# Concurrency: An Introduction

Talk about how tasks in Rust differ from threads in C. A thread in C “is very much like a separate process, except for one difference: they *share* the same address space and thus can access the same data.”

“The sate of a single thread is thus very similar to that of a process. It has a program counter (PC) that tracks where the program is fetching instructions from. Each thread has its own private set of registers it uses for computation; thus, if there are two threads that are running on a single processor, when switching from running one (T1) to running the other (T2), a context switch must take place. The context switch between threads is quite similar to the context switch between processes, as the register state of T1 must be saved and the register state of T2 restored before running T2.”

State is stored in one or more thread control blocks (TCBs)

Address space remains the same in threads, but not in processes

Each thread has its own stack