seL4 Kernel Recap

Recap on seL4 kernel functionalities

For emulating a simple hello world roottask, the minimal requirements of the kernel objects are **cnode**, **vspace**, **endpoints**, and **tcb**.

- the **capability** is the most critical part of the seL4 system. As far as I understand it's **`access rights + reference`**. With the capability, we can retrieve the virtual address of this kernel object then access the object.
- the **vspace** implements the architecture dependent paging structures so that we can map virtual addresses to the physical addresses.
- the **endpoint** facilitates the message-passing communication between threads and implements the rendezvous IPC model. The endpoint structure internally is implemented as a queue which only has a head and a tail. They both point to a **tcb** structure.
- the **tcb** represents the seL4 thread. It contains all information about the seL4 thread, such as the thread state, capability pointer to the IPB buffer, fault, scheduling priority, etc.

  However, we can see the most important part is how to resolve the pointer referencing so that the kernel emulator can access any seL4 thread's kernel objects. We explore how the original kernel does that. The answer is the seL4 kernel map the entire physical memory into the kernel window.(discuss in next slide)