

SRI KRISHNA ARTS AND SCIENCE COLLEGE

Coimbatore-641 008



RECORD NOTE

DEPARTMENT: COMPUTER TECHNOLOGY AND DATA SCIENCE

NAME

REGISTER NUMBER

PROGRAMME: B.Sc COMPUTER TECHNOLOGY **CLASS:** III B.Sc CT-'A'

COURSE: PRACTICAL: LINUX PROGRAMMING

COURSE CODE: 22CTU39

SRI KRISHNA ARTS AND SCIENCE COLLEGE

Coimbatore-641 008



REGISTER NUMBER:

Certified bonafide record of work done by _____
during the year **2024-2025**

Staff-in-charge

Head of the Department

Submitted to Sri Krishna Arts and Science College (Autonomous) End Semester

Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I _____ hereby declare that this record of observations is based on the experiments carried out and recorded by me during the laboratory classes of **“PRACTICAL: LINUX PROGRAMMING”** conducted by SRI KRISHNA ARTS AND SCIENCE COLLEGE, Coimbatore- 641 008.

Date:

Signature of the student:

Name of the Student :

Roll Number :

Countersigned by Staff

TABLE OF CONTENT

S.NO	DATE	TITLE	PAGE NO	SIGN
1		Write a Linux program using basic Shell Commands		
2		Write a Linux Program Using Terminal navigation Commands		
3		Write a Linux Program Using Directory Manipulation Commands		
4		Write a Linux Program Using File Commands		
5		Write a Linux Program Using Conditional Execution Commands		
6		Write a Linux Program Using Variables		
7		Write a Linux Program Using Scheduling Tasks		
8		Write a Linux Program Using Iteration in shell script		
9		Write a bash shell script using Background Process		
10		Write a Linux Program Using Basic Networking Command		
11		Write a Linux Program Using File Transfer Protocol		
12		Write a Linux Program Using Mail Utility		

DATE:

**WRITE A LINUX PROGRAM USING BASIC
SHELL COMMANDS**

EX NO: 01

AIM:

ALGORITHM:

OUTPUT:

```
skasc@administrator: /home
File Edit View Search Terminal Help
skasc@administrator:/home$ pwd
/home
skasc@administrator:/home$ who
skasc      :0                2025-02-27 15:41 (:0)
skasc@administrator:/home$ whoami
skasc
skasc@administrator:/home$ cal
      February 2025
Su Mo Tu We Th Fr Sa
                1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28

skasc@administrator:/home$ date
Thu Feb 27 15:44:08 IST 2025
skasc@administrator:/home$ date +%T
15:44:14
skasc@administrator:/home$ date +%Y
2025
skasc@administrator:/home$ date +%M
44
skasc@administrator:/home$
```

```
skasc@administrator: ~/Desktop
File Edit View Search Terminal Help
skasc@administrator:~$ cd ~/Desktop
skasc@administrator:~/Desktop$ cat languages.sh
java
cobol
c++
javascript
python
c#

skasc@administrator:~/Desktop$ more languages.sh
java
cobol
c++
javascript
python
c#

skasc@administrator:~/Desktop$ head -2 languages.sh
java
cobol
skasc@administrator:~/Desktop$ tail -4 languages.sh
javascript
python
c#
```

RESULT:

DATE:

EX NO:02

**WRITE A LINUX PROGRAM USING
TERMINAL NAVIGATION COMMANDS**

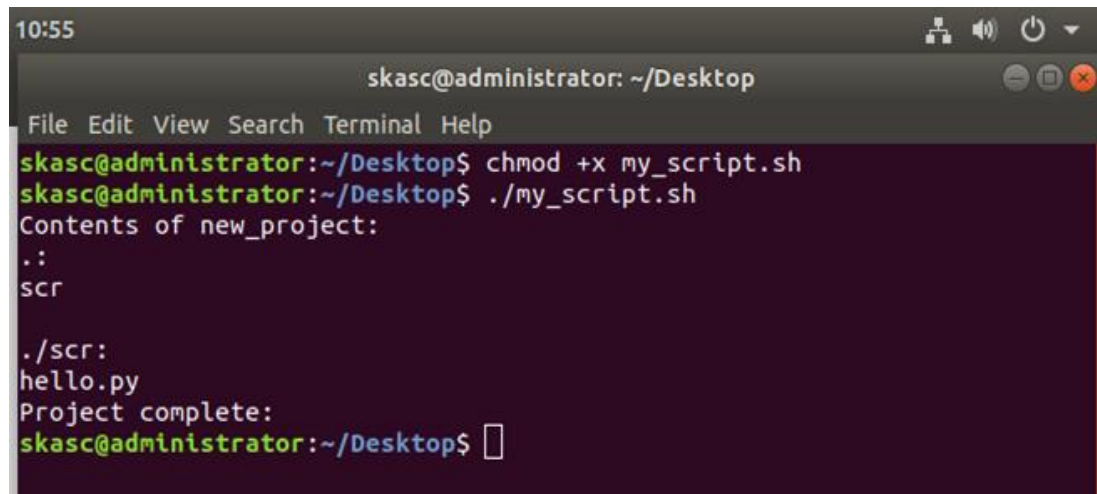
AIM:

ALGORITHM:

CODING:

```
#!/bin/bash
cd ~
mkdir new_project
cd new_project
mkdir scr
echo "print('Hello, World!')">scr/hello.py
echo "Contents of new_project:"
ls -R
echo "Project complete."
```

OUTPUT:

A screenshot of a Linux terminal window. The title bar shows the time as 10:55 and the user as skasc@administrator. The terminal content shows the execution of a script. The prompt is skasc@administrator:~/Desktop. The user enters 'chmod +x my_script.sh'. The prompt changes to skasc@administrator:~/Desktop\$ and the user enters './my_script.sh'. The script outputs 'Contents of new_project:', followed by a recursive listing of the directory: './.' and 'scr'. Then it lists the contents of the 'scr' directory: './scr:', 'hello.py', and 'Project complete:'. The prompt returns to skasc@administrator:~/Desktop\$ with a cursor.

```
10:55
skasc@administrator: ~/Desktop
File Edit View Search Terminal Help
skasc@administrator:~/Desktop$ chmod +x my_script.sh
skasc@administrator:~/Desktop$ ./my_script.sh
Contents of new_project:
.:
scr

./scr:
hello.py
Project complete:
skasc@administrator:~/Desktop$
```

RESULT:

DATE:

**WRITE A LINUX PROGRAM USING
DIRECTORY MANIPULATION**

EX NO:03

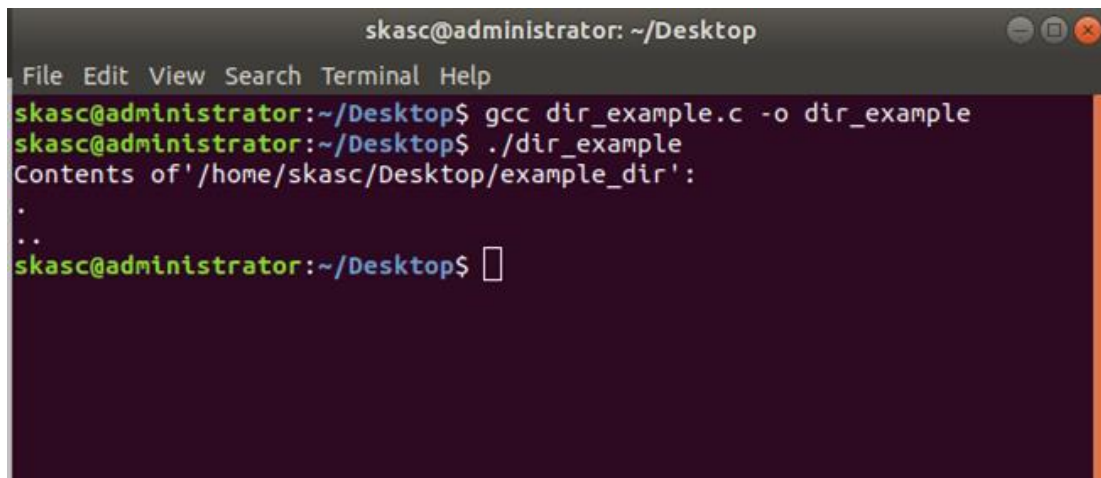
AIM:

ALGORITHM:

CODING:

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/stat.h>
#include <unistd.h>
#include <dirent.h>
int main(){
    mkdir("example_dir",0755);
    chdir("example_dir");
    DIR *d=opendir(".");
    struct dirent*dir;
    printf("Contents of'%s':\n",getcwd(NULL,0));
    while((dir=readdir(d))!=NULL){
        printf("%s\n",dir->d_name);
    }
    closedir(d);
    return 0;
}
```

OUTPUT:

A terminal window titled 'skasc@administrator: ~/Desktop' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the following commands and output:

```
skasc@administrator:~/Desktop$ gcc dir_example.c -o dir_example
skasc@administrator:~/Desktop$ ./dir_example
Contents of '/home/skasc/Desktop/example_dir':
.
..
skasc@administrator:~/Desktop$
```

RESULT:

DATE:

EX NO:04

WRITE A LINUX PROGRAM USING FILE COMMANDS

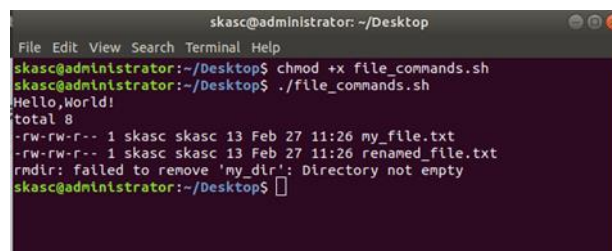
AIM:

ALGORITHM:

CODING:

```
#!/bin/bash
mkdir my_dir && cd my_dir
echo "Hello,World!"> my_file.txt
cat my_file.txt
cp my_file.txt my_file_copy.txt
mv my_file_copy.txt renamed_file.txt
ls -l
rm my_file.txt
cd .. && rmdir my_dir
```

OUTPUT:

A terminal window titled 'skasc@administrator: ~/Desktop' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the following commands and output:

```
skasc@administrator:~/Desktop$ chmod +x file_commands.sh
skasc@administrator:~/Desktop$ ./file_commands.sh
Hello,World!
total 8
-rw-rw-r-- 1 skasc skasc 13 Feb 27 11:26 my_file.txt
-rw-rw-r-- 1 skasc skasc 13 Feb 27 11:26 renamed_file.txt
rmdir: failed to remove 'my_dir': Directory not empty
skasc@administrator:~/Desktop$
```

RESULT:

DATE:

**WRITE A LINUX PROGRAM USING CONDITIONAL
EXECUTION COMMANDS**

EX NO:05

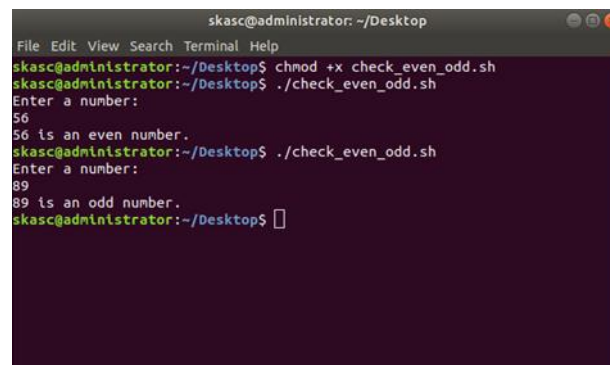
AIM:

ALGORITHM:

CODING:

```
#!/bin/bash
echo "Enter a number: "
read number
if [  $$(number \% 2)$  -eq 0 ]; then # Check if the number is even or odd
    echo "$number is an even number."
else
    echo "$number is an odd number."
```

OUTPUT:

A terminal window titled 'skasc@administrator: ~/Desktop' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the following commands and output:

```
skasc@administrator:~/Desktop$ chmod +x check_even_odd.sh
skasc@administrator:~/Desktop$ ./check_even_odd.sh
Enter a number:
56
56 is an even number.
skasc@administrator:~/Desktop$ ./check_even_odd.sh
Enter a number:
89
89 is an odd number.
skasc@administrator:~/Desktop$
```

RESULT:

DATE:

**WRITE A LINUX PROGRAM USING
VARIABLES**

EX NO:06

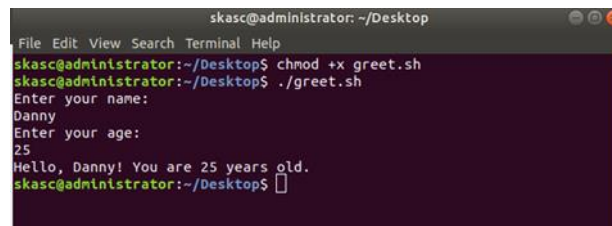
AIM:

ALGORITHM:

CODING:

```
name=""  
age=0  
echo "Enter your name:"  
read name  
echo "Enter your age:"  
read age  
echo "Hello, $name! You are $age years old."
```

OUTPUT:

A terminal window titled 'skasc@administrator: ~/Desktop' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the execution of a shell script. The user runs 'chmod +x greet.sh' and then './greet.sh'. The script prompts for a name, which is 'Danny', and then for an age, which is '25'. Finally, it outputs 'Hello, Danny! You are 25 years old.' and returns to the shell prompt.

```
skasc@administrator: ~/Desktop  
File Edit View Search Terminal Help  
skasc@admtinistrator:~/Desktop$ chmod +x greet.sh  
skasc@admtinistrator:~/Desktop$ ./greet.sh  
Enter your name:  
Danny  
Enter your age:  
25  
Hello, Danny! You are 25 years old.  
skasc@admtinistrator:~/Desktop$
```

RESULT:

DATE:

**WRITE A LINUX PROGRAM USING
SCHEDULING TASKS**

EX NO:07

AIM:

ALGORITHM:

CODING:

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

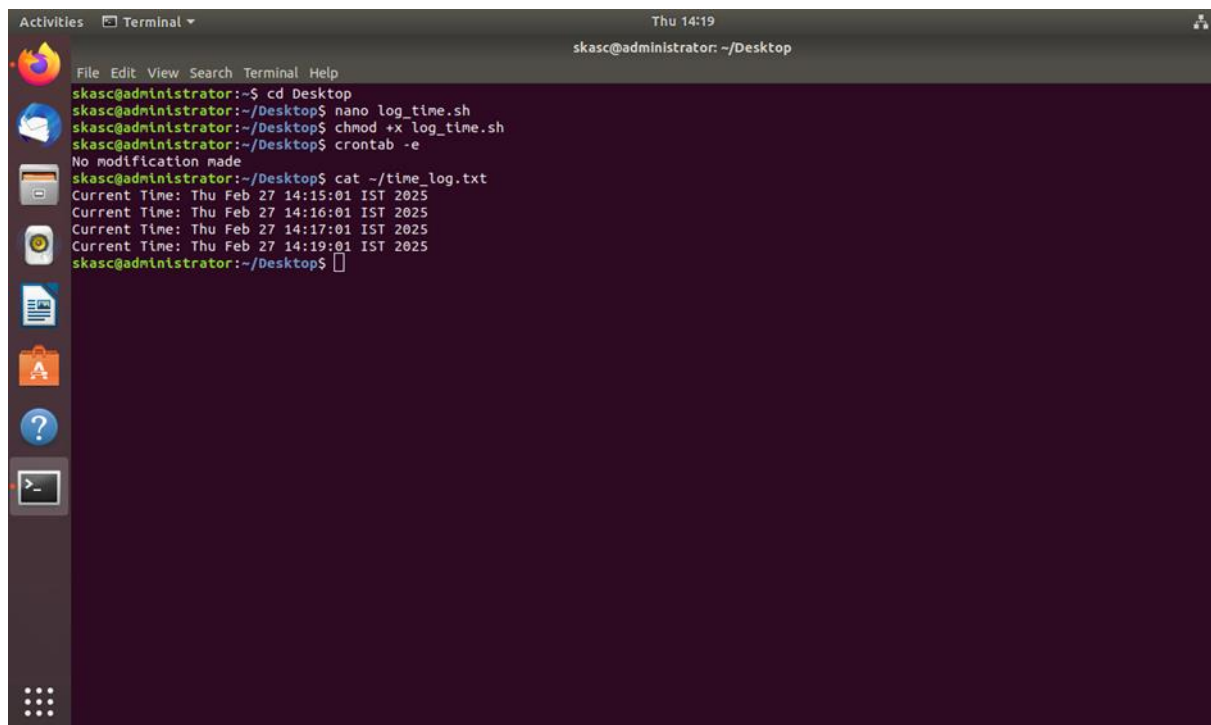
```
# Define the log file
```

```
LOG_FILE="$HOME/time_log.txt"
```

```
# Append the current date and time to the log file
```

```
echo "Current Time: $(date)" >> "$LOG_FILE"
```

OUTPUT:

A terminal window titled 'Terminal' with a dark background and light text. The window shows a series of commands and their outputs. The user is in the directory ~/Desktop. The commands executed are: 'cd Desktop', 'nano log_time.sh', 'chmod +x log_time.sh', 'crontab -e' (which results in 'No modification made'), and 'cat ~/time_log.txt'. The output of the 'cat' command shows four lines of log entries, each stating 'Current Time: Thu Feb 27 14:15:01 IST 2025' through '14:19:01 IST 2025'. The terminal window has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The status bar at the top right shows 'Thu 14:19' and 'skasc@administrator: ~/Desktop'. On the left side of the terminal window, there is a vertical dock with various application icons including a web browser, a file manager, a terminal, and a help icon.

```
Activities Terminal Thu 14:19
skasc@administrator: ~/Desktop
File Edit View Search Terminal Help
skasc@administrator:~$ cd Desktop
skasc@administrator:~/Desktop$ nano log_time.sh
skasc@administrator:~/Desktop$ chmod +x log_time.sh
skasc@administrator:~/Desktop$ crontab -e
No modification made
skasc@administrator:~/Desktop$ cat ~/time_log.txt
Current Time: Thu Feb 27 14:15:01 IST 2025
Current Time: Thu Feb 27 14:16:01 IST 2025
Current Time: Thu Feb 27 14:17:01 IST 2025
Current Time: Thu Feb 27 14:19:01 IST 2025
skasc@administrator:~/Desktop$
```

RESULT:

DATE:

**WRITE A LINUX PROGRAM USING
ITERATION IN SHELL SCRIPT**

EX NO:08

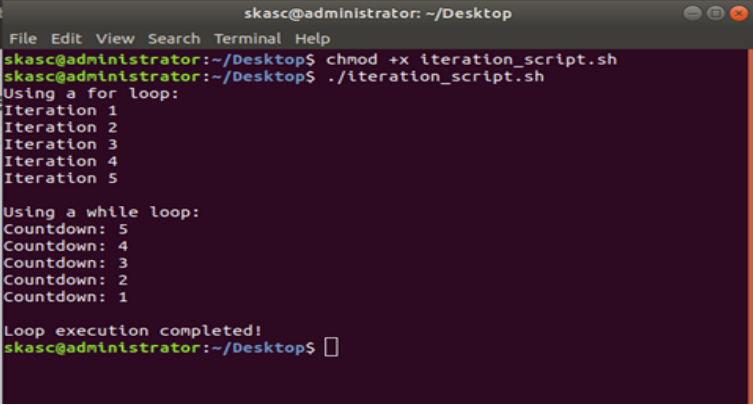
AIM:

ALGORITHM:

CODING:

```
#!/bin/bash
echo "Using a for loop:"
for i in {1..5}; do
echo "Iteration $i"
done
echo -e "\nUsing a while loop:"
# While loop example - Counting down from 5
count=5
while [ $count -gt 0 ]; do
echo "Countdown: $count"
((count--)) # Decrement the counter
done
echo -e "\nLoop execution completed!"
```

OUTPUT:

A terminal window titled 'skasc@administrator: ~/Desktop' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the execution of a script. The first command is 'chmod +x iteration_script.sh'. The second command is './iteration_script.sh'. The output of the script is: 'Using a for loop:', 'Iteration 1', 'Iteration 2', 'Iteration 3', 'Iteration 4', 'Iteration 5', 'Using a while loop:', 'Countdown: 5', 'Countdown: 4', 'Countdown: 3', 'Countdown: 2', 'Countdown: 1', 'Loop execution completed!'. The prompt 'skasc@administrator:~/Desktop\$' is visible at the bottom.

```
skasc@administrator: ~/Desktop
File Edit View Search Terminal Help
skasc@administrator:~/Desktop$ chmod +x iteration_script.sh
skasc@administrator:~/Desktop$ ./iteration_script.sh
Using a for loop:
Iteration 1
Iteration 2
Iteration 3
Iteration 4
Iteration 5
Using a while loop:
Countdown: 5
Countdown: 4
Countdown: 3
Countdown: 2
Countdown: 1
Loop execution completed!
skasc@administrator:~/Desktop$
```

RESULT:

DATE:

**WRITE A BASH SHELL SCRIPT USING
BACKGROUND PROCESS**

EX NO:09

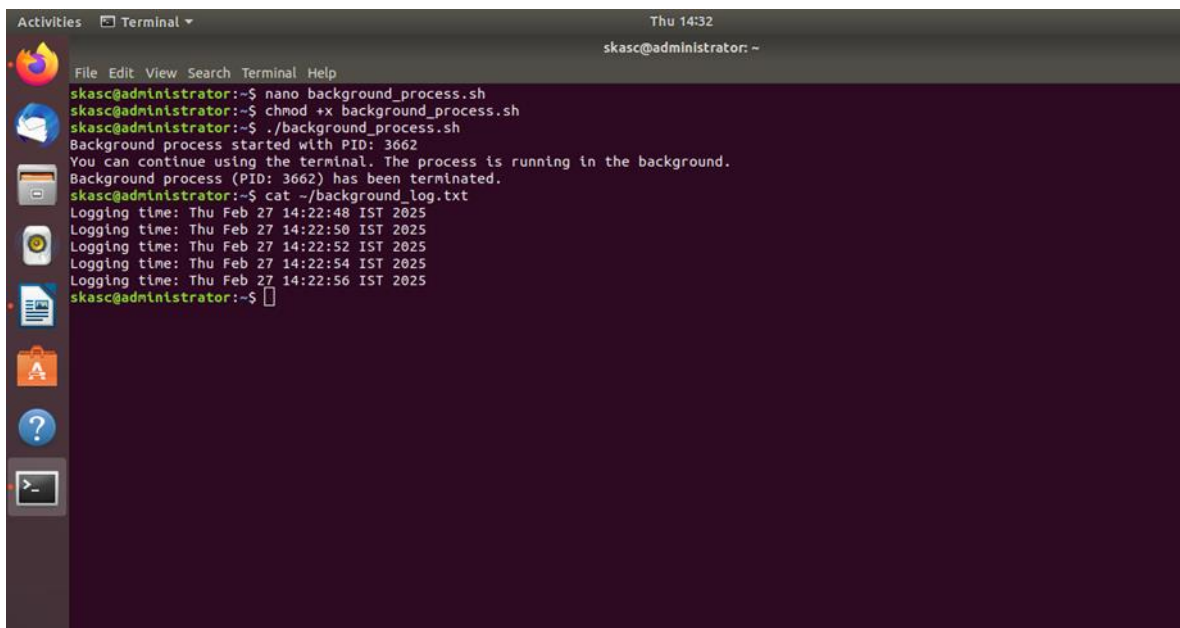
AIM:

ALGORITHM:

CODING:

```
#!/bin/bash
LOG_FILE="$HOME/background_log.txt"
log_time() {
    while true; do
        echo "Logging time: $(date)" >> "$LOG_FILE"
        sleep 2 # Pause for 2 seconds
    done
}
log_time &
BG_PID=$!
echo "Background process started with PID: $BG_PID"
echo "You can continue using the terminal. The process is running in the background."
sleep 10
kill $BG_PID
echo "Background process (PID: $BG_PID) has been terminated."
```

OUTPUT:



The screenshot shows a terminal window titled "Terminal" with a dark background. The user is logged in as "skasc" on a machine named "administrator". The terminal output shows the following sequence of commands and responses:

```
skasc@administrator:~$ nano background_process.sh
skasc@administrator:~$ chmod +x background_process.sh
skasc@administrator:~$ ./background_process.sh
Background process started with PID: 3662
You can continue using the terminal. The process is running in the background.
Background process (PID: 3662) has been terminated.
skasc@administrator:~$ cat ~/background_log.txt
Logging time: Thu Feb 27 14:22:48 IST 2025
Logging time: Thu Feb 27 14:22:50 IST 2025
Logging time: Thu Feb 27 14:22:52 IST 2025
Logging time: Thu Feb 27 14:22:54 IST 2025
Logging time: Thu Feb 27 14:22:56 IST 2025
skasc@administrator:~$
```

RESULT:

DATE:

**WRITE A LINUX PROGRAM USING BASIC
NETWORKING COMMAND**

EX NO:10

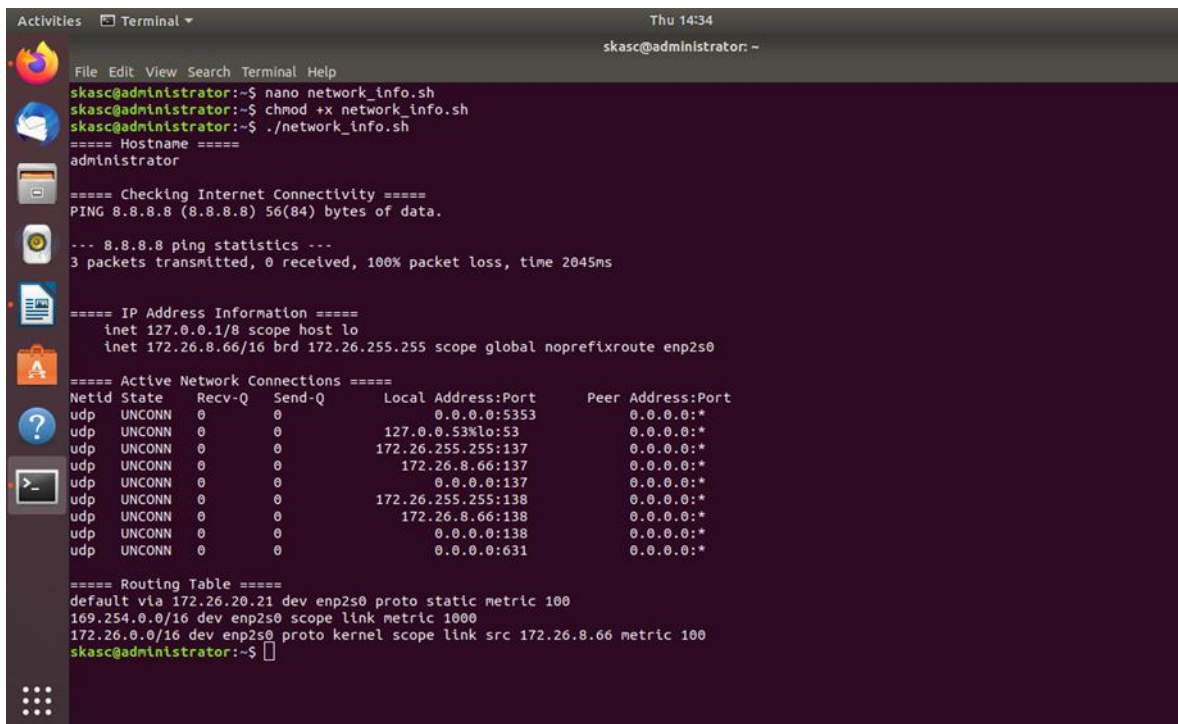
AIM:

ALGORITHM:

CODING

```
#!/bin/bash
echo "===== Hostname ====="
hostname
echo -e "\n===== Checking Internet Connectivity ====="
ping -c 3 8.8.8.8
echo -e "\n===== IP Address Information ====="
ip a | grep "inet "
echo -e "\n===== Active Network Connections ====="
ss -tulnp | head -10
echo -e "\n===== Routing Table ====="
ip route show
```

OUTPUT:



```
skasc@administrator:~$ nano network_info.sh
skasc@administrator:~$ chmod +x network_info.sh
skasc@administrator:~$ ./network_info.sh
===== Hostname =====
administrator

===== Checking Internet Connectivity =====
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
--- 8.8.8.8 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2045ms

===== IP Address Information =====
inet 127.0.0.1/8 scope host lo
inet 172.26.8.66/16 brd 172.26.255.255 scope global noprefixroute enp2s0

===== Active Network Connections =====
Netid State Recv-Q Send-Q Local Address:Port Peer Address:Port
udp UNCONN 0 0 0.0.0.0:5353 0.0.0.0:*
udp UNCONN 0 0 127.0.0.53%lo:53 0.0.0.0:*
udp UNCONN 0 0 172.26.255.255:137 0.0.0.0:*
udp UNCONN 0 0 172.26.8.66:137 0.0.0.0:*
udp UNCONN 0 0 0.0.0.0:137 0.0.0.0:*
udp UNCONN 0 0 172.26.255.255:138 0.0.0.0:*
udp UNCONN 0 0 172.26.8.66:138 0.0.0.0:*
udp UNCONN 0 0 0.0.0.0:138 0.0.0.0:*
udp UNCONN 0 0 0.0.0.0:631 0.0.0.0:*

===== Routing Table =====
default via 172.26.20.21 dev enp2s0 proto static metric 100
169.254.0.0/16 dev enp2s0 scope link metric 1000
172.26.0.0/16 dev enp2s0 proto kernel scope link src 172.26.8.66 metric 100
skasc@administrator:~$
```

RESULT:

DATE:

EX NO:11

**WRITE A LINUX PROGRAM USING FILE
TRANSFER PROTOCOL**

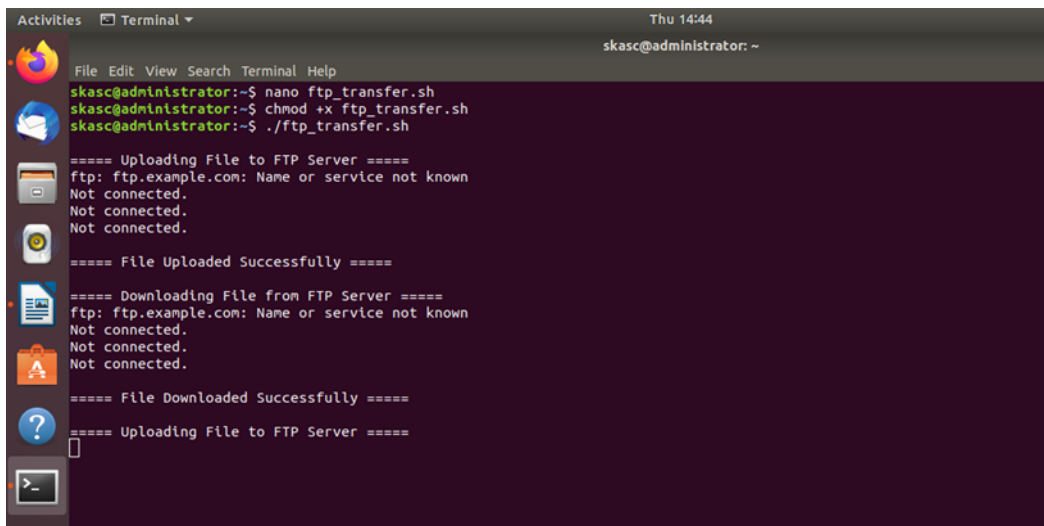
AIM:

ALGORITHM:

CODING

```
#!/bin/bash
FTP_SERVER="ftp.example.com"
FTP_USER="your_username"
FTP_PASS="your_password"
REMOTE_DIR="/remote/path/"
LOCAL_FILE="testfile.txt"
echo "This is a test file for FTP transfer." > $LOCAL_FILE
echo -e "\n===== Uploading File to FTP Server ====="
ftp -inv $FTP_SERVER <<EOF
user $FTP_USER $FTP_PASS
cd $REMOTE_DIR
put $LOCAL_FILE
bye
EOF
echo -e "\n===== File Uploaded Successfully ====="
echo -e "\n===== Downloading File from FTP Server ====="
ftp -inv $FTP_SERVER <<EOF
user $FTP_USER $FTP_PASS
cd $REMOTE_DIR
get $LOCAL_FILE downloaded_$LOCAL_FILE
bye
EOF
echo -e "\n===== File Downloaded Successfully ====="
```

OUTPUT:



```
Activities Terminal Thu 14:44
skasc@administrator: ~
File Edit View Search Terminal Help
skasc@administrator:~$ nano ftp_transfer.sh
skasc@administrator:~$ chmod +x ftp_transfer.sh
skasc@administrator:~$ ./ftp_transfer.sh

===== Uploading File to FTP Server =====
ftp: ftp.example.com: Name or service not known
Not connected.
Not connected.
Not connected.

===== File Uploaded Successfully =====

===== Downloading File from FTP Server =====
ftp: ftp.example.com: Name or service not known
Not connected.
Not connected.
Not connected.

===== File Downloaded Successfully =====

===== Uploading File to FTP Server =====
```

RESULT:

DATE:

**WRITE A LINUX PROGRAM USING MAIL
UTILITY**

EX NO:12

AIM:

ALGORITHM:

CODING

```
#!/bin/bash
# Define email details
TO_EMAIL="recipient@example.com"
SUBJECT="Test Email from Linux"
BODY="Hello,\n\nThis is a test email sent from a Linux shell script using the mail utility.\n\nBest
Regards,\nYour Linux Server"
# Send email
echo -e "$BODY" | mail -s "$SUBJECT" "$TO_EMAIL"
echo "Email has been sent successfully to $TO_EMAIL."
```

OUTPUT:

Email has been sent successfully to recipient@example.com.

If the recipient checks their email, they should see:

Subject: Test Email from Linux

Hello,

This is a test email sent from a Linux shell script using the mail utility.

Best Regards,

Your Linux Server

RESULT: