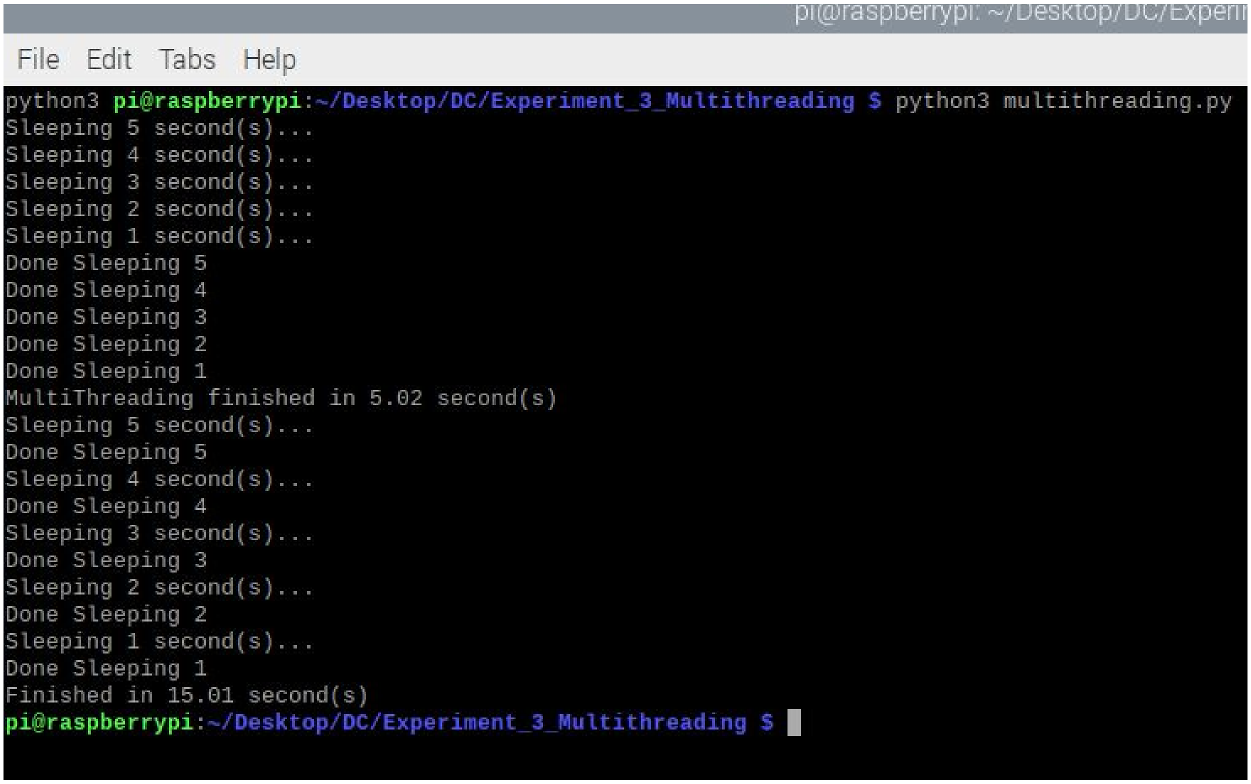
Experiment_3_Multithreading

1. Multithreading

```
import time
import concurrent.futures

def doSomething(seconds):
    print(f"Sleeping {seconds} second(s)...")
    time.sleep(seconds)
    return f"Done Sleeping {seconds}"

secs = [5,4,3,2,1]

###Multithreading - I/O bound
start = time.perf_counter()
with concurrent.futures.ThreadPoolExecutor() as executor:
    results = executor.map(doSomething , secs)
    for result in results:
        print(result)

finish = time.perf_counter()
print(f"MultiThreading finished in {round(finish-start,2)} second(s)")

###Normal
start = time.perf_counter()
r = map(doSomething, secs)
for rs in r:
    print(rs)
finish = time.perf_counter()
print(f"Finished in {round(finish-start,2)} second(s)")
```

Output



```
python3 pi@raspberrypi:~/Desktop/DC/Experiment_3_Multithreading $ python3 multithreading.py
Sleeping 5 second(s)...
Sleeping 4 second(s)...
Sleeping 3 second(s)...
Sleeping 2 second(s)...
Sleeping 1 second(s)...
Done Sleeping 5
Done Sleeping 4
Done Sleeping 3
Done Sleeping 2
Done Sleeping 1
MultiThreading finished in 5.02 second(s)
Sleeping 5 second(s)...
Done Sleeping 5
Sleeping 4 second(s)...
Done Sleeping 4
Sleeping 3 second(s)...
Done Sleeping 3
Sleeping 2 second(s)...
Done Sleeping 2
Sleeping 1 second(s)...
Done Sleeping 1
Finished in 15.01 second(s)
pi@raspberrypi:~/Desktop/DC/Experiment_3_Multithreading $
```

MultiThreading finished in 5.02 second (s) while normally it took 15.01 second (s) to complete