seL4 Thread Emulation Implementation

In the real seL4 system, the program execution are modeled as seL4 thread. For the emulation, we also have two implementation options, map each seL4 thread to a Linux user space **thread** or a Linux user space **process**.

Trade-offs of mapping to threads

- creation / deletion are relatively light weighted. ✓
- we can share the address spaces. ✓
- porting pthread from standard libc is complicated. ✗
- starting an entire program requires loading the ELF file into the memory first then jumping to the entry point. ✗
- dealing with concurrency issues. ✗
- handling signals can be complicated. ✗
- pthreads already use TLS, which conflicts the seL4 system. ✗

Trade-offs of mapping to process

- much fewer concurrency issues. ✓
- handling signals is easy. ✓
- execution of the entrie program is easy. ✓
- no need to port pthread from libc. ✓
- process creation / deletion is relatively heavy. ✗
- different address spaces, requires explicit process communication mechanisms for sharing data. ✗