MLDS 422 - Intro to Python
Lab 8

Sungsoo Lim

November 15, 2023

NORTHWESTERN
UNIVERSITY

Today's Lab Materials

- Multithreading
- Multiprocessing
- Project 2

Terminologies

- **Process**: An execution environment of a computer program (e.g. a Python script). Multiple processes can be running the same program, but they can use different data and compute resources.

- **Thread**: A unit of execution in a process. Threads only execute serially, but a process can have multiple threads running, taking on different parts of the task.

Threading

Difference Table Between Process and Thread

Process	Thread
A process is an instance of a program that is being executed or processed.	Thread is a segment of a process or a lightweight process that is managed by the scheduler independently.
Processes are independent of each other and hence don't share a memory or other resources.	Threads are interdependent and share memory.
Each process is treated as a new process by the operating system.	The operating system takes all the user-level threads as a single process.
If one process gets blocked by the operating system, then the other process can continue the execution.	If any user-level thread gets blocked, all of its peer threads also get blocked because OS takes all of them as a single process.

Figure: https://www.javatpoint.com/process-vs-thread

Multithreading - Advantages and Use Cases

- Response time, speed
- Input/Output in Python - downloading data from the internet and writing data to files
- Debugging can be more difficult

Multiprocessing - Advantages and Use Cases

- If a process can be broken down into subprocesses that can run independent of one another
- CPU bound tasks - multiple cares can be used simultaneously
- More efficient debugging