

Process vs Thread.

Single-threaded process is represented by its *address space* (code, data, files...) and *execution context* (values of registers, stack pointer, program counter etc...). OS encapsulate all these information in *process control block (PCB)*.

Threads represent multiple independent execution contexts, but same address space. So they will share code, data, files... But they will execute different instructions, access different portions of that address space, operate on different portions of the input etc.

PCB for multi-threaded process will contain one address space mappings and multiple instances of execution context – one per thread.

Benefits of multithreading

1. Several threads, that belongs to once process, can execute same code (but different instructions) *simultaneously* on different CPU cores. For example, they can process different parts of input at one time. So, we achieve *speed up*.