An extra discussion on the scheduling emulation

About emulating scheduling, I've just come up with the idea recently, still implementing it.

- To emulate thread suspension, we can use **blocking sockets**, so each time kernel uses the scheduling algorithm to determine the next running seL4 thread and **only** replies to it. Hence all other seL4 threads will block on the socket except one that has been chosen.
- To emulate the preemption, we can send signal to our seL4 application, then the signal handler routine is invoked and will be blocked on the socket until the next time when the kernel emulator sends a message and tells it to resume.