A **thread** is a unit of CPU utilization. When we open a thread, we are assigning a subsection of the main process' resources to handle different tasks concurrently.

Motivation and Introduction

A thread allows us to carry different types of calculations at the same time. The threads start as a subset of the main process resources. A good example is a server, where many different requests can be coming in. If we were to open a new process for every request on a server, it would not be the most efficient since creating an entirely new process can be expensive.

There are some benefits to creating a thread, such as:

- **Responsiveness**: Threads allow a program to carry out multiple calcuations at once, especially on multicore systems. For a GUI program, if we press a button that causes some expensive calculation, then the program might spawn a new thread to handle it in the background. If we had a single-threaded application, then we would have to wait for the application to finish to use the program again.
- **Resource Sharing**: Previously we had said that there were two ways to share information between processes: shared memory and direct messaging. Since the threads share the same memory space as the parent process, then we don't have to do this.
- **Economy**:Again, it is easier to create threads because they already share the space of the parent's and don't need another memory space, loading, and separate executing.
- **Scalability**: The benefits of multithreading might be even greater in a multi-threaded system.

When the olden days only had only one core, then multithreading only meant interleaving execution of different threads. However, with mutlicores we can have one thread assigned to one core.
[1]

[1] **Amdahl's Law** measures the amount of performance gain we could obtain from doing things serially and concurrently. Let S be the percentage performed serially. The possible speedup we could receive from concurrency is

$$\text{speedup} \leq \frac{1}{S - \frac{1-S}{N}}$$

where N is the number of processing cores.