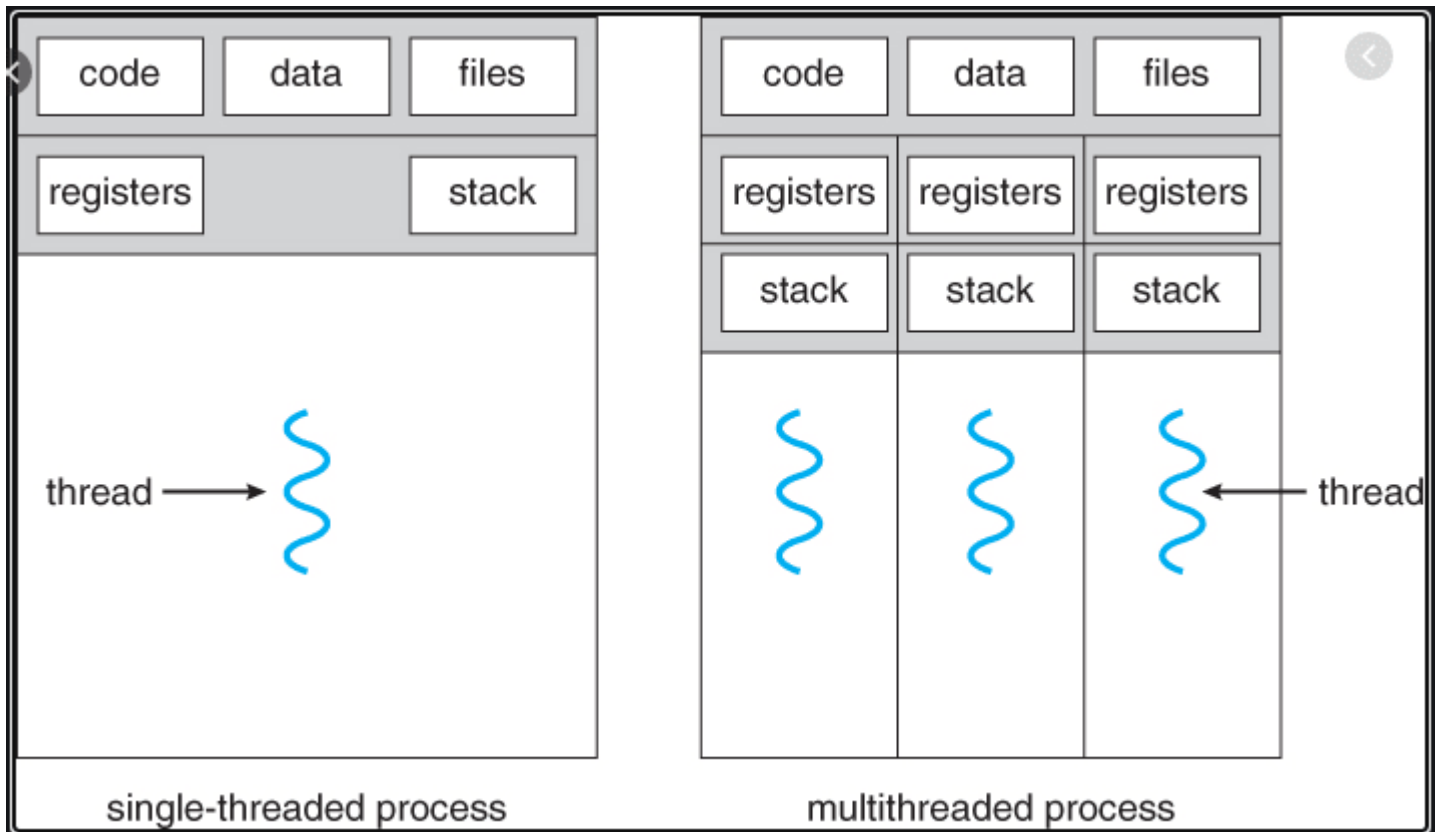


Multi-threading in-depth

A **multi-threaded** program contains two or more parts that can run concurrently and each part can handle a different task at the same time making optimal use of the available resources specially when your computer has multiple CPUs.



Multi-threading extends the idea of multitasking into applications where you can subdivide specific operations within a single application into individual threads. It enables you to write in a way where multiple activities can proceed concurrently in the same program.

There are a handful of programming languages that give room for **multi-threading**, and most of the languages are Object Oriented

Programming languages (OOP). Languages like **Java**, **C**, **C++** and even **.NET** frameworks. Some other interpreted languages also made the cut, like **Ruby MRI** for **Ruby** and **CPython** for **Python**. If you were waiting to see **Javascript**, well you won't because JavaScript **does not support** multi-threading and that's because the JavaScript interpreter in the browser is a single thread.

Heavily Multi-threaded Applications

Almost all well-built applications support multi-threading. Let's look at browsers. Most browsers are multi-threaded from **firefox** to **Safari** to **Chrome** and many others. But today we'd talk more about **chrome**.

Google Chrome

Chrome has a multi-process architecture and each process is heavily multi-threaded. The main goal is to keep the main thread ("UI" thread in the browser process) and IO thread (each process' thread for handling IPC) responsive. This means offloading any blocking I/O or other expensive operations to other threads.