**PART 3**

**Q1**. Create compress a input.txt to output.zip

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#!/bin/bash

touch "input.txt"

echo "Somestuff">>input.txt

zip -v output.zip input.txt

# v is verbose flag in the zip.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Q2.** Identify the IP address of your machine. Check the routes of machine which has IP address “172.27.16.154”.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#!/bin/bash

hostname -I

ip route get 172.27.16.154

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Q3.** Check if a remote host 172.27.16.154 is alive or not. Find which hosts are connecting to your host on a particular port 10123?

-------------------

#!/bin/bash

if ping -c 1 -W 1 172.27.16.154 ; then

echo "172.27.16.154 is alive";

else

echo "172.27.16.154 is not alive";

fi

# Please check this.Isko extra permissions lag sakti hai run karte vaqt.

# Aur red vala part bhi nahi hai

----------------

**Q4.** Find whether your system is 32 bit or 64 bit? How do you see

command line history?

# Not a Bash script

-------------------

getconf LONG\_BIT

# The above command checks the kernel version (64 or 32).

history 10

# The above command gets you history for last 10 lines.

#Can run without the additional no. of lines parameter

----------------

**Q5.** Display the resource utilization of each running processes in the system. Check how much memory is free and also which process has maximum share in the CPU and memory.

# Not a Bash script

-------------------

# To display the resource utilization sorted according to memory use.

ps -eo pid,ppid,cmd,%mem,%cpu --sort=-%mem

# To display memory usage

free -m

# To display the process with maximum share in CPU

top

# To display the process with maximum memory usage

top -o %MEM

-------------------

**Q6.** How many CPU cores does the machine have? How much memory, and what fraction of it is free? How many context switches has the system performed since bootup? How many processes has it forked since bootup?

# Not a Bash script

-------------------

# To display the no. of CPU cores

nproc

# To display the Total and used memory

free -m

# To display % of memory free

% free | grep Mem | awk '{print $4/$2 \* 100.0}'

# No. of context switches nahi mila. Koi aur dekh lena please.

# To display no of processes forked since bootup

cat /proc/stat | grep ‘processes’

-------------------

**Q7.** Every process consumes some resources (CPU, memory, disk or network bandwidth, and so on). When a process runs as fast as it can, one of these resources is fully utilized, limiting the maximum rate at which the process can make progress. Such a resource is called the bottleneck resource of a process. A process can be bottlenecked by different resources at different points of time, depending on the type of work it is doing.

Run four processes separately, and identify what the bottleneck resource for each is.

# Not a Bash script

# No such script can be created.

#We have to manually monitor resources to find the Bottleneck.

-------------------

# After running the process take it pid

# Command like top helps us monitor CPU, memory etc.

top -p $PID

# Command like iotop helps us monitor Disk usage

iotop -p $PID

# iotop may need root privileges . Please confirm -AJ

-------------------

**Q8.** Every process runs in one of two modes at any time: user mode and kernel mode. It runs in user mode when it is executing instructions / code from the user. It executes in kernel mode when running code corresponding to system calls etc. Compare the programs in terms of the amount of time each spends in the user mode and kernel mode, using information from the file system.

# Not a Bash Script

-------------------

# Command to see execution time of a script

# Aage execution styles ke examples hain. Mat likho to bhi chlega -AJ

time scriptname

# For the ls commands.

time ls

# For a script named new.sh

time ./new.sh

-------------------

**Q9.** Recall that a running process can be interrupted for several reasons. When a process must stop running and give up the processor, it’s CPU state and registers are stored, and the state of another process is loaded. A process is said to have experienced a context switch when this happens. Context switches are of two types: voluntary and involuntary. A process can voluntarily decide to give up the CPU and wait for some event, e.g., disk I/O. A process can be made to give up its CPU forcibly, e.g., when it has run on a processor for too long, and must give a chance to other processes sharing the CPU. The former is called a voluntary context switch, and the latter is called an involuntary context switch. Compare the processes in terms of the number of voluntary and involuntary context switches.

# Not a Bash Script

-------------------

# To see voluntary and non voluntary context switches of a process of pid=pid.

grep ctxt /proc/$pid/status

-------------------

**Q10.** Create a script to check whether the remote host 172.27.16.154

is alive or not. If alive then display the alive message otherwise

display dead.

-------------------

#!/bin/bash

if ping -c 1 -W 1 172.27.16.154 ; then

echo "172.27.16.154 is alive";

else

echo "172.27.16.154 is not alive";

fi

----------------

**Q11.** Create a script to send email message to reciever@gmail.com

with

greeting message.

# The following script needs a well configured SMTP server to run

----------------

#!/bin/bash

# The text file greetings.txt contains the message

cat greetings.txt | mail -s "Greetings" [reciever@gmail.com](mailto:reciever@gmail.com)

----------------