Memory Mapping Emulation Implementation

What do we want to achieve

- The program execution will stop if the memory access is invalid and without any mapping.
- The program will continue execution after an explicit seL4 mapping invocation.
- Assume we implement a lazy mapping mechanism. The program will stop upon accessing the memory address and the vm fault handler will update the paging structure. Eventually, the faulting program restarts at the faulting instruction and continues.

How to implement?

- set up a signal handler during the runtime initial stage.
- the seL4 application tries to access an unmapped memory address.
- **`SIGSEGV`** triggered and invokes the siganl handler routine.
- the signal handler sends an IPC to the kernel emulator.
- the kernel emulator will verify the mapping and reply.
- the signal handler will check the reply. It will either map the faulting address by using **`MAP_FIXED`** then return back to the previous context or halt the program execution.