint.application1</groupId>

<artifactId>my-app</artifactId>

<version>1</version>

</project>

Maven pom.xml file with additional elements

Here, we are going to add other elements in pom.xml file such as:

Element Description

packaging defines packaging type such as jar, war etc.

name defines name of the maven project.

url defines url of the project.

dependencies defines dependencies for this project.

dependency defines a dependency. It is used inside dependencies.

scope defines scope for this maven project. It can be compile, provided, runtime, test and system.

File: pom.xml

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0

http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.javatpoint.application1</groupId>

<artifactId>my-application1</artifactId>

<version>1.0</version>

<packaging>jar</packaging>

<name>Maven Quick Start Archetype</name>

<url>http://maven.apache.org</url>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.8.2</version>

<scope>test</scope>

</dependency>

</dependencies>

</project>

Q1. What is the difference between Git and SVN?

Git vs SVN

Git SVN

1. Git is a Decentralized Version Control tool 1. SVN is a Centralized Version Control tool

2. It belongs to the 3rd generation of Version Control tools 2. It belongs to the 2nd generation of Version Control tools

3. Clients can clone entire repositories on their local systems 3. Version history is stored on a server-side repository

4. Commits are possible even if offline 4. Only online commits are allowed

5. Push/pull operations are faster 5. Push/pull operations are slower

6. Works are shared automatically by commit 6. Nothing is shared automatically

Q2. What is Git?

I will suggest you to attempt this question by first telling about the architecture of git as shown in the below diagram just try to explain the diagram by saying:

Git is a Distributed Version Control system (DVCS). It can track changes to a file and allows you to revert back to any particular change.

Its distributed architecture provides many advantages over other Version Control Systems (VCS) like SVN one major advantage is that it does not rely on a central server to store all the versions of a project’s files.

Instead, every developer “clones” a copy of a repository I have shown in the diagram with “Local repository” and has the full history of the project on his hard drive so when there is a server outage all you need for recovery is one of your teammate’s local Git repository.

There is a central cloud repository as well where developers can commit changes and share it with other teammates as you can see in the diagram where all collaborators are commiting changes “Remote repository”.

Git Architecture - Git Interview Questions - Edureka

Now, the next set of Git interview questions will test your experience with Git:

Q3. What is the command to write a commit message in Git?

Answer to this is pretty straightforward.

Command that is used to write a commit message is “git commit -a”.

Now explain about -a flag by saying -a on the command line instructs git to commit the new content of all tracked files that have been modified. Also mention you can use “git add<file>” before git commit -a if new files need to be committed for the first time.

Q4. What is ‘bare repository’ in Git?

You are expected to tell the difference between a “working directory” and “bare repository”.

A “bare” repository in Git just contains the version control information and no working files (no tree) and it doesn’t contain the special .git sub-directory. Instead, it contains all the contents of the .git sub-directory directly in the main directory itself, where as working directory consist of:

A .git subdirectory with all the Git related revision history of your repo.

A working tree, or checked out copies of your project files.

Q5. What language is used in Git?

Instead of just telling the name of the language, you need to tell the reason for using it as well. I will suggest you to answer this by saying:

Git uses ‘C’ language. GIT is fast, and ‘C’ language makes this possible by reducing the overhead of run times associated with high level languages.

Learn Git With DevOps Now

Q6. In Git how do you revert a commit that has already been pushed and made public?

There can be two answers to this question and make sure that you include both because any of the below options can be used depending on the situation:

Remove or fix the bad file in a new commit and push it to the remote repository. This is the most natural way to fix an error. Once you have made necessary changes to the file, commit it to the remote repository for that I will use

git commit -m “commit message”

Create a new commit that undoes all changes that were made in the bad commit.to do this I will use a command

git revert <name of bad commit>

Q7. What is the difference between git pull and git fetch?

Git pull command pulls new changes or commits from a particular branch from your central repository and updates your target branch in your local repository.

Git fetch is also used for the same purpose but it works in a slightly different way. When you perform a git fetch, it pulls all new commits from the desired branch and stores it in a new branch in your local repository. If you want to reflect these changes in your target branch, git fetch must be followed with a git merge. Your target branch will only be updated after merging the target branch and fetched branch. Just to make it easy for you, remember the equation below:

Git pull = git fetch + git merge

Q8. What is ‘staging area’ or ‘index’ in Git?

For this answer try to explain the below diagram as you can see:

That before completing the commits, it can be formatted and reviewed in an intermediate area known as ‘Staging Area’ or ‘Index’. From the diagram it is evident that every change is first verified in the staging area I have termed it as “stage file” and then that change is committed to the repository.

Staging Area - Git Interview Questions - Edureka

If your interviewer has good knowledge on Git he/she will dig in deep, so the next set of Git interview questions will be more challenging.

Q9. What is Git stash?

According to me you should first explain the need for Git stash.

Often, when you’ve been working on part of your project, things are in a messy state and you want to switch branches for sometime to work on something else. The problem is, you don’t want to do a commit of half-done work just so you can get back to this point later. The answer to this issue is Git stash.

Now explain what is Git stash.

Stashing takes your working directory that is, your modified tracked files and staged changes and saves it on a stack of unfinished changes that you can reapply at any time.

Q10. What is Git stash drop?

Begin this answer by saying for what purpose we use Git ‘stash drop’.

Git ‘stash drop’ command is used to remove the stashed item. It will remove the last added stash item by default, and it can also remove a specific item if you include it as an argument.

Now give an example.

If you want to remove a particular stash item from the list of stashed items you can use the below commands:

git stash list: It will display the list of stashed items like:

stash@{0}: WIP on master: 049d078 added the index file

stash@{1}: WIP on master: c264051 Revert “added file\_size”

stash@{2}: WIP on master: 21d80a5 added number to log

If you want to remove an item named stash@{0} use command git stash drop stash@{0}.

Q11. How do you find a list of files that has changed in a particular commit?

For this answer instead of just telling the command, explain what exactly this command will do.

To get a list files that has changed in a particular commit use the below command:

git diff-tree -r {hash}

Given the commit hash, this will list all the files that were changed or added in that commit. The -r flag makes the command list individual files, rather than collapsing them into root directory names only.

You can also include the below mentioned point, although it is totally optional but will help in impressing the interviewer.

The output will also include some extra information, which can be easily suppressed by including two flags:

git diff-tree –no-commit-id –name-only -r {hash}

Here –no-commit-id will suppress the commit hashes from appearing in the output, and –name-only will only print the file names, instead of their paths.

Q12. What is the function of ‘git config’?

First tell why we need ‘git config‘.

Git uses your username to associate commits with an identity. The git config command can be used to change your Git configuration, including your username.

Now explain with an example.

Suppose you want to give a username and email id to associate commit with an identity so that you can know who has made a particular commit. For that I will use:

git config –global user.name “Your Name”: This command will add username.

git config –global user.email “Your E-mail Address”: This command will add email id.

Q13. What does commit object contains?

Commit object contains the following components, you should mention all the three points present below:

A set of files, representing the state of a project at a given point of time

Reference to parent commit objects

An SHAI name, a 40 character string that uniquely identifies the commit object.

Q14. How can you create a repository in Git?

This is probably the most frequently asked questions and answer to this is really simple.

To create a repository, create a directory for the project if it does not exist, then run command “git init”. By running this command .git directory will be created in the project directory.

Q15. How do you squash last N commits into a single commit?

There are two options to squash last N commits into a single commit include both of the below mentioned options in your answer:

If you want to write the new commit message from scratch use the following command

git reset –soft HEAD~N &&

git commit

If you want to start editing the new commit message with a concatenation of the existing commit messages then you need to extract those messages and pass them to Git commit for that I will use

git reset –soft HEAD~N &&

git commit –edit -m”$(git log –format=%B –reverse .HEAD@{N})”

Q16. What is Git bisect? How can you use it to determine the source of a (regression) bug?

I will suggest you to first give a small definition of Git bisect.

Git bisect is used to find the commit that introduced a bug by using binary search. Command for Git bisect is

git bisect <subcommand> <options>

Now since you have mentioned the command above explain them what this command will do.

This command uses a binary search algorithm to find which commit in your project’s history introduced a bug. You use it by first telling it a “bad” commit that is known to contain the bug, and a “good” commit that is known to be before the bug was introduced. Then Git bisect picks a commit between those two endpoints and asks you whether the selected commit is “good” or “bad”. It continues narrowing down the range until it finds the exact commit that introduced the change.

Q17. How do you configure a Git repository to run code sanity checking tools right before making commits, and preventing them if the test fails?

I will suggest you to first give a small introduction to sanity checking.

A sanity or smoke test determines whether it is possible and reasonable to continue testing.

Now explain how to achieve this.

This can be done with a simple script related to the pre-commit hook of the repository. The pre-commit hook is triggered right before a commit is made, even before you are required to enter a commit message. In this script one can run other tools, such as linters and perform sanity checks on the changes being committed into the repository.

Finally, give an example, you can refer the below script:

#!/bin/sh

files=$(git diff –cached –name-only –diff-filter=ACM | grep ‘.go$’)

if [ -z files ]; then

exit 0

fi

unfmtd=$(gofmt -l $files)

if [ -z unfmtd ]; then

exit 0

fi

echo “Some .go files are not fmt’d”

exit 1

This script checks to see if any .go file that is about to be committed needs to be passed through the standard Go source code formatting tool gofmt. By exiting with a non-zero status, the script effectively prevents the commit from being applied to the repository.

The Interviewer has not started asking questions on branching yet, so the next set of Git interview questions will be dealing with branching in Git.

Q18. Describe branching strategies you have used?

This question is asked to test your branching experience with Git so, tell them about how you have used branching in your previous job and what purpose does it serves, you can refer the below mention points:

Feature branching

A feature branch model keeps all of the changes for a particular feature inside of a branch. When the feature is fully tested and validated by automated tests, the branch is then merged into master.

Task branching

In this model each task is implemented on its own branch with the task key included in the branch name. It is easy to see which code implements which task, just look for the task key in the branch name.

Release branching

Once the develop branch has acquired enough features for a release, you can clone that branch to form a Release branch. Creating this branch starts the next release cycle, so no new features can be added after this point, only bug fixes, documentation generation, and other release-oriented tasks should go in this branch. Once it is ready to ship, the release gets merged into master and tagged with a version number. In addition, it should be merged back into develop branch, which may have progressed since the release was initiated.

In the end tell them that branching strategies varies from one organization to another so I know basic branching operations like delete, merge, checking out a branch etc..

Q19. How will you know in Git if a branch has already been merged into master?

The answer is pretty direct.

To know if a branch has been merged into master or not you can use the below commands:

git branch –merged It lists the branches that have been merged into the current branch.

git branch –no-merged It lists the branches that have not been merged.

Q20. What is Git rebase and how can it be used to resolve conflicts in a feature branch before merge?

According to me you should start by saying git rebase is a command which will merge another branch into the branch where you are currently working, and move all of the local commits that are ahead of the rebased branch to the top of the history on that branch.

Now, once you have defined Git rebase time for an example to show how it can be used to resolve conflicts in a feature branch before merge.

If a feature branch was created from the master, and since then the master branch has received new commits, Git rebase can be used to move the feature branch to the tip of master. The command effectively will replay the changes made in the feature branch at the tip of master, allowing conflicts to be resolved in the process. When done with care, this will allow the feature branch to be merged into master with relative ease and sometimes as a simple fast-forward operation.

You can also expect some off track questions, so the next question in this Git interview questions blog will be regarding SubGit.

Q21. What is SubGit?

Begin this answer by explaining what is SubGit used for.

SubGit is a tool for SVN to Git migration. It creates a writable Git mirror of a local or remote Subversion repository and uses both Subversion and Git as long as you like.

Now you can include some advantages like you can do a fast one-time import from Subversion to Git or use SubGit within Atlassian Bitbucket Server.We can use SubGit to create a bi-directional Git-SVN mirror of existing Subversion repository. You can push to Git or commit to Subversion at your convenience. Synchronization will be done by SubGit.

Amending the Last Commit

To change the last commit, you can simply commit again, using the --amend flag:

$ git commit --amend -m "New and correct message"

Simply put, this overwrites your last commit with a new one. This also means that you're not limited to just editing the commit's message: you could also add another couple of changes you forgot.

Changing Your Committer Name & Email Globally

You can run the "git config" command with the --global flag; this will make sure all of your future commits use the given information:

$ git config --global user.name "John Doe"

$ git config --global user.email "john@doe.org"

Changing Your Committer Name & Email per Repository

If you want to use special settings only when working in a certain repository, you can simply omit the --global flag. This makes the configuration valid only in that repository:

$ git config user.name "John Doe"

$ git config user.email [john@doe.org](mailto:john@doe.org)

Deleting local branches in Git

$ git branch -d feature/login

Using the "-d" flag, you tell "git branch" which item you want to delete.

Note that you might also need the "-f" flag if you're trying to delete a branch that contains unmerged changes. Use this option with care because it makes losing data very easy.

Deleting remote branches in Git

To delete a remote branch, we do not use the "git branch" command - but instead "git push" with the "--delete" flag:

$ git push origin --delete feature/login

Undoing the Last Commit

However, of course, there a tons of situations where you really want to undo that last commit. E.g. because you'd like to restructure it extensively - or even discard it altogether!

In these cases, the "reset" command is your best friend:

$ git reset --soft HEAD~1

Reset will rewind your current HEAD branch to the specified revision. In our example above, we'd like to return to the one before the current revision - effectively making our last commit undone.

Note the --soft flag: this makes sure that the changes in undone revisions are preserved. After running the command, you'll find the changes as uncommitted local modifications in your working copy.

If you don't want to keep these changes, simply use the --hard flag. Be sure to only do this when you're sure you don't need these changes anymore.

$ git reset --hard HEAD~1

Reverting a Commit

Using the revert command doesn't delete any commits. Quite the contrary: it creates a new revision that reverts the effects of a specified commit:

The syntax to do this is easy. Just use the revert command and provide the commit you want to "undo":

$ git revert 0ad5a7a6

The value of the first octet in an IP address determines whether theaddress is a Class A, Class B, or Class C address. This value also determinesthe default number of bits in the address that are used to represent thenetwork number and the host number.

If the value of the first octet in an IP address is a number from 1 to126, the address is a Class A address. In this case, the first octet (8bits) specifies the network number, and the last three octets (24 bits)specify the host number. For example, the IP address 102.56.187.5 representsthe network number 102 and the host number 56.187.5.

If the value of the first octet in the IP address is a number from 128to 191, the address is a Class B address. In this case, the first two octets(16 bits) specify the network number, and the last two octets (16 bits)specify the host number. For example, the IP address 154.2.91.240 representsthe network number 154.2 and the host number 91.240.

Finally, if the value of the first octet in the IP address is a numberfrom 192 to 223, the address is a Class C address. In this case, the firstthree octets (24 bits) specify the network number, and the last octet (8bits) specifies the host number. IP addresses with first octet values of0, 127, and 224[shy ]255 are reserved, as explained in the next section.

Because only the first octet in a Class A address is used for the networknumber and the value of the octet must be between 1 and 126, only 126 ClassA networks can exist on any network.

##### Reserved Host Numbers

Although the number of bits used to specify a host number may be 8, 16,or 24 bits, depending on the address class, two host numbers are alwaysreserved:

* The number that results when all of the bits in the host number of the IP address are set to zero (zero)
* The number that results when all of the bits in the host number of the IP address are set to one (a number that varies with the number of bits)

The two reserved host numbers have special uses: An IP address with ahost number of zero is used as the address of the entire network. For example,for a Class C network with the network number 199.60.32, the IP address199.60.32.0 indicates the entire network.

If all of the bits in the host number of the IP address are set to avalue of one, a broadcast message will be sent to every host on the network.For example, suppose that a host on a Class C network with the network addressof 199.60.32.0 wanted to send a message to all of the other hosts on thenetwork. The host would send this message to 199.60.32.255.

After subtracting the reserved host numbers in a Class A network, 16,777,214unique hosts are possible.