Top 50

UNIX

Interview Questions Answers

Knowledge Powerhouse

Top 50
UNIX
Interview Questions & Answers

www.KnowledgePowerhouse.com

Introduction

This book contains top 50 Unix interview questions that are asked in a technical interview. The focus is on commands and concepts inside Unix. It is an important topic for a software developer to know about Unix.

This book is a compilation of Unix interview questions after attending dozens of technical interviews in top-notch companies like- HP, Google, Oracle, Ebay, Amazon etc.

Each question is accompanied with an answer so that you can save your time while preparing for an interview.

The difficulty rating on these Questions varies from a Junior level programmer to Architect level.

Once you go through them in the first pass, mark the questions that you could not answer by yourself. Then, in second pass go through only the difficult questions.

After going through this book 2-3 times, you will be very well prepared to face a technical interview on Unix for an experienced engineer.

Table of Contents

26. What is the use of more command with cat command?

27. What are the File modes in Unix?

28. We wrote a shell script in Unix but it is not doing anything. What could be the reason?

29. What is the significance of 755 in chmod 755 command?

30. How can we run a process in background in Unix? How can we kill a process running in background?

31. How will you create a read only file in Unix?

32. How does alias work in Unix?

33. How can you redirect I/O in Unix?

34. What are the main steps taken by a Unix Shell for processing a command?

35. What is a Sticky bit in Unix?

36. What are the different outputs from Kill command in Unix?

37. How will you customize your environment in Unix?

38. What are the popular commands for user management in Unix?

39. How will you debug a shell script in Unix?

40. What is the difference between a Zombie and Orphan process in Unix?

41. How will you check if a remote host is still alive?

42. How will you get the last executed command in Unix?

43. What is the meaning of "2>&1" in a Unix shell?

44. How will you find which process is taking most CPU time in Unix?

45. What is the difference between Soft link and Hard link in Unix?

46. How will you find which processes are using a file?

47. What is the purpose of nohup in Unix?

48. How will you remove blank lines from a file in Unix?

49. How will you find the remote hosts that are connecting to your system on a specific port in Unix?

50. What is xargs in Unix?

References

1. How will you remove all files in current directory? Including the files that are two levels down in a sub-directory.

In Unix we have rm command to remove files and sub-directories. With rm command we have –r option that stands for recursive. The –r option can delete all files in a directory recursively.

It means if we our current directory structure is as follows:

My_dir
->Level_1_dir
-> Level_1_dir ->Level_2_dir
-> Level_1_dir ->Level_2_dir->a.txt

With rm –r * command we can delete the file a.txt as well as sub-directories Level_1_dir and Level_2_dir.

Command:
rm – r *

The asterisk (*) is a wild card character that stands for all the files with any name.

2. What is the difference between the –v and –x options in Bash shell scripts?

In a BASH Unix shell we can specify the options –v and –x on top of a script as follows:

#!/bin/bash -x –v

With –x option BASH shell will echo the commands like for, select, case etc. after substituting the arguments and variables. So it will be an expanded form of the command that shows all the actions of the script. It is very useful for debugging a shell script.

With –v option BASH shell will echo every command before substituting the values of arguments and variables. In –v option Unix will print each line as it reads.

In –v option, If we run the script, the shell prints the entire file and then executes. If we run the script interactively, it shows each command after pressing enter.

3. What is a Filter in Unix command?

In Unix there are many Filter commands like- cat, awk, grep, head, tail cut etc.

A Filter is a software program that takes an input and produces an output, and it can be used in a stream operation.

E.g. cut -d : -f 2 /etc/passwd | grep abc

We can mix and match multiple filters to create a complex command that can solve a problem.

Awk and Sed are complex filters that provide fully programmable features.

Even Data scientists use Unix filters to get the overview of data stored in the files.

4. What is Kernel in Unix operating system?

Kernel is the central core component of a Unix operating system (OS).

A Kernel is the main component that can control everything within Unix OS.

It is the first program that is loaded on startup of Unix OS. Once it is loaded it will manage the rest of the startup process.

Kernel manages memory, scheduling as well as communication with peripherals like printers, keyboards etc.

But Kernel does not directly interact with a user. For a new task, Kernel will spawn a shell and user will work in a shell.

Kernel provides many system calls. A software program interacts with Kernel by using system calls.

Kernel has a protected memory area that cannot be overwritten accidentally by any process.

5. What is a Shell in Unix OS?

Shell in Unix is a user interface that is used by a user to access Unix services.

Generally a Unix Shell is a command line interface (CLI) in which users enter commands by typing or uploading a file.

We use a Shell to run different commands and programs on Unix operating system.

A Shell also has a command interpreter that can take our commands and send these to be executed by Unix operating system.

Some of the popular Shells on Unix are: Korn shell, BASH, C shell etc.

6. What are the different shells in Unix that you know about?

Unix has many flavors of Shell. Some of these are as follows:

- Bourne shell: We use sh for Bourne shell.
- Bourne Again shell: We use bash to run this shell.
- Korn shell: We can use ksh to for Korn shell.
- Z shell: The command to use this is zsh
- C shell: We use csh to run C shell.
- Enhanced C shell: tcsh is the command for enhanced C shell.

7. What is the first character of the output in ls –l command ?

We use ls -l command to list the files and directories in a directory. With -l option we get long listing format.

In this format the first character identifies the entry type. The entry type can be one of the following:

b	Block special file
c	Character special file
d	Directory
l	Symbolic link
s	Socket link
p	FIFO
-	Regular file

In general we see d for directory and - for a regular file.

8. What is the difference between Multi-tasking and Multi-user environment?

In a Multi-tasking environment, same user can submit more than one tasks and operating system will execute them at the same time.

In a Multi-user environment, more than one user can interact with the operating system at the same time.

3. What is Command Substitution in Unix?
Command substitution is a mechanism by which Shell passes the output of a command as an argument to another command. We can even use it to set a variable or use an argument list in a for loop.

E.g. rm `cat files_to_delete`
In this example files_to_delete is a file containing the list of files to be deleted. cat command outputs this file and gives the output to rm command. rm command deletes the files.

In general Command Substitution is represented by back quotes `.

9. What is an Inode in Unix?

An Inode is a Data Structure in Unix that denotes a file or a directory on file system. It contains information about file like- location of file on the disk, access mode, ownership, file type etc.

Each Inode has a number that is used in the index table. Unix kernel uses Inode number to access the contents of an Inode.

We can use ls -i command to get the inode number of a file.

10. What is the difference between absolute path and relative path in Unix file system?

Absolute path is the complete path of a file or directory from the root directory. In general root directory is represented by / symbol. If we are in a directory and want to know the absolute path, we can use pwd command.

Relative path is the path relative the current location in directory.

E.g. In a directory structure /var/user/kevin/mail if we are in kevin directory then pwd command will give absolute path as /var/user/kevin.

Absolute path of mail folder is /var/user/kevin/mail. For mail folder ./mail is the relative path of mail directory from kevin folder.

11. What are the main responsibilities of a Unix Shell?

Some of the main responsibilities of a Unix Shell are as follows:

1. Program Execution: A shell is responsible for executing the commands and script files in Unix. User can either interactively enter the commands in Command Line Interface called terminal or they can run a script file containing a program.

2. Environment Setup: A shell can define the environment for a user. We can set many environment variables in a shell and use the value of these variables in our program.

3. Interpreter: A shell acts as an interpreter for our scripts. It has a built in programming language that can be used to implement the logic.

4. Pipeline: A shell also can hookup a pipeline of commands. When we run multiple commands separated by | pipe character, the shell takes the output of a command and passes it to next one in the pipeline.

5. I/O Redirection: Shell is also responsible for taking input from command line interface (CLI) and sending the output back to CLI. We use >, <, >> characters for this purpose.

12. What is a Shell variable?

A Unix Shell variable is an internal variable that a shell maintains. It is local to that Shell. It is not made available to the parent shell or child shell.

We generally use lower case names for shell variables in C shell.

We can set the value of a shell variable by set command.

E.g. % set max_threads=10

To delete a Shell variable we can use unset command.

To use a Shell variable in a script we use $ sign in front of the variable name.

E.g. echo $max_threads

13. What are the important Shell variables that are initialized on starting a Shell?

There are following important Shell variables that are automatically initialized when a Shell starts:
user:
term:
home:
path:

These Shell variables take values from environment variables.

If we change the value of these Shell variables then the corresponding environment variable value is also changed.

14. How will you set the value of Environment variables in Unix?

We can use 'setenv' command to set the value of environment variables.
E.g. % setenv [Name] [value]
% setenv MAX_TIME 10

To print the value of environment variable we can use 'printenv' command.
E.g. % printenv MAX_TIME

If we just use printenv then it lists all the environment variables and their values.

To unset or delete an environment variable we use unsetenv command.

E.g. % unsetenv MAX_TIME

To use an environment variable in a command we use the prefix $ with the name of variable.

What is the special rule about Shell and Environment variable in Bourne Shell?

In Bourne Shell, there is not much difference between Shell variable and Environment variable.

Once we start a Bourne Shell, it gets the value of environment variables and defines a corresponding Shell variable. From that time onwards the shell only refers to Shell variable. But if a change is made to a Shell variable, then we have to explicitly export it to environment so that other shell or child processes can use it.

Also for Shell variables we use set and unset commands.

15. What is the difference between a System Call and a library function?

System calls are low-level kernel calls. These are handled by the kernel. System calls are implemented in kernel of Unix. An application has to execute special hardware and system dependent instruction to run a System call.

A library function is also a low level call but it is implemented in user space. A library call is a regular function call whose code resides in a shared library.

16. What are the networking commands in Unix that you have used?

Some of the popular networking commands in Unix that we use are as follows:

1. **ping**: We use this command to test the reachability of a host on an Internet Protocol (IP) network.

2. **telnet**: This is another useful command to access another machine on the network. This is command uses Telnet protocol.

3. **tracert**: This is short for Traceroute. It is a diagnostic command to display the route and transit delays of packets across Internet Protocol.

4. **ftp**: We use ftp commands to transfer files over the network. ftp uses File Transfer Protocol.

5. **su**: This unix command is used to execute commands with the privileges of another user. It is also known as switch user, substitute user.

6. **ssh**: This is a secure command that is preferred over Telnet for connecting to another machine. It creates a secure channel over an unsecured network. It uses cryptographic protocol to make the communication secure.

17. What is a Pipeline in Unix?

A Pipeline in Unix is a chain of commands that are connected through a stream in such a way that output of one command becomes input for another command.

E.g. ls –l | grep “abc” | wc –l

In the above example we have created pipeline of three commands ls, grep and wc.

First ls –l command is executed and gives the list of files in a directory. Then grep command searches for any line with word “abc” in it. Finally wc –l command counts the number of lines that are returned by grep command.

In general a Pipeline is uni-directional. The data flows from left to right direction.

18. What is the use of tee command in Unix?

We use tee command in a shell to read the input by user (standard input) and write it to screen (standard output) as well as to a file.

We can use tee command to split the output of a program so that it is visible on command line interface (CLI) as well as stored on a file for later use.

Syntax is tee [-a] [-i] [file …]

19. How will you count the number of lines and words in a file in Unix?

We can use wc (word count) command for counting the number of lines and words in a file. The wc command provides very good options for collecting statistics of a file. Some of these options are:

l : This option gives line count
m : This option gives character count
c : This option gives byte count
w : This option gives word count
L: This option gives the length of the longest line

In case we give more than one files as input to wc command then it gives statistics for individual files as well as the total statistics for all files.

20. What is Bash shell?

Bash stands for Bourne Again Shell. It is free software written to replace Bourne shell.

We can see following line in shell scripts for Bash shell.
#!/bin/bash

In Bash we use ~/.profile at login to set environment variables.

In Bash we can execute commands in batch mode or concurrent mode.

In batch mode commands are separated by semi colon.
% command1; command2

In concurrent mode we separate commands by & symbol.
% command1 & command2

21. How will you search for a name in Unix files?

We can use grep command to search for a name or any text in a Unix file.

Grep stands for Globally search a Regular Expression and Print.

Grep command can search for a text in one file as well as multiple files.

We can also specify the text to be searched in regular expression pattern.

% grep ^z *.txt

Above command searches for lines starting with letter z in all the .txt files in current directory.

22. What are the popular options of grep command in Unix?

In Unix, grep is one of the very useful commands. It provides many useful options. Some of the popular options are:

% grep –i : This option ignores case while doing search.

% grep –x : This option is used to search exact word in a file.

% grep –v: We use this option to find the lines that do not have the text we are searching.

% grep –A 10: This option displays 10 lines after the match is found.

% grep –c: We can use it to count the number of matching lines.

23. What is the difference between whoami and who am i commands in Unix?

Both the commands whoami and who am i are used to get the user information in Unix.

When we login as root user on the network, then both whoami and who am i commands will show the user as root.

But when any other user let say john logs in remotely and runs su –root, whoami will show root, but who am i will show the original user john.

24. What is a Superuser in Unix?

Superuser is a special user account. It is used for Unix system administration. This user can access all files on the file system. Also Superuser can also run any command on a system.

Generally Superuser permission is given to root user.

Most of the users work on their own user accounts. But when they need to run some additional commands, they can use su to switch to Superuser account.

It is a best practice to not use Superuser account for regular operations.

25. How will you check the information about a process in Unix?

We can use ps command to check the status of a process in Unix. It is short for Process Status.

On running ps command we get the list of processes that are executing in the Unix environment.

Generally we use ps –ef command. In this e stands for every process and f stands for full format.

This command gives us id of the process. We can use this id to kill the process.

26. What is the use of more command with cat command?

We generally use cat command to display the contents of a file.

If a file is very big then the contents of the file will not fit in screen, therefore screen will scroll forward and in the end we just see the last page of information from a file.

With more command we can pause the scrolling of data from a file in display. If we use cat command with more then we just see the first page of a file first. On pressing enter button, more command will keep changing the page. In this way it is easier to view information in a file.

When using the cat command to display file contents, large data that does not fit on the screen would scroll off without pausing, therefore making it difficult to view. On the other hand, using the more command is more appropriate in such case because it will display file contents one screen page at a time.

27. What are the File modes in Unix?

In Unix, there are three main permissions for a File.

1. r = It means a user can read the file
2. w = It means that a user can write to this file
3. x = It means the a user can execute a file like a shell script

Further there are three permission sets.

1. Owner: User who created the file
2. Group: This applies to user of a group to which owner belongs
3. Other: This is rest of the users in Unix system

With the combination of these three sets permissions of file in Unix are specified.

E.g. If a file has permissions –rwxr-xr-- , it means that owner has read, write, execute access. Group has read and execute access. Others have just read access. So the owner or admin has to specifically grant access to Others to execute the file.

28. We wrote a shell script in Unix but it is not doing anything. What could be the reason?

After writing a shell script we have to give it execute permission so that it can be run in Unix shell.

We can use chmod command to change the permission of a file in Unix. In general we use chmod +x to give execute permission to users for executing the shell script.

E.g. chmod +x abc.txt will give execute permission to users for executing the file abc.txt.

With chmod command we can also specify to which user/group the permission should be granted. The options are:

1. u is the owner user
2. g is the owner group
3. o is others
4. a is all users

29. What is the significance of 755 in chmod 755 command?

We use chmod command to change the permissions of a file in Unix. In this command we can pass the file permissions in the form of a three-digit number.

In this number 755, first digit 7 is the permissions given to owner, second digit 5 is the permissions of group and third digit 5 is the permissions of all others.

Also the numbers 7 and 5 are made from following rules:
4 = read permission
2 = write permission
1 = execute permission

So 7 = 4 + 2 + 1 = Read + Write + Execute permission
5 = 4 + 1 = Read + Execute permission

In out example 755 means, owner has read, write and execute permissions. Group and others have read and execute permissions.

30. How can we run a process in background in Unix? How can we kill a process running in background?

In Unix shell we can use symbol & to run a command in background.

E.g. % ls –lrt &

Once we use & option it runs the process in background and prints the process ID. We cannot down this process ID for using it in kill command.

We can also use ps –ef command to get the process ID of processes running in background.

Once we know the process ID of a process we can kill it by following command:

% kill -9 processId

31. How will you create a read only file in Unix?

We can create a file with Vi editor, cat or any other command. Once the file is created we have to give read only permissions to file. To change file permission to read only we use following command:

% chmod 400 filename

32. How does alias work in Unix?

We use alias in Unix to give a short name to a long command. We can even use it to combine multiple commands and give a short convenient name.

E.g. alias c=’clear’

With this alias we just need to type c for running clear command.

In bash we store alias in .bash_profile file.

To get the list of all active alias in a shell we can run the alias command without any argument on command line.

```
% alias
alias h='history'
alias ki='kill -9'
alias l='last'
```

33. How can you redirect I/O in Unix?

In Unix we can redirect the output of command or operation to a file instead of command line interface (CLI). For this we sue redirection pointers. These are symbols > and >>.

If we want to write the output of ls –lrt command to a file we use following:
% ls –lrt > fileList.txt

If we want to copy one file to another file we use following:
% cat srcFile > copyFile

If we want to append the contents of one file at the end of another file we use following:
% cat srcFile >> appendToFile

34. What are the main steps taken by a Unix Shell for processing a command?

A Unix Shell takes following main steps to process a command:

1. **Parse**: First step is to parse the command or set of commands given in a Command Line Interface (CLI). In this step multiple consecutive spaces are replaced by single space. Multiple commands that are delimited by a symbol are divided into multiple individual actions.

2. **Variable**: In next step Shell identifies the variables mentioned in commands. Generally any word prefixed by $ sign is a variable.

3. **Command Substitution**: In this step, Shell executes the commands that are surrounded by back quotes and replaces that section with the output from the command.

4. **Wild Card**: Once these steps are done, Shell replaces the Wild card characters like asterisk * with the relevant substitution.

5. **Execute**: Finally, Shell executes all the commands and follows the sequence in which Commands are given in CLI.

35. What is a Sticky bit in Unix?

A Sticky bit is a file/directory permission feature in Unix.

Sometimes when we give write permission to another user then that user can delete the file without the owner knowing about it. To prevent such an accidental deletion of file we use sticky bit.

When we mark a file/directory with a sticky bit, no user other than owner of file/directory gets the privilege to delete a file/directory.

To set the sticky bit we use following command:

% chmod +t filename

When we do ls for a file or directory, the entries with sticky bit are listed with letter t in the end of permissions.

E.g. % ls –lrt

-rwxrwxrwt 5 abc abc 4096 Jan 1 10:10 abc.txt

To remove the sticky bit we use following command:
% chmod –t filename

36. What are the different outputs from Kill command in Unix?

Kill command in Unix can return following outputs:

1. 0: It means Kill command was successful
2. -1: When we get -1 from Kill command it shows that there was some error. In addition to -1 we get EPERM or ESRCH in output.

   EPERM denotes that system does not permit the process to be killed.

   ESRCH denotes that process with PID mentioned in Kill command does not exist anymore. Or due to security restrictions we cannot access that process.

37. How will you customize your environment in Unix?

In Unix, almost all the popular shells provide options to customize the environment by using environment variables. To make these customizations permanent we can write these to special files that are specific to a user in a shell.

Once we write our customizations to these files, we keep on getting same customization when we open a new shell with same user account.

The special files for storing customization information for different shells at login time are:

1. C shell: /etc/.login or ~/.cshrc
2. TC shell: /etc/.login or ~/.tshrc
3. Korn shell: ~etc/ksh.kshrc
4. Bash: ~/.bash_profile

38. What are the popular commands for user management in Unix?

In Unix we use following commands for User Management:

1. **id**: This command gives the active user id with login and groups to which user belongs.

2. **who**: This command gives the user that is currently logged on system. It also gives the time of login.

3. **last**: This command shows the previous logins to the system in a chronological order.

4. **adduser**: We use this command to add a new user.

5. **groupadd**: We use this command to add a new group in the system.

6. **usermod**: We user usermod command to add/remove a user to a group in Unix.

39. How will you debug a shell script in Unix?

A shell script is a program that can be executed in Unix shell. Sometimes a shell script does not work as intended. To debug and find the problem in shell script we can use the options provided by shell to debug the script.

In bash shell there are x and v options that can be used while running a script.

% bash –xv <scriptName>

With option v all the input lines are printed by shell. With option x all the simple commands are printed in expanded format. We can see all the arguments passed to a command with –x option.

40. What is the difference between a Zombie and Orphan process in Unix?

Zombie is a defunct child process in Unix that still has entry in process table.

Sometimes a child process is terminated in Unix, but the parent process still waits on it.

A Zombie process is different from an Orphan process. An orphan process is a child process whose parent process had died. Once a process is orphan it is adopted by init process. So effectively it is not an orphan.

Therefore if a process exits without cleaning its child processes, they do not become Zombie. Instead init process adopts these child processes.

Zombie processes are the ones that are not yet adopted by init process.

41. How will you check if a remote host is still alive?

We can use one of the networking commands in Unix. It is called ping. With ping command we can ping a remote host.

Ping utility sends packets in an IP network with ICMP protocol. Once the packet goes from source to destination and comes back it records the time.

We can even specify the number of packets we want to send so that we collect more statistics to confirm the result.

% ping www.google.com

Another option is to use telnet to remote host to check its status.

42. How will you get the last executed command in Unix?

We can use history command to get the list commands that were executed in Unix. Since we are only interested in the last executed command we have to use tail to get the last entry.

Exact command would be as follows:
% history | tail -2

43. What is the meaning of “2>&1” in a Unix shell?

In Unix shell file descriptor 1 is for standard output.
File description 2 is for standard error.

We can use “2>&1” in a command so that all the errors from standard error go to standard output.

% cat file 2>&1

44. How will you find which process is taking most CPU time in Unix?

In Unix, we can use top command to list the CPU time and memory used by various processes. The top command lists the process IDs and CPU time, memory etc used by top most processes.

Top command keeps refreshing the screen at a specified interval. So we can see over the time which process is always appearing on the top most row in the result of top command.

This is the process that is consuming most CPU time.

45. What is the difference between Soft link and Hard link in Unix?

A soft link is a pointer to a file, directory or a program located in a different location. A hard link can point to a program or a file but not to a directory.

If we move, delete or rename a file, the soft link will be broken. But a hard link still remains after moving the file/program.

We use the command ln –s for creating a soft link. But a hard link can be created by ln command without –s option.

46. How will you find which processes are using a file?

We can use lsof command to find the list of Process IDs of the processes that are accessing a file in Unix.

Lsof stands for List Open Files.

Sample command is:
% lsof /var

It will list the processes that are accessing /var directory in current unix system.

We can use options –i, -n and –P for different uses.

% lsof –i will only list IP sockets.

47. What is the purpose of nohup in Unix?

In Unix, nohup command can be used to run a command in background. But it is different from & option to run a process in background.

Nohup stands for No Hangup. A nohup process does not stop even if the Unix user that started the process has logged out from the system.

But the process started with option & will stop when the user that started the process logs off.

48. How will you remove blank lines from a file in Unix?

We can use grep command for this option. Grep command gives –v option to exclude lines that do not match a pattern.

In an empty line there is nothing from start to end. In Grep command, ^ denotes that start of line and $ denotes the end of line.

% grep –v '^$' lists the lines that are empty from start to the end.

Once we get this result, we can use > operator to write the output to a new file. So exact command will be:

% grep –v '^$' file1.txt > file2.txt

49. How will you find the remote hosts that are connecting to your system on a specific port in Unix?

We can use netstat command for this purpose. Netstat command lists the statistics about network connections. We can grep for the port in which we are interested.

Exact command will be:
% netstst –a | grep “port number”

50. What is xargs in Unix?

We use xargs command to build and execute commands that take input from standard input. It is generally used in chaining of commands.

Xargs breaks the list of arguments into small sub lists that can be handled by a command.

Following is a sample command:

% find /path -type f -print | xargs rm

The above command uses find to get the list of all files in /path directory. Then xargs command passes this list to rm command so that they can be deleted.

Thank You

If you enjoyed this book or gained knowledge from it in any way, then I'd like to ask you for a favor. Would you be kind enough to leave a review for this book on Amazon.com? It'd be greatly appreciated!

References

- https://www.linux.com
- http://www.opengroup.org/unix