

**POORNIMA COLLEGE OF ENGINEERING**

**Affiliated to Rajasthan Technical University, Kota and approved by AICTE**

**LAB RECORD**

**Name:** Abhishek Sharma

**Name of the laboratory:** Internet of Things Lab **Lab Code:** 7CS4-21

**Section:** C **Batch:** C1

**Department:** Computer Science and Engineering

**SESSION: 2020-2021**

**POORNIMA COLLEGE OF ENGINEERING, JAIPUR**

**Evaluation Report**

**Name of the laboratory:** Internet of Things Lab **Lab Code:** 7CS4-21

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Expt. No.** | **Name of Experiment** |
| 1. | 1 | Start Raspberry Pi and try various Linux commands in command terminal window:  *ls, cd, touch, mv, rm, man, mkdir, rmdir, tar, gzip, cat, more, less,*  *ps, sudo, cron, chown, chgrp, ping etc.* |
| 2. | 2 | Run some Python programs on Raspberry Pi:  a) Read your name and print Hello message with name.  b) Read two numbers and print their sum, difference, product and division.  c) Word and character count of a given string.  d) Area of a given shape (rectangle, triangle and circle) reading shape and appropriate values from standard input. |
| 3. | 3 | Run some Python programs on Raspberry Pi:  a) Print a name 'n' times, where name and n are read from standard input, using for and while loops.  b) Handle Divided by Zero Exception.  c) Print current time for 10 times with an interval of 10 seconds.  d) Read a file line by line and print the word count of each line. |
| 4. | 4 | a) Light an LED through Python program.  b) Get input from two switches and switch on corresponding LEDs.  c) Flash an LED at a given on time and off time cycle, where the two times are taken from a file. |
| 5. | 5 | Flash an LED based on cron output (acts as an alarm). |

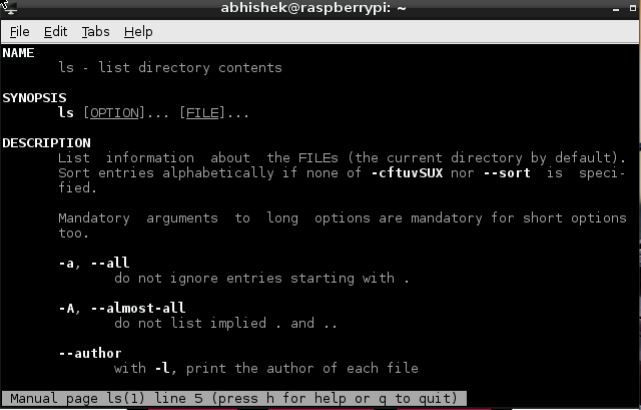
**EXPERIMENT 1**

**Objective:** Start Raspberry Pi and try various Linux commands in command terminal window:

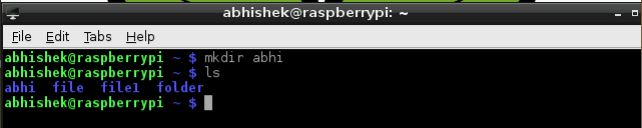
*ls, cd, touch, mv, rm, man, mkdir, rmdir, tar, gzip, cat, more, less, ps, sudo, cron, chown, chgrp, ping etc.*

**1. List Command**

**ls –** This command is used to list the contents of the present working directory on screen.

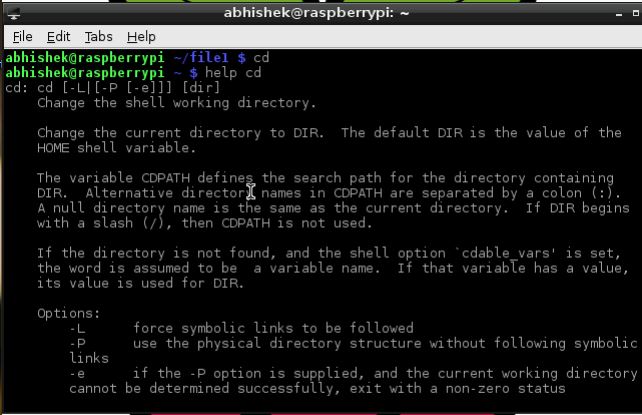


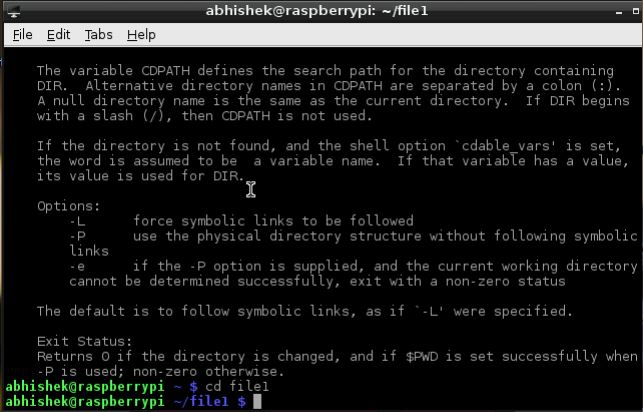




**2. Change Directory Command**

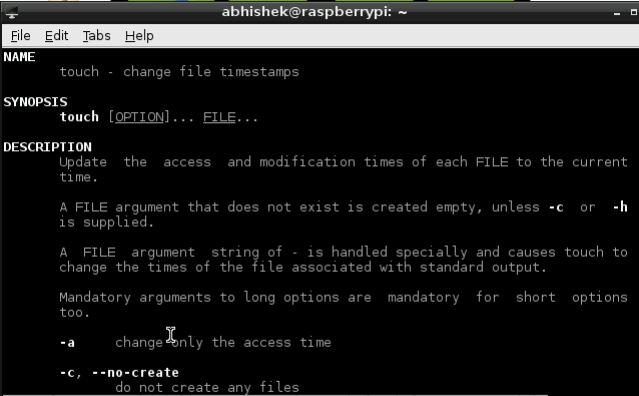
**cd –** To change the present working directory.

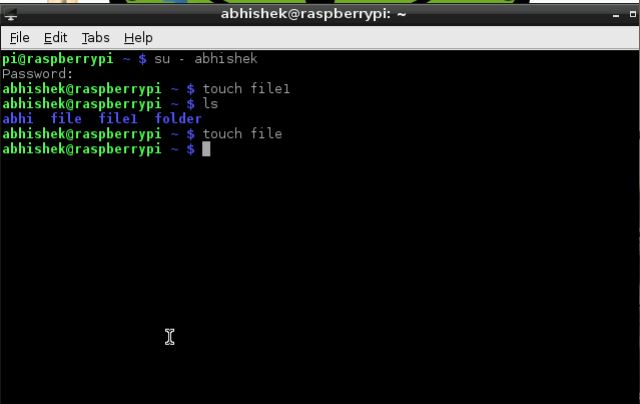




**3. Touch Command**

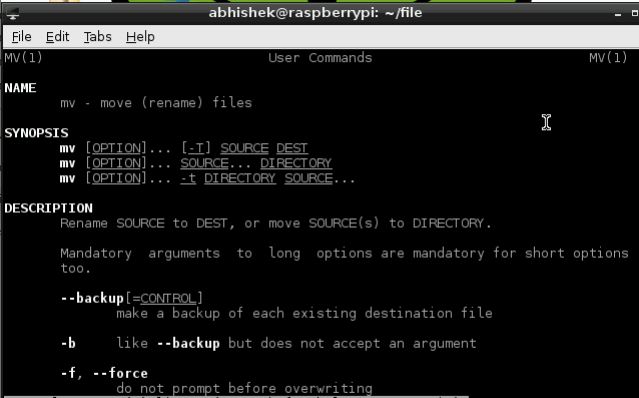
**touch –** This command changes the time stamp of a file that is already created or is to create a new file with current time stamp.

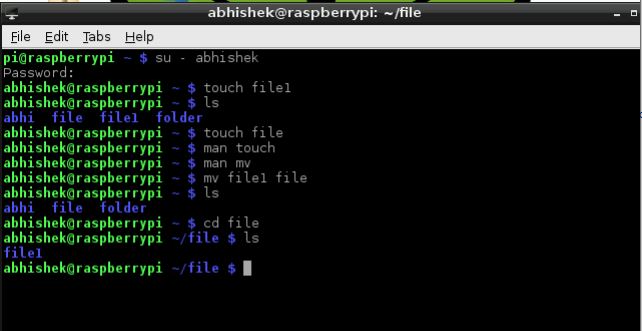




**4. Move Command**

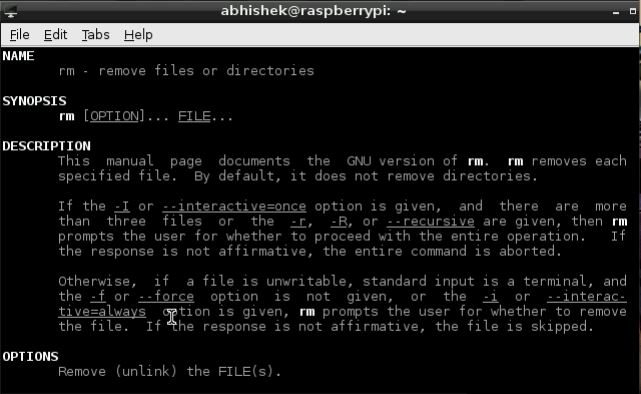
**mv –** Used to move one file into another file or directory.

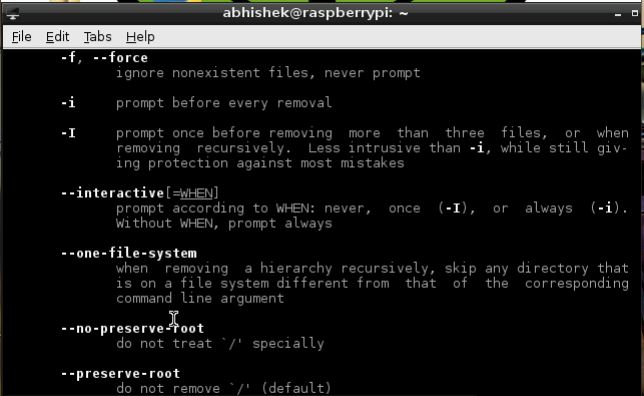


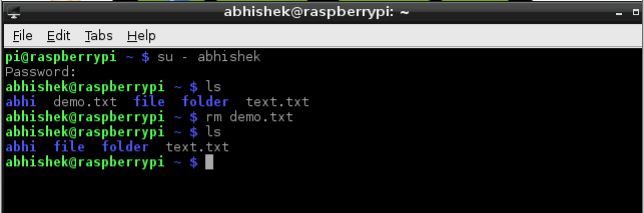


**5. Remove Command**

**rm –** Used to remove a file or a directory.

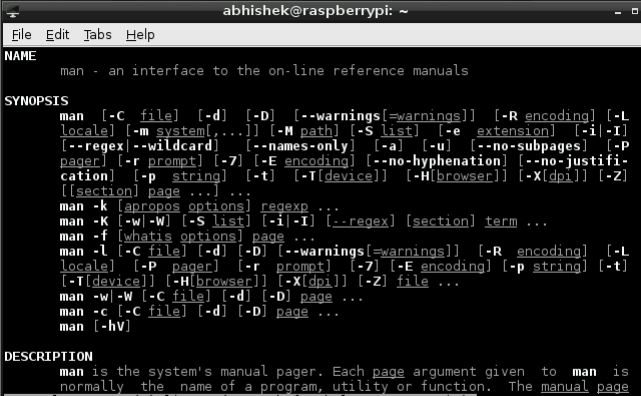


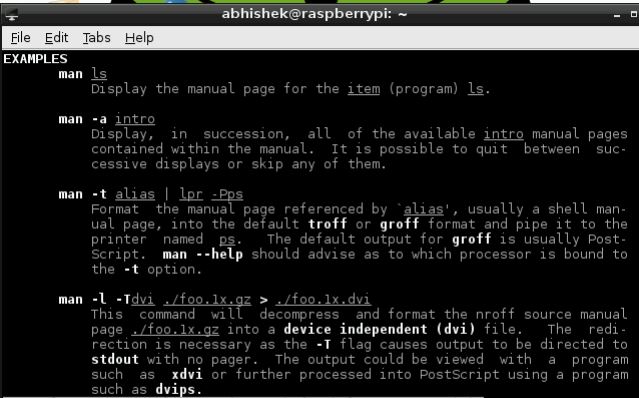


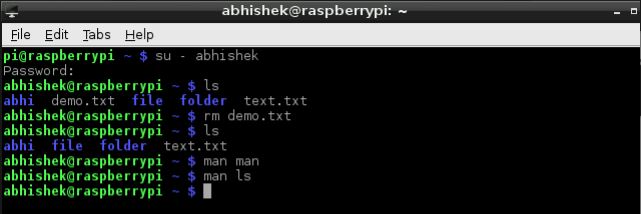


**6. Manual Command**

**man –** Opens an interface that displays the reference manual of any command in terminal.

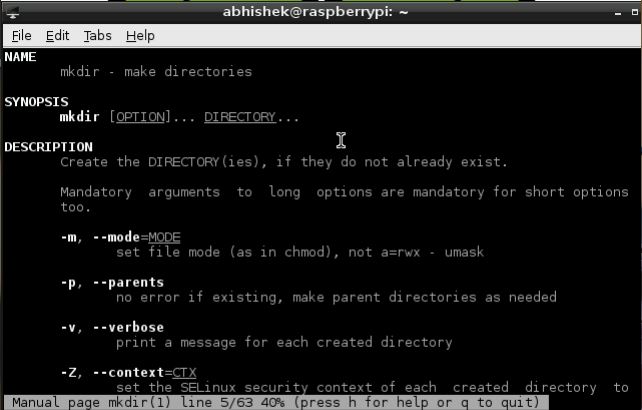


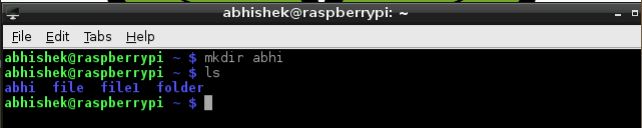




**7. Make Directory Command**

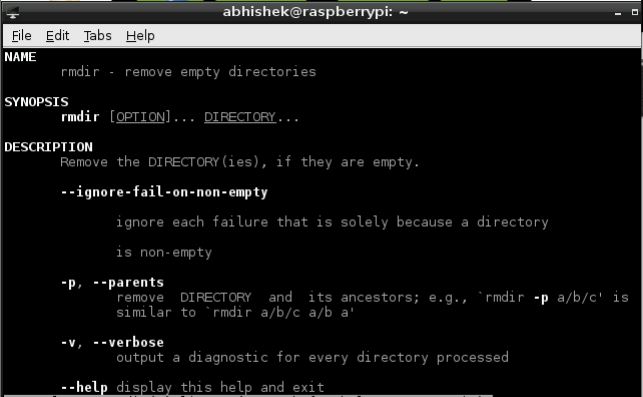
**mkdir –** Used to make directories.





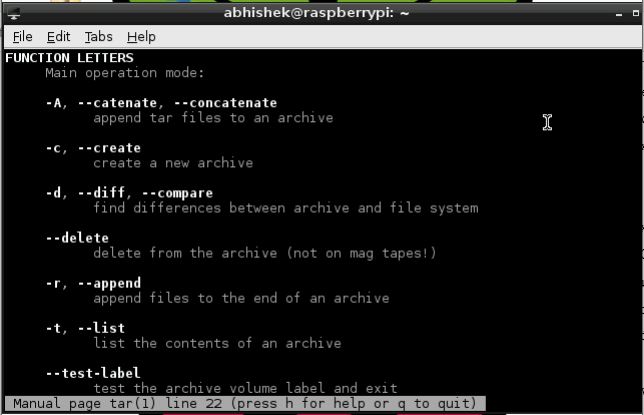
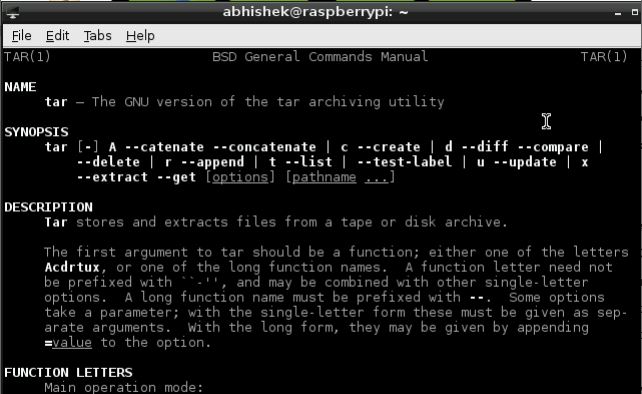
**8. Remove Directory Command**

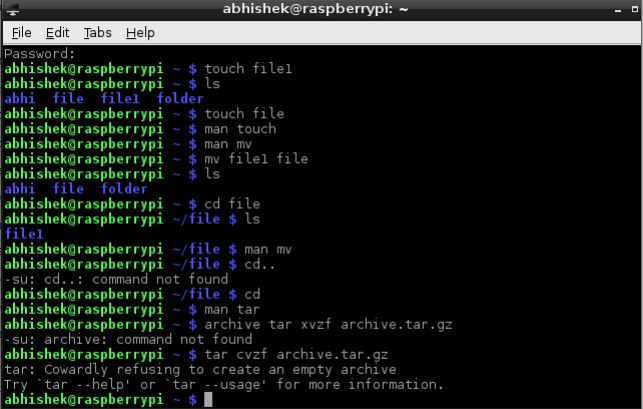
**rmdir –** Used to remove directories.



**9. TAR Command**

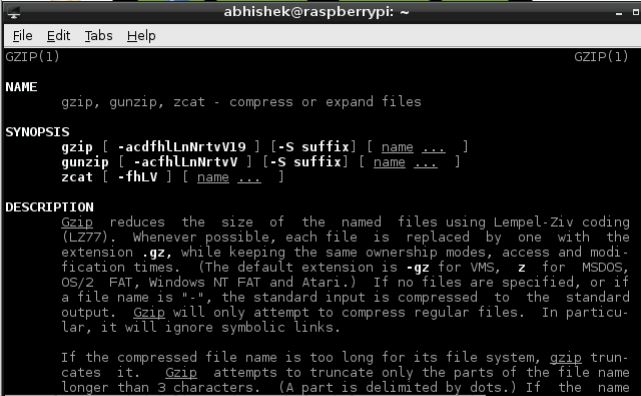
**tar –** Used to rip a collection of files and directories into highly compressed archive file.

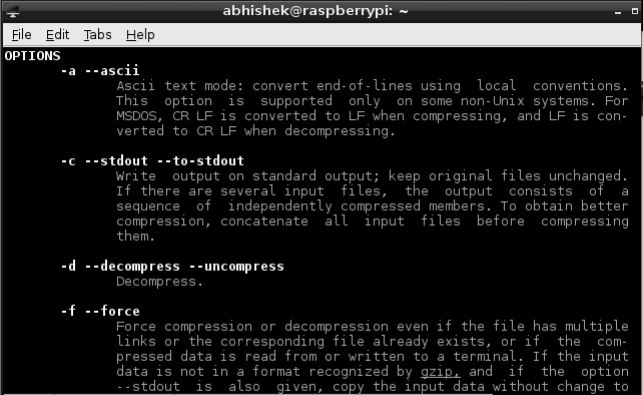




**10. GNU Zip Command**

**gzip –** Used to compress or expand files.

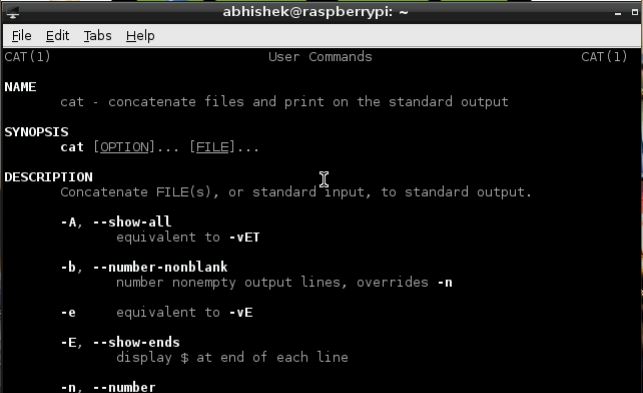


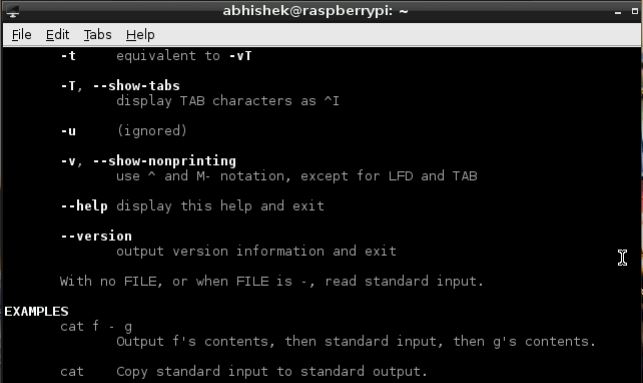


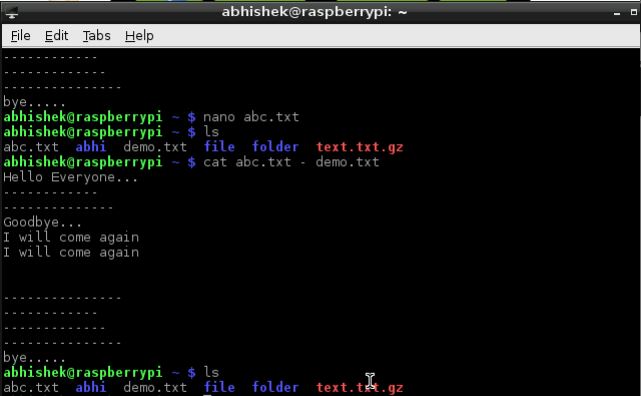


**11. Concatenate Command**

**cat –** Used to concatenate files and print standard output.

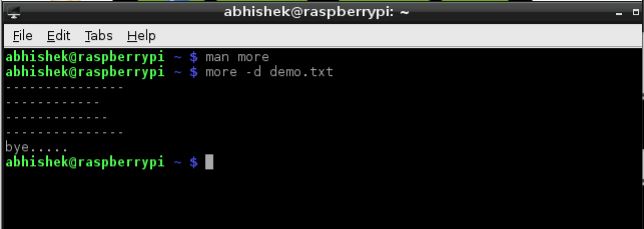
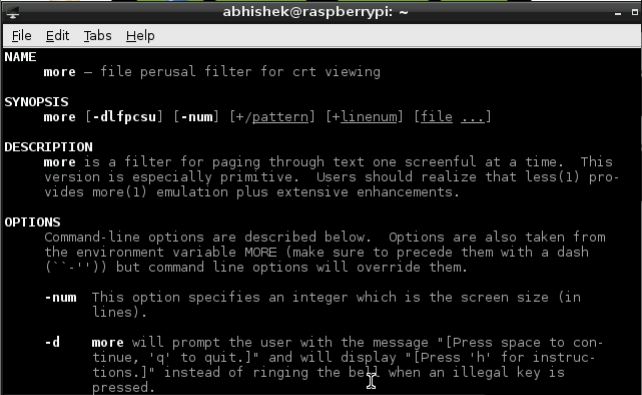






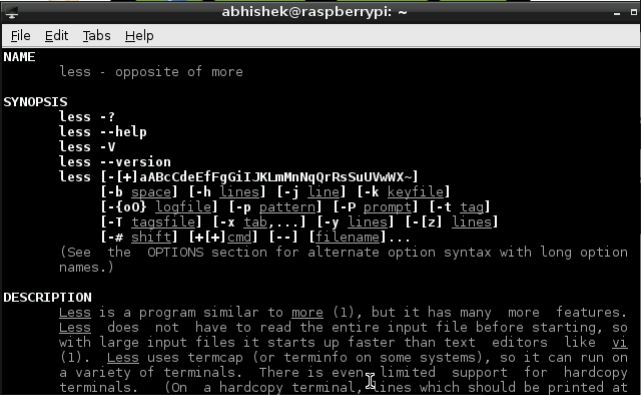
**12. MORE Command**

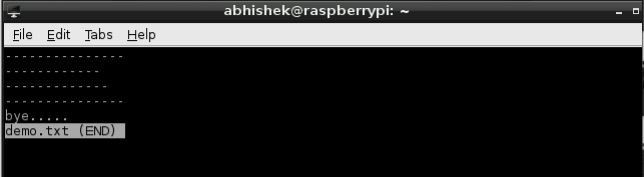
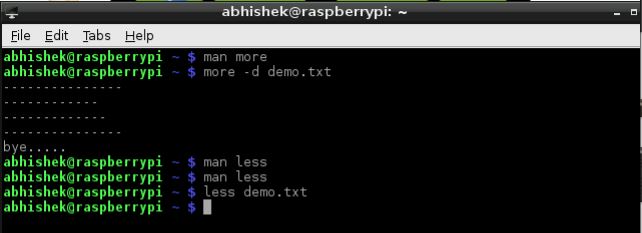
**more –** It is a file perusal filter for crt viewing.



**13. LESS Command**

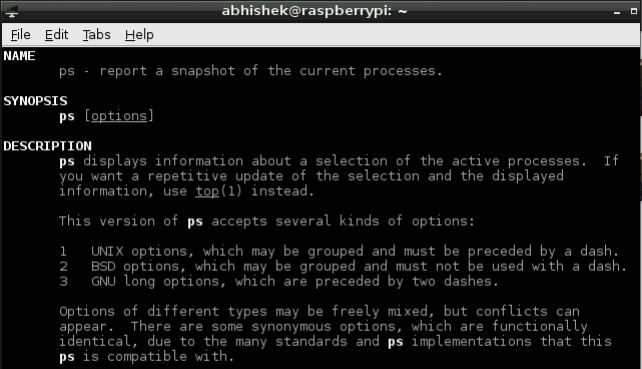
**less –** It is opposite of more.

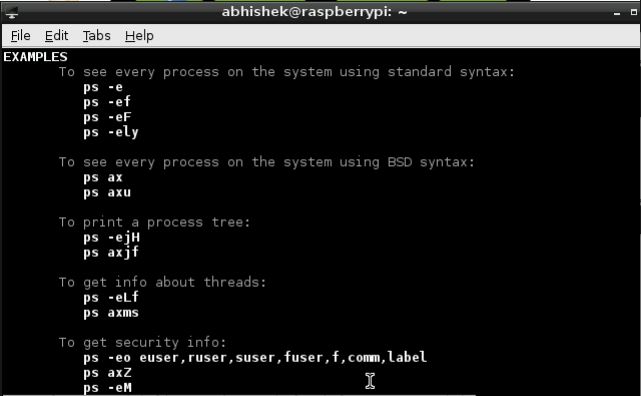


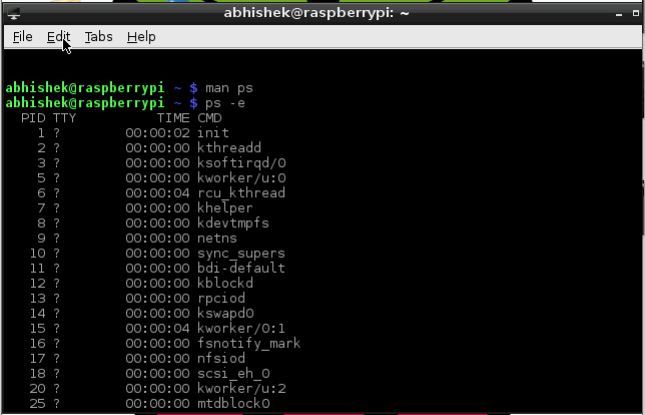


**14. Present Status Command**

**ps –** Used to report a snapshot of present processes.

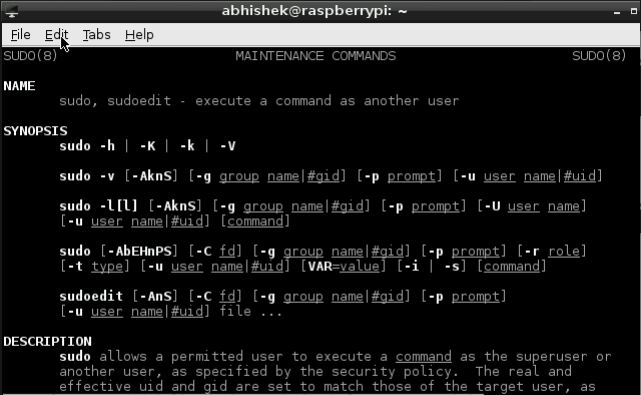


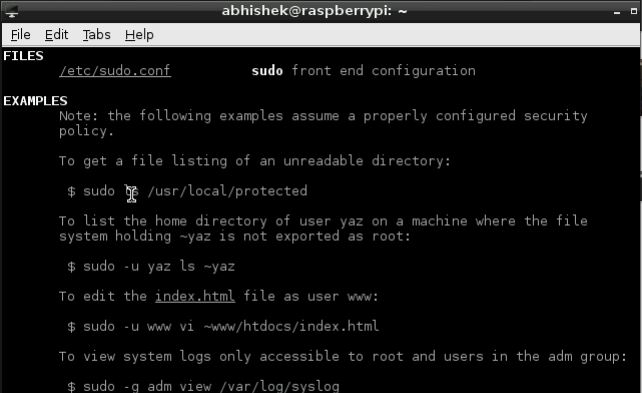


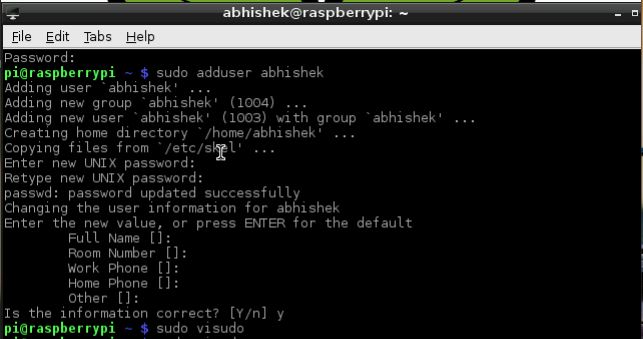


**15. SUDO Command**

**sudo –** allows you to run programs with the security privileges of another user (by default, as the superuser).

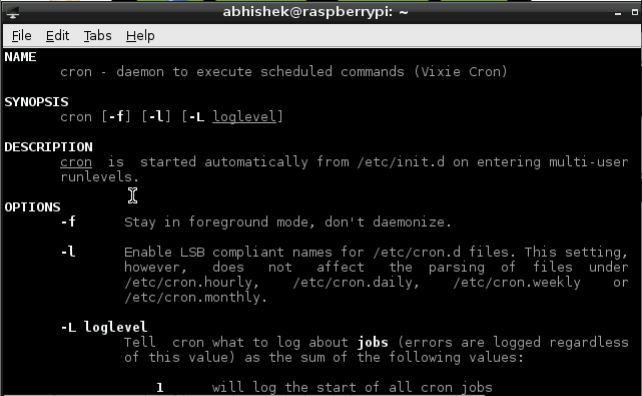


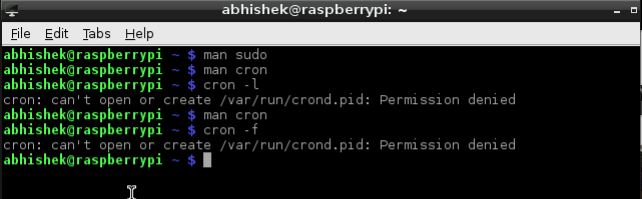




**16. Crontab Command**

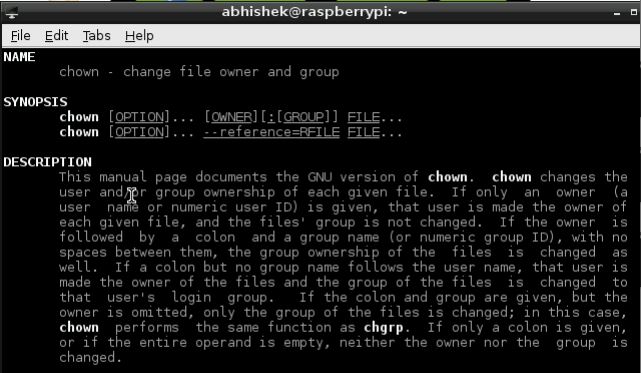
**cron –** Automates the scheduled task at a predetermined time.

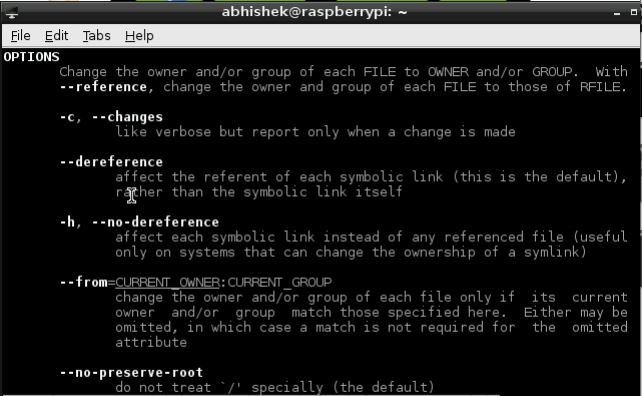


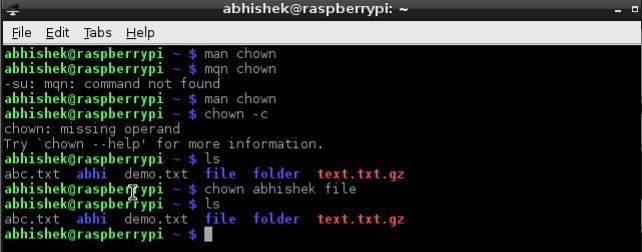


**17. Change Owner/Group Command**

**chown –** allows you to change the user and/or group ownership of a given file, directory, or symbolic link.

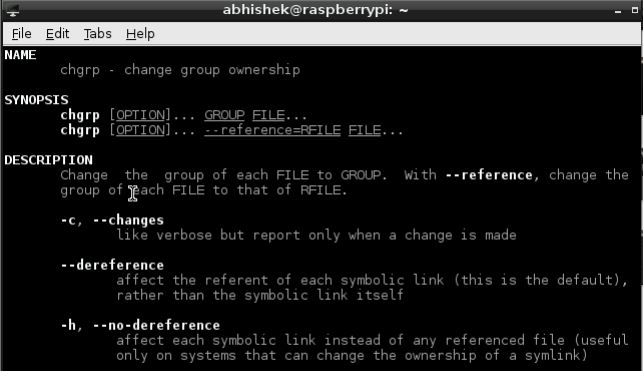


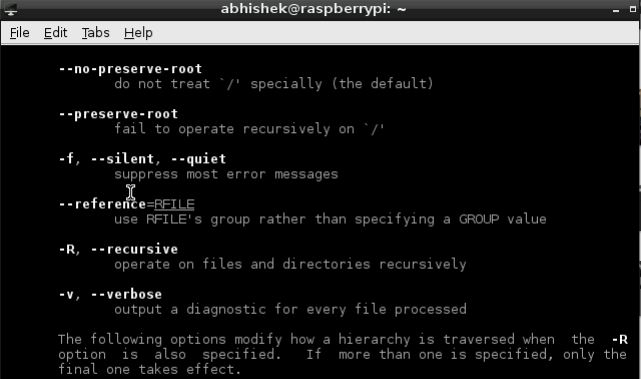


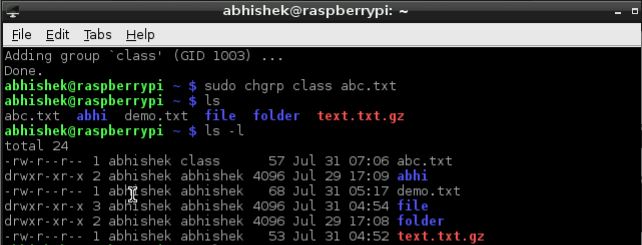


**18. Change Group Ownership Command**

**chgrp –** used to change the group ownership of a file or directory.

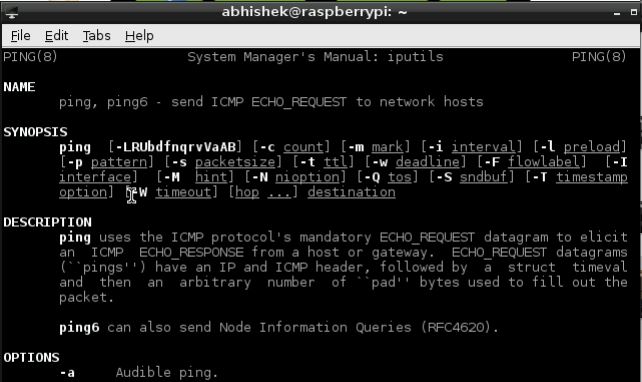


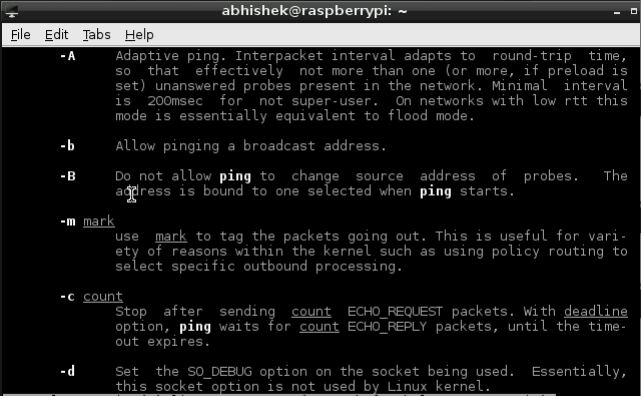


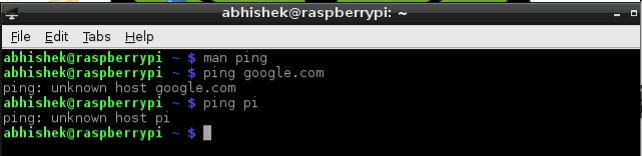


**19. PING Command**

**ping –** one of the most used tools for troubleshooting, testing, and diagnosing network connectivity issues.





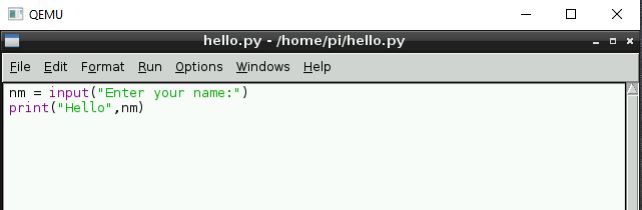


**Result:** Thus, we have learnt to implement basic Linux commands in command terminal window of Raspberry Pi.

**EXPERIMENT 2**

**Run some Python programs on PI like:**

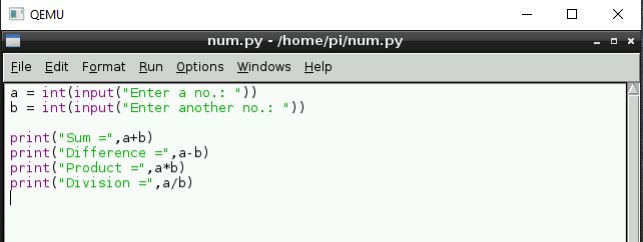
**a)** Read your name and print Hello message with name.



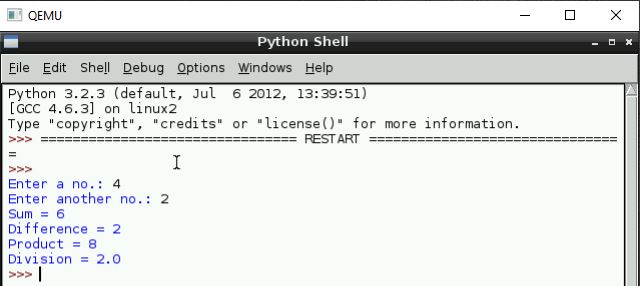
**Output:**



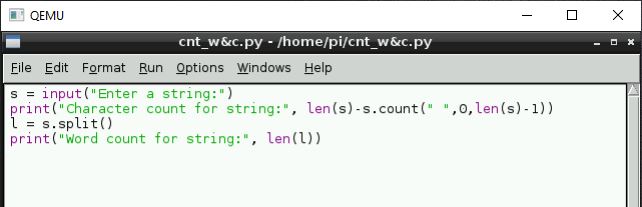
**b)** Read two numbers and print their sum, difference, product and division.



**Output:**



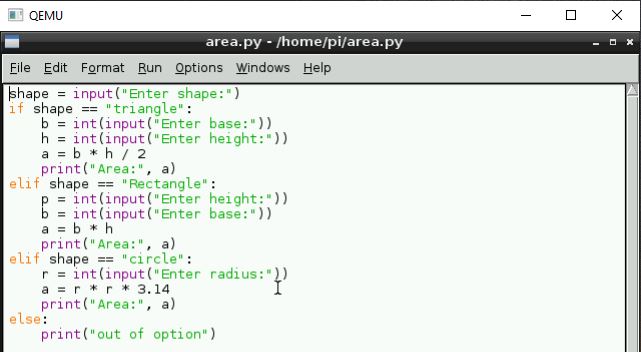
**c)** Word and character count of a given string.



**Output:**



**d)** Area of a given shape (rectangle, triangle and circle) reading shape and appropriate values from standard input.



**Output:**

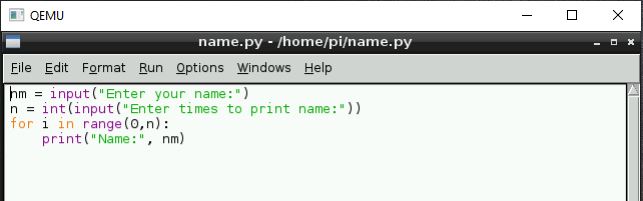


**Result:** Thus, we have learnt to make Python programs in Pi.

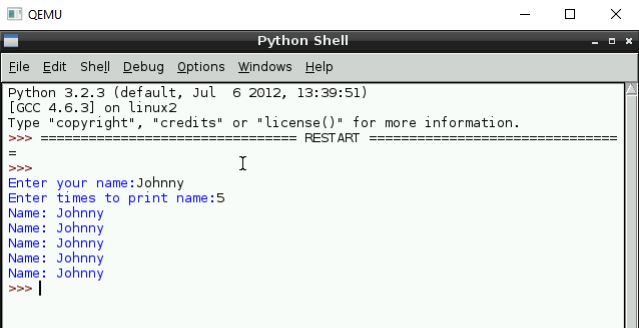
**EXPERIMENT 3**

**Run some Python programs on PI like:**

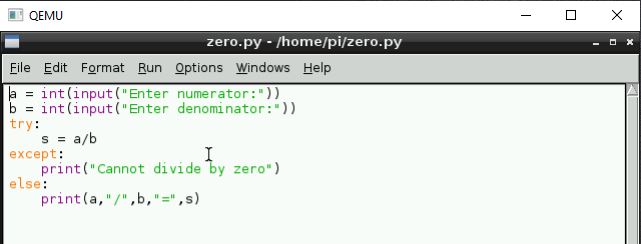
**a)** Print a name ‘n’ times, where name and n are read from standard input, using for and while loops.



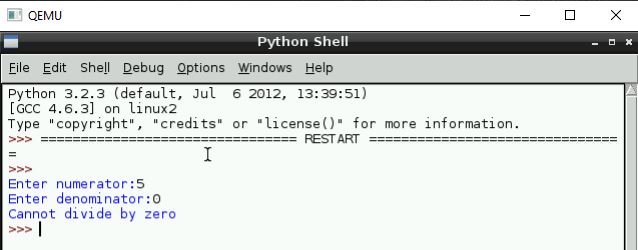
**Output:**



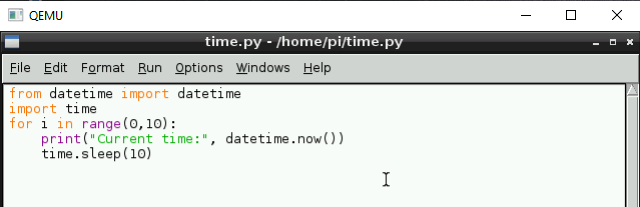
**b)** Handle Divided by Zero Exception.



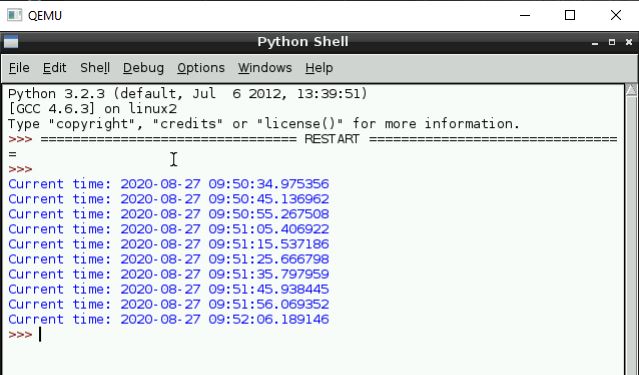
**Output:**



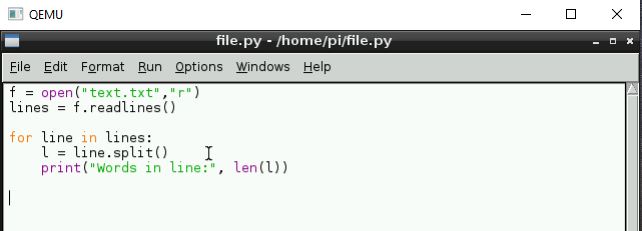
**c)** Print current time for 10 times with an interval of 10 seconds.



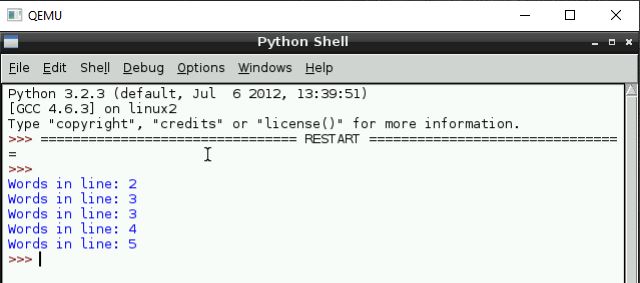
**Output:**



**d)** Read a file line by line and print the word count of each line.



**Output:**



**Result:** Thus, we have learnt to make python programs in Pi.

**EXPERIMENT 4**

**a)** Write a python program to Light an LED.

**Program:**

import RPi.GPIO as GPIO

import time

GPIO.setmode(GPIO.BCM)

GPIO.setwarnings(False)

GPIO.setup(18,GPIO.OUT)

print "LED on"

GPIO.output(18,GPIO.HIGH)

time.sleep(1)

print "LED off"

GPIO.output(18,GPIO.LOW)

Once you have typed all the code and checked it, save and exit the text editor with “Ctrl + x” then “y” then “enter”.

**Output:**

To run this code type: sudo python LED.py

You will see the LED turn on for a second and then turn off.

**b)** Write a python program to Get input from two switches and switch on corresponding LED’s.

**Program:**

#!/usr/bin/env python

import RPi.GPIO as GPIO

import time

LedPin = 11 # pin11 --- led

BtnPin = 12 # pin12 --- button

Led\_status = 1

def setup():

GPIO.setmode(GPIO.BOARD) # Numbers GPIOs by physical location

GPIO.setup(LedPin, GPIO.OUT) # Set LedPin's mode is output

GPIO.setup(BtnPin, GPIO.IN, pull\_up\_down=GPIO.PUD\_UP) # Set BtnPin's mode

is input, and pull up to high level(3.3V)

GPIO.output(LedPin, GPIO.HIGH) # Set LedPin high(+3.3V) to off led

def swLed(ev=None):

global Led\_status

Led\_status = not Led\_status

GPIO.output(LedPin, Led\_status) # switch led status(on-->off; off-->on)

if Led\_status == 1:

print 'led off...'

else:

print '...led on'

def loop():

GPIO.add\_event\_detect(BtnPin, GPIO.FALLING, callback=swLed, bouncetime=200)

# wait for falling and set bouncetime to prevent the callback function from being called multiple

times when the button is pressed

while True:

time.sleep(1) # Don't do anything

def destroy():

GPIO.output(LedPin, GPIO.HIGH) # led off

GPIO.cleanup() # Release resource

if \_\_name\_\_ == '\_\_main\_\_': # Program start from here

setup()

try:

loop()

**Output:**

sudo python 02\_btnAndLed.py

Now, press the button, and the LED will light up; press the button again, and the LED will go out.

At the same time, the state of the LED will be printed on the screen.

**c)** Flash an LED at a given on time and off time cycle, where the two times are taken from a file.

**Program:**

import time

import RPi.GPIO as GPIO ## Import GPIO library

GPIO.setmode(GPIO.BOARD) ## Use board pin numbering

GPIO.setup(11, GPIO.OUT) ## Setup GPIO Pin 11 to OUT

while True:

GPIO.output(11,True) ## Turn on Led

time.sleep(1) ## Wait for one second

GPIO.output(11,False) ## Turn off Led

time.sleep(1) ## Wait for one second

Output:

After running the code, the LED will start blinking.

**EXPERIMENT 5**

**Objective:** Python program on Flash an LED based on cron output (acts as an alarm).

**Step 1:**

First, create a ‘blink’ folder and then create an empty ‘blink.py’ file. It should be inside the path /home/pi/blink. (The /home/pi/ is the default user files path).

**Python code: blink.py**

import RPi.GPIO as GPIO

import time

LedPin = 11 # pin11

def setup():

GPIO.setmode(GPIO.BOARD) # Numbers GPIOs by physical location

GPIO.setup(LedPin, GPIO.OUT) # Set LedPin's mode is output

GPIO.output(LedPin, GPIO.HIGH) # Turn ON led

def blink():

while True:

GPIO.output(LedPin, GPIO.HIGH) # led on

time.sleep(1)

GPIO.output(LedPin, GPIO.LOW) # led off

time.sleep(1)

def destroy():

GPIO.output(LedPin, GPIO.LOW) # led off

GPIO.cleanup() # Release resource

if \_\_name\_\_ == '\_\_main\_\_': # Program start from here

setup()

try:

blink()

except KeyboardInterrupt: # When 'Ctrl+C' is pressed, the child program

destroy() will be executed.

destroy()

**Step 2:**

Open the terminal and navigate to the ‘blink’ folder by using the following command cd blink Create a launcher script by typing the command in the terminal. nano launcher.sh

•Now the editor will get launched. Copy and paste the following code to the editor. This shell script will navigate to the blink code.

#!/bin/sh

# launcher.sh

# navigate to home directory, then to this directory, then execute

python script, then back home

cd /

cd home/pi/blink

sudo python blink.py

cd /

**Step 3:**

We want to make this launcher.sh script executable. Type the following command for it.

chmod 755 launcher.sh

**Output:** This will run the launcher script during the startup.