What’s are the Differences between Processes and Threads

Summary: in this tutorial, you’ll learn about the processes and threads, and more importantly, the main differences between them.

Introduction to processes and threads

Suppose that you have a simple Python program:

```
x = 10
y = 20
z = x + y
```

Computers don’t understand Python. They only understand machine code, which is a set of instructions containing zero and one.

Therefore, you need a Python interpreter to execute this Python program that translates the Python code to machine code.

When you execute the `python app.py` command, Python interpreter (CPython) compiles the `app.py` into machine code.

The operating system (OS) needs to load the program into the memory (RAM) to run the program.

Once the OS loads the program to memory, it moves the instructions to the CPU for execution via bus.

In general, the OS moves the instructions to a queue, also known as a pipeline. Then, the CPU will execute the instructions from the pipeline.

By definition, a process is an instance of a program running on a computer. And a thread is a unit of execution within a process.

Notice that if you launch a program multiple times, you'll have one program but multiple processes which are instances of the program.

The following picture illustrates the flow of running a Python program on a computer:

So far, you've learned how to develop a program that has one process with one thread. Therefore, the terms process and thread are often used interchangeably sometimes.

A program may have one or more processes and a process can have one or more threads.

When a program has multiple processes, it's called **multiprocessing**. If a program has multiple threads, it's called **multithreading**.

Single-core processors

In the past, a CPU has only one core. In other words, it can run only a single process at one time. To execute multiple processes “at the same time”, the OS uses a software component called **scheduler**:

The scheduler is like a switch that handles process scheduling. The main task of the scheduler is to select the instructions and submit them for execution regularly.

The scheduler switches between processes so fast (about 1ms) that you feel the computer can run multiple processes simultaneously.

Multicore processors

Today, the CPU often has multiple cores, e.g., two cores (dual-core) and four cores (quad-core).

The number of cores will determine the number of processes that the CPU can execute simultaneously. Generally, the more cores the CPU has, the more processes it can truly execute simultaneously.

For example, a dual-core CPU can execute exactly two processes simultaneously and a quad-core CPU can execute at most four processes simultaneously.

Multiprocessing uses a multi-core CPU within a single computer, which indeed executes multiple processes in parallel.

CPU-bound vs. I/O bound tasks

In general, programs handle two types of tasks: I/O-bound or CPU-bound.

- An I/O-bound process spends more time doing I/O than doing computations. The typical examples of I/O bound processes are network requests, database connections, and file I/O.
- In contrast, a CPU-bound process uses more time doing computation than generating I/O requests. The typical examples of CPU-bound processes are matrix multiplication, finding prime

numbers, video compression, and video streaming.

Technically, multithreading is suitable for I/O bound processes, and multiprocessing is suitable for CPU-bound processes.

The main differences between a process and a thread

The following table illustrates the main differences between a process and a thread:

Criteria	**Process**	**Thread**
Memory Sharing	Memory is not shared between processes	Memory is shared between threads within a process
Memory footprint	Large	Small
CPU-bound & I/O-bound processing	Optimized for CPU-bound tasks	Optimized for I/O bound tasks
Starting time	Slower than a thread	Faster than a process
Interruptablity	Child processes are interruptable	Threads are not interruptible

Summary

- A process is an instance of a program running on a computer.
- A program can have one or more processes and a process can have one or more threads.
- A thread is a unit of execution within a process.
- A process can have one or more threads.