* **Follow the instructions given in each section.**
* **Make sure that you attempt the questions in order.**

**SECTION-A (10\*1 mark=10 marks)**

***(All questions are compulsory)***

Q1 Which command is used to make all files and sub-directories in the directory (prog) executable by all users?

**a) chmod –R a+xprogs**

b) chmod –x a+xprogs

c) chmod -1 a+xprogs

d) chmod –R 222 progs

Q2 The dmesg command

a) Shows user login logoff attempts

b) Shows the syslog file for info messages

**c) Kernel log messages**

d) Shows the daemon log messages

Q3 What command is used to list contents of directories?

a) tar

b) dir

c) lp

**d) ls**

Q4 What command is used to remove files?

a) dm

**b) rm**

c) delete

d) erase

Q5 What special character is used to designate the users home directory in the C shell?

a) @

b) ~

**c) &**

d) $

Q6 Similar to read and write system calls

a) Send

b) Recv

**c) (a) and (b) both**

d) None

Q7 Which command is used to set terminal IO characteristic?

a) tty

b) ctty

c) ptty

**d) stty**

Q8 Which command is used to record a user login session in a file ?

a) macro

b) read

**c) script**

d) none of the above

Q9 pwd command displays

a) user password

b) password file content

**c) present working directory**

d) none of the mentioned

Q10 The command which transcribes the standard input to the standard output and also makes

a copy of the same in a file is

a) tee

b) sort

**c) tr**

d) grep

**SECTION-B (5\*2 mark=10 marks)**

***(All questions are compulsory)***

Q11 If you are a root user, how can you grand execute permission only for the owner of the file

project1?

a) chmod +x project1

**b) chmod u+x project1**

c) chmod a+x project1

d) chmod U+X project1

Q12 What is the output of this program?

#!/bin/bash

for i in 2 3 7

do

echo "Sanfoundry"

done

exit 0

**a) ‘Sanfoundry’ will print 3 times**

b) nothing will print

c) program will generate an error message

d) none of the mentioned

Q13 The 32-bit internet address 10000000 00001010 00000010 00011110 will be written in

dotted decimal notation as

**a) 128.10.2.30**

b) 210.20.2.64

c) 164.100.9.61

d) 148.20.2.30

Q14 A user issues the following command sequence:

$ a.out &

$ bash

$ a.out &

If the user kills the bash process, then which of the following is true?

a) the second a.out process is also terminated

b) the second a.out process becomes a defunct process

c) the first a.out process becomes a zombie process

**d) init process becomes parent of second a.out process**

Q15 How do you print the lines between 5 and 10, both inclusive

**a) cat filename | head | tail -6**

b) cat filename | head | tail -5

c) cat filename | tail +5 | head

d) cat filename | tail -5 | head -10

**SECTION-C() (4x5 marks=20 marks)**

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 ls

done