Linux Descriptive / case based Question

Q 16 How can you enhance the security of the password file in Linux?

Ans-

It is in the test file named '/etc/passwd' that Linux usually keeps its user account details, including the one-way encrypted passwords. However, this file can be accessed with the help of different tools, which might throw security issues.

To minimize this risk, we will make use of the shadow password format that saves the account details in a regular file /etc/passwd as in the traditional method but with the password stored as a single 'x' character, i.e., it is not the original password that is actually stored in this file. Meanwhile, a second file /etc/shadow will have the encrypted password, along with the other relevant information, such as the account/password expiration date, etc. Most importantly, the latter file is readable only by the root account, and thus it minimizes the security risk.

Q 17 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 adequately. There are roughly 5 methods that determine the total memory used by Linux.

This is explained as below:

Free command: This is the most simple 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 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 memory usage along with other details.

Q 18 How to check which ports are listening in my Linux Server?

Ans-

We have two commands to check which ports are in listening in Linux Server. Following are the two commands

```
# netstat --listen
```

```
# netstat -l
```

Q 19 How to declare and delete variables in bash?

Ans-

The variable can be declared in bash by data type or without data type. If any bash variable is declared without declare command, then the variable will be treated as a string. Bash variable is declared with declare command to define the data type of the variable at the time declaration.

–r, -i, -a, -A, -l, -u, -t and –x options can be used with declare command to declare a variable with different data types.

Example:

```
#!/bin/bash
#Declare variable without any type
num=10

#Values will be combined but not added
result=$num+20
echo $result

#Declare variable with integer type
declare -i num=10

#Values will be added
declare -i result=num+20
echo $result
```

unset command is used to remove any bash variable. The variable will be inaccessible or undefined after using unset command.

Example:

```
#!/bin/bash
str="Linux Hint"
echo $str
unset $str
echo $str
```

Q16 “The Linux firewall has more functionality than windows” Agree or Disagree? Justify your answer.

Ans-

The Linux firewall functionality is provided by Netfilter. Netfilter is far more sophisticated than the Windows Firewall. A firewall worthy of protecting an enterprise can be crafted using a hardened Linux computer and the netfilter firewall, while the Windows Firewall is suitable only for protecting the host on which it resides. The utility that is used to manage the firewall rules and otherwise manipulate the firewall is iptables.

Assuming you are using a default policy action of DROP for your INPUT chain, you can use the following command to add a rule to allow inbound connections to port 22 from the 192.168.1.0 subnet:

```
iptables –A INPUT –p tcp –s 192.168.1.0/24 -–dport 22 –j ACCEPT
```

In the example above, –A tells iptables you want to add a rule. The –p options specifies the protocol to match, while the –s option specifies the sources to match. The --dport option is for the destination port and

–j tells netfilter what action to perform when this rule is matched. You can verify the rules that are currently configured by entering iptables –L; your output should include the following rule:

```
#iptables –L
Chain INPUT (policy DROP)
Target prot opt source destination
ACCEPT tcp -- 192.168.1.0/24 anywhere tcp dpt:ssh
```

Q17 Explain different types of users available in Linux System

Ans-

A user account is a systematic approach to track and monitor the usage of system resources. Each user account contains two unique identifiers; username and UID.

When a user account is created, its username is mapped to a unique UID.

There are three types of user in linux: - root, regular and service.

a) The root user account

This is the main user account in Linux system. It is automatically created during the installation. It has the highest privilege in system. It can do any administrative work and can access any service. This account is intended for system administration and should be used only for this purpose. It should not be used for routine activities. It can't be deleted. But if require, it can be disabled.

b) The regular user account

This is the normal user account. During the installation, one regular user account is created automatically. After the installation, we can create as many regular user accounts as we need. This account has moderate privilege. This account is intended for routine works. It can perform only the tasks for which it is allowed and can access only those files and services for which it is authorized. As per requirement, it can be disabled or deleted.

c) The service account

Service accounts are created by installation packages when they are installed. These accounts are used by services to run processes and execute functions. These accounts are neither intended nor should be used for routine work.

Q18 Explain Conditional Statements in shell programs.

Ans-

Conditional Statements: There are total 5 conditional statements which can be used in bash programming

if statement

if-else statement

if..elif..else..fi statement (Else If ladder)

if..then..else..if..then..fi..fi..(Nested if)

switch statement

Their description with syntax is as follows:

if statement

This block will process if specified condition is true.

Syntax:

```
if [ expression ]
then
   statement
fi
```

if-else statement

If specified condition is not true in if part then else part will be execute.

Syntax

```
if [ expression ]
then
   statement1
else
   statement2
fi
```

if..elif..else..fi statement (Else If ladder)

To use multiple conditions in one if-else block, then elif keyword is used in shell. If expression1 is true then it executes statement 1 and 2, and this process continues. If none of the condition is true then it processes else part.

Syntax

```
if [ expression1 ]
then
   statement1
   statement2
   .
   .
elif [ expression2 ]
then
   statement3
   statement4
   .
```

```
  .
else
  statement5
fi
```

if..then..else..if..then..fi..fi..(Nested if)

Nested if-else block can be used when, one condition is satisfies then it again checks another condition. In the syntax, if expression1 is false then it processes else part, and again expression2 will be check.

Syntax:

```
if [ expression1 ]
then
  statement1
  statement2
  .
else
  if [ expression2 ]
  then
    statement3
    .
  fi
fi
```

switch statement

case statement works as a switch statement if specified value match with the pattern then it will execute a block of that particular pattern

When a match is found all of the associated statements until the double semicolon (;;) is executed.

A case will be terminated when the last command is executed.

If there is no match, the exit status of the case is zero.

Syntax:

```
case  in
  Pattern 1) Statement 1;;
  Pattern n) Statement n;;
esac
```

Q19 Create digital clock using bash script?

Ans-

```
#!/bin/bash
```

```
while true
do
        clear
        echo $(date +%T)
        sleep 1
done
```

Q16 What are system calls? How these are implemented in Linux?

Ans-

A system call is a procedure that provides the interface between a process and the operating system. It is the way by which a computer program requests a service from the kernel of the operating system.

Different operating systems execute different system calls.

In Linux, making a system call involves transferring control from unprivileged user mode to privileged kernel mode; the details of this transfer vary from architecture to architecture. The libraries take care of collecting the system-call arguments and, if necessary, arranging those arguments in the special form necessary to make the system call.

System calls are divided into 5 categories mainly :

1.Process Control

2.File Management

3.Device Management

4.Information Maintenance

5.Communication

1. Process Control :

This system calls perform the task of process creation, process termination, etc.

The Linux System calls under this are fork() , exit() , exec().

fork()

A new process is created by the fork() system call.

A new process may be created with fork() without a new program being run-the new sub-process simply continues to execute exactly the same program that the first (parent) process was running.

It is one of the most widely used system calls under process management.

exit()

The exit() system call is used by a program to terminate its execution.

The operating system reclaims resources that were used by the process after the exit() system call.

exec()

A new program will start executing after a call to exec()

Running a new program does not require that a new process be created first: any process may call exec() at any time. The currently running program is immediately terminated, and the new program starts executing in the context of the existing process.

2.File Management :

File management system calls handle file manipulation jobs like creating a file, reading, and writing, etc. The Linux System calls under this are open(), read(), write(), close().

open():

It is the system call to open a file.

This system call just opens the file, to perform operations such as read and write, we need to execute different system call to perform the operations.

read():

This system call opens the file in reading mode

We can not edit the files with this system call.

Multiple processes can execute the read() system call on the same file simultaneously.

write():

This system call opens the file in writing mode

We can edit the files with this system call.

Multiple processes can not execute the write() system call on the same file simultaneously.

close():

This system call closes the opened file.

3.Device Management :

Device management does the job of device manipulation like reading from device buffers, writing into device buffers, etc. The Linux System calls under this is ioctl().

ioctl():

ioctl() is referred to as Input and Output Control.

ioctl is a system call for device-specific input/output operations and other operations which cannot be expressed by regular system calls.

4.Information Maintenance:

It handles information and its transfer between the OS and the user program. In addition, OS keeps the information about all its processes and system calls are used to access this information. The System calls under this are getpid(), alarm(), sleep().

getpid():

getpid stands for Get the Process ID.

The getpid() function shall return the process ID of the calling process.

The getpid() function shall always be successful and no return value is reserved to indicate an error.

alarm():

This system call sets an alarm clock for the delivery of a signal that when it has to be reached.

It arranges for a signal to be delivered to the calling process.

sleep():

This System call suspends the execution of the currently running process for some interval of time

Meanwhile, during this interval, another process is given chance to execute

5.Communication :

These types of system calls are specially used for inter-process communications.

Two models are used for inter-process communication-

-Message Passing(processes exchange messages with one another)

-Shared memory(processes share memory region to communicate)

The system calls under this are pipe() , shmget() ,mmap().

pipe():

The pipe() system call is used to communicate between different Linux processes.

It is mainly used for inter-process communication.

The pipe() system function is used to open file descriptors.

shmget():

shmget stands for shared memory segment.

It is mainly used for Shared memory communication.

This system call is used to access the shared memory and access the messages in order to communicate with the process.

mmap():

This function call is used to map or unmap files or devices into memory.

The mmap() system call is responsible for mapping the content of the file to the virtual memory space of the process.

These are the various system calls involved in LINUX operating system.

Q17 Mention some ways to perform arithmetic operations in bash?

Ans-

Arithmetic operations can be done in multiple ways in bash. ‘let’, ‘expr’, ‘bc’ and double brackets are the most common ways to perform arithmetic operations in bash. The uses of these commands are shown in the following example.

Example:

```
#!/bin/bash
# Calculating the subtraction by using expr and parameter expansion
var1=$( expr 120 - 100 )
# print the result
echo $var1
# Calculate the addition by using let command
let var2=200+300
# Print the rsult
echo $var2
# Calculate and print the value of division using ‘bc’ to get the result
# with fractional value
echo "scale=2; 44/7" | bc
# Calculate the value of multiplication using double brackets
var3=$(( 5*3 ))
# Print the result
echo $var3
```

Q18 How shell scripts are executed? Explain with example.

Ans-

Steps to write and execute a script

1.Open the terminal. Go to the directory where you want to create your script.

2.Create a file with .sh extension.

3.Write the script in the file using an editor.

4.Make the script executable with command chmod +x <fileName>.

5.Run the script using ./<fileName>.

Note: In the last step you have to mention the path of the script if your script is in other directory.

Example:

Hello World script-

Here we'll write a simple programme for Hello World.

First of all, create a simple script in any editor or with echo. Then we'll make it executable with chmod +x command. To find the script you have to type the script path for the shell.

```
# chmod +x hello_world
./hello_world
```

Look at the above, script echo Hello World is created with echo command as hello_world. Now command chmod +x hello_world is passed to make it executable. We have given the command ./hello_world to mention the hello_world path. And output is displayed.

Q19 How to use command-line arguments in bash?

Ans-

How to use command-line arguments in bash?

Command-line arguments are read by $1, $2, $3…$n variables. Command-line argument values are provided in the terminal when executing the bash script. $1 is used to read the first argument, $2 is used to read the second argument and so on.

Example:

```
#!/bin/bash
#Check any argument is provided or not
if [[ $# -eq 0 ]]; then
   echo "No argument is given."
   exit 0
fi
#Store the first argument value
color=$1
# Print the argument with other string
printf "You favorite color is %s\n" $color.
```

Q1. How would you use the "lsof" command to troubleshoot a network connection issue?

Answer: The "lsof" command can be used to identify which processes have open network connections. To troubleshoot a network connection issue, you can use "lsof -i" to list all processes with network connections. You can then filter the output to identify the process associated with the problematic connection.

Q2. Write commands to execute following:

a.display the current date and time on the command line

Answer: Use the date command. For example, date +"%Y-%m-%d %H:%M:%S" will display the date and time in the format of "year-month-day hour:minute:second".

b.How do you check the system uptime?

Answer: Use the uptime command. It will display the current time, how long the system has been running, the number of users currently logged on, and the system load averages for the past 1, 5, and 15 minutes.

Q3. You have recently installed a new operating system on your server and want to check the system details.

a) Which command will you use to find the uptime of the system?

b) How can you find the system hostname?

Answer:

a) To find the uptime of the system, we can use the "uptime" command.

b) To find the system hostname, we can use the "hostname" command.

Q4.You want to manage running processes on your Linux system.

a) How can you see the list of running processes using the "ps" command?

b) How can you move a process from the foreground to the background using the "bg" command?

Answer:

a) To see the list of running processes, we can use the "ps" command with the "-ef" option.

b) To move a process from the foreground to the background, we can use the "bg" command followed by the job ID of the process.

Q1.What is a signal in Unix-based operating systems, and how can you send a signal to a running process?

Answer: A signal is a software interrupt that can be sent to a running process to notify it of a particular event or to request it to perform a particular action. To send a signal to a running process, you can use the "kill" command with the appropriate signal number or name. For example, "kill -9 1234" sends the "SIGKILL" signal to the process with the PID "1234".

Q2.What is a shebang in shell scripting, and why is it important?

Answer: A shebang is a special character sequence at the beginning of a shell script that specifies the interpreter to use when running the script. For example, "#!/bin/bash" specifies that the script should be run using the Bash shell. It is important because it ensures that the script is executed by the correct interpreter

Q3. Write commands to execute following:

a.How would you use the "netstat" command to display all listening network ports?

b.How would you use the "lsof" command to list all open files owned by a specific user?

Answer:

a) netstat -tln

b) lsof -u [username]

Q4.You want to perform some calculations on your Linux system.

a) How can you perform basic calculations using the "bc" command?

b) How can you use the "cal" command to see the calendar for the current month?

Answer:

a) To perform basic calculations using the "bc" command, we can simply enter the calculations in the terminal and press enter. For example, to perform the calculation "5+3", we would enter "5+3" in the terminal and press enter.

b) To see the calendar for the current month, we can use the "cal" command without any arguments.

Q1. Explain the difference between a process and a thread.

Answer: A process is a running instance of a program, while a thread is a subset of a process that can execute concurrently with other threads in the same process. A process has its own memory space and system resources, while threads share the same memory space and resources within a process.

Q2. How would you use the "ps" command to list all processes that are consuming more than 50% of the CPU?

Answer: You can use the "ps" command with the "axo" options to list all running processes and their CPU usage percentage. To filter the output for processes consuming more than 50% CPU, you can pipe the output to the "awk" command to compare the CPU usage percentage with the threshold value. The command would look like this: "ps axo pid,%cpu | awk '$2 > 50 {print $1}'".

Q3. Write commands to execute following:

a.How would you use the "ps" command to list all processes that are consuming more than 50% of the CPU?

Answer: "ps -eo pid,%cpu | awk '{if ($2 > 50.0) print $1}'"

b.How would you use the "kill" command to terminate a process by its process ID (PID)?

Answer: "kill [PID]"

Q4.Scenario: You want to check the system information and see which operating system you are running.

a) Which command will you use to see the operating system information?

b) How can you see the system's kernel version?

Answer:

a) To see the operating system information, we can use the "uname" command with the "-a" option.

b) To see the system's kernel version, we can use the "uname" command with the "-r" option.

Q16) What are the advantages of using shell scripts in linux?

Shell scripts provide numerous advantage

-Accuracy – The scripts once written, can be implemented again and again, hence reducing the risk of any typing mistake

-Efficient – The shell scripts helps save a lot of time by only executing the file name and all the commands are automatically executed

-Automation – The shell scripts can help us automate a lot of processes

Q17) How to create a shell script in linux?

To create a shell script in Linux-

-Open a text editor and write the shebang line

-Now write the commands that you want to execute in the script.

-Save the file with a .sh extension make it executable using the chmod command.

-Execute the file in the terminal to test

Q18) Create a directory named test_shell. In this directory create a shell file named add_file.sh . In this shell script take input of two numbers from the user and give the sum of the two numbers as output of the shell file execution.

```
mkdir test_shell

cd test_shell

touch add_file.sh

vi add_file.sh


!#/bin/bash
```

```
read num1

read num2

echo $((num1 + num2))


:wq
```

Q19) Create a directory named test_shell. In this directory create a shell file named prod_file.sh. In this shell script take input of two numbers from the user and give the product of the two numbers as output of the shell file execution.

```
mkdir test_shell

cd test_shell

touch prod_file.sh

vi prod_file.sh


!#/bin/bash

read num1

read num2

echo $((num1 \* num2))


:wq
```

Q16) What is a shell script in linux?

A shell script is a text file containing a series of commands that are executed by the shell program in Linux.
These commands can be written on the Linux terminal but in order to automate the whole process of execution of these commands, we use shell scripts as the help to automate this process.

Q17) What is the difference between shell script and shell command in linux?

A shell command is a single command that is executed from the command line. Cd, ls, pwd, mkdir, rmdir, echo etc. all these commands individually are called as Shell commands.
A shell script is a text file containing a series of commands that are executed by the shell program in Linux.

Q18) Create a directory named test_shell. In this directory create a shell file named sub_file.sh . In this shell script take input of two numbers from the user and give the difference of the two numbers as output of the shell file execution.

```
mkdir test_shell
cd test_shell
touch sub_file.sh
vi sub_file.sh

!#/bin/bash
read num1
read num2
echo $((num1 - num2))

:wq
```

Q16) How do you redirect output to a file in a shell script in Linux? What is the purpose of doing it?
To redirect output to a file in a shell script in Linux, you can use the > or >> operators.
The > operator overwrites the contents of the file with the output of the command, while the >> operator appends the output to the end of the file. The purpose of saving the output of the commands in the file is to reuse it later to either compare the solutions or test the commands with those outputs.

Q.1. Write all the steps of shell script, from creating a shell script file till executing it.

Q.2. Write a script that will accept two numbers as input, and outputs the sum and product (as shown below).

```
Enter a number: 5

Enter another number: 2

Sum: 5 + 2 = 7

Product: 5 x 2 = 10
```

Solution:-

```
#!/bin/bash

echo -n "Enter a number : "

read n1
```

```
echo -n "Enter another number : "

read n2

let sum="$n1+$n2"

let pro="$n1*$n2"

echo -e "Sum\t: $n1 + $n2 = $sum"

echo -e "Product\t: $n1 * $n2 = $pro”
```

Q.3. Write a note on nice command with example.

Q. 4. Write a shell script for following, (Write all the commands starting from creation of a shell file till the execution of it.)

a) create a directory named college

b) create a directory named courses within college directory.

c) create 4 files named Engineering, MBA, Hotel management, Pharmacy within courses directory.

d) Write data in particular files as mentioned below, (1 mark)

Engineering- CSE, Mech, EEE, ETE

MBA- HR, Finance

Pharmacy- B.Pharm, D.Pharm

Q.1. What types of operators are used in shell scripting? Enlist all the operators with example.

Q.2. Write a Menu-Driven shell script to perform a simple calculation according to user’s choice, which includes + for Addition, - for Subtraction, * for Multiplication and / for Division with case statement.

Example Input
5
5
+

Example Output
Addition of 5 and 5 is 10

Solution:-

```
#!/bin/bash
echo “Enter two numbers:”
read num1
read num2
read action
case $action in
“+”)
echo “Addition of $num1 and $num2 is `expr $num1 + $num2`”;;
“-”)
echo “Subtraction of $num1 and $num2 is `expr $num1 - $num2`”;;
“*”)
echo “Multiplication of $num1 and $num2 is `expr $num1 \* $num2`”;;
“/”)
echo “Division of $num1 and $num2 is `expr $num1 / $num2`”;;
esac
```

Q.3. What is date command? Explain options of date command with example (Minimum 5 options).

Q.4. Write commands for following,

a) see all the suspended jobs.

b) keep second commands from the above list of suspended jobs in running state in background.

c) bring fourth command to foreground and terminate it.

d) what ‘+’ and ‘-‘ sign represent in jobs.

e) bring second command to foreground with passing a string.

Q.1. Enlist the string operators with example. (Minimum 5)

Q.2. Write a script to accept rating as input and display the message as per the below, using case statement.

User Points Rating

30 to 50 Average

51 to 60 Good

61 to 80 Excellent

81 to 100 Outstanding

For e.g.

input - 77

output - Excellent

Q.3. Explain “which” command and its options and with examples of each.

Q.4. Write commands for following,

a) Display all the network interfaces name.

b) Check all active and inactive network interfaces.

c) Change the mtu to 3000 of the default network interfaces.

d) Change the ip address of the default network interface.

e) Enable promiscuous mode for current network interface.

Q.1. Enlist the different types of shell. (Minimum 5)

Q.2. write a script to read temperature in centigrade and display suitable message according to temperature as stated below,

Temperature less than 0 = Freezing
Temperature 1 to 10 = Very Cold
Temperature 11 to 20 = Cold
Temperature 21 to 30 = Normal
Temperature 31 to 40 = Hot
Temperature greater than equal 40 = Too Hot

For e.g.

input - 42

output – Too hot

Q. 3. Write a note on cal command with examples. (Any 5 options)

Q.4. Write commands for following,

a) ping any website only 5 times.

b) ping any website and set size to 40 bytes.

c) ping any website with an interval time of 3 seconds.

Q.1. What is the syntax of case statement in shell scripting?

Q.3. Write a script to find a eligibility for admission for a course based on following criteria, if marks in any of the subject are not upto the mark then display message “Not eligible”.

Marks in Phy >=65

Marks in Chem >=55

Marks in Maths >=50

For e.g.

input - 72 65 51

output – Eligible.

Q.3. Explain arithmetic operators in bc command with examples?

Q.4. Write the following,

a) shortcut key to clear the terminal screen.

b) shortcut key to suspend a running job.

c) shortcut key to terminate running command.

d) which command is used to pause the terminal for a specific time interval.

e) shortcut key to bring the cursor to the beginning of the line on the terminal.

Q.1. Which commands are used in process management and system monitoring?

Q.2. write a script to accept a x and y co-ordinates from user as input and find a quadrant in which the given they lie,

If it is in 1st Quadrant, then display **Ist Quadrant**, if it is in 2nd Quadrant, then display **IInd Quadrant**,

if it is in 3rd Quadrant, then display **IIIrd Quadrant**,

if it is in 4th Quadrant, then display **IVth Quadrant** and

if it is in center display **Origin**

e.g.

input – 5 6

output – Ist Quadrant.

Q.3. Write a note on ps command in linux.

Q.4. Write a shell script to check whether the entered number is positive or negative or zero.

Q.1. What is bc command in linux. Explain any one operator in it with example.

Q.2. write a script to display largest number among three numbers.

Example

10
19
02

Example output
19 is Largest Number

Q.3. What is bg and fg commands, explain with examples.

Q.4. Write commands for following, (1 Mark each)

a) To check all the running processes.

b) to check the nice value of terminal.

c) to set highest priority to any command.

d) set the nice value of the terminal to 5.

e) to set lowest priority to any command.

Q.1. Which commands are used in troubleshooting in linux?

Q.2. Write a shell script in linux to find whether the entered character is vowel or consonant.

Q.3. Write a note on nice commands, with examples.

Q.4. Write a shell script to determine the type of a triangle depending on the input values of the sides of the triangle. (set8)

Scalene: A triangle where every side is different in length.

Isosceles: A triangle where 2 sides are equal.

Equilateral: A triangle where all sides are equal.