15 feb Assignment

1. What is multiprocessing in python? why is it useful?

Multiprocessing in Python is a built-in package that allows the system to run multiple processes simultaneously. it will enable the breaking of applications into smaller threads that can run independently.

APPOINTMENT NOTES

2. Difference b/w Multiprocessing & Multithreading?

Multiprocessing	Multithreading
CPU are added for increasing computing power.	while in Multithreading, many threads are created of a single process for increasing computing power.

31 JULY TUESDAY

212-153 • WK 31

APPOINTMENT/MEETING

Multi processing

(2) In Multiprocessing, Many process are executed simultaneously.

(3) are classified into Symmetric & Asymmetric

4. process creation is a time-consuming process.

5. every process owns a separate address space.

Multithreading

(2) while in multithreading, many threads of a process are executes simultaneously.

(3) While Multithreading is not classified in any categories.

4. while in Multithreading, process creation is according to economical.

-> a common address space is shared by all the threads.

3. Write a python code to create a process using the multiprocessing module.

```
#Importing the multiprocessing module.
import multiprocessing.
def print_cube(num):
    """ function to print cube of given num """
    print("cube: {}".format(num*num*num))
def print_square(num):
    """ function to print square of given num """
    print("Square: {}".format(num*num))
if __name__ == "__main__":
    p1 = multiprocessing.Process(target=print_square, args=(10,))
    p2 = multiprocessing.Process(target=print_cube, args=(10,))
    p1.start()
    p2.start()
    print("Done!")
```

APPOINTMENT NOTES

Q.4 What is a multiprocessing pool in python is used?

→ Python multiprocessing pool can be used for parallel execution of a function across multiple input values, distributing the input data across process (data parallelism)

Q.5 How can use create a pool of worker processes in python using the multi processing module?

1. Create the Process Pool
2. Submit Tasks to the Process Pool.
3. wait for Tasks to complete.
4. Shutdown the Process Pool.

Q.6 Write a python program to create 4 processes, each process should print a different no using the multiprocessing module in Python.

APPOINTMENT NOTES

```
from multiprocessing import Pool  def f(n)
return x*x  if __name__ == '__main__':
    with Pool(5) as p:
        print(p.map(f, [1,2,3]))
```