# practice04

> 2021011158 김선희

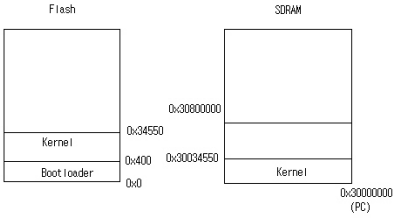
## ➕ Analyzing VPOS

- vpos: One of the Real-Time OS (RTOS). A combination of a bootloader and a kernel.
  - RTOS: An essential component for embedded systems where immediate processing is critical.

> reference: https://www.likewind.cloud/doku.php?id=computer:rtcclab:vpos_%EB%B6%84%EC%84%9D_-_1.%EC%B4%88%EA%B8%B0%ED%99%94

### bootloader

: when the bootloader is launched




1. ***hal/bootloader/vpos_bootloader.S***

- 1: Initialize the hardware, copy the VPOS kernel into memory, and set the PC to the first address in memory.
- 2~8: Interrupt Vector Table. Serves as an interrupt handler for interrupts that occur during program execution.

2. ***hal/cpu/vpos_kernel-ld-script***

- Move your PC to the kernel startup point.

### kernel

***hal/cpu/HAL_arch_startup.S***

**Configuration**

- Startup code

  : Assembly code that initializes the embedded target board before performing the main() function.

- HAL related code

  for Hardware Abstraction Layer.

  : Associated with tasks that are primarily done by hardware, such as interrupt handling, context switching.

## Kernel Porting | Implement Startup Code

### Implementation

***hal/cpu/HAL_arch_startup.S***

```
vh_VPOS_reset:
    ...
    // Mask interrupt and fast interrupt
    mrs     r0, cpsr
    orr     r0, r0, #vh_NOINT
    msr     cpsr, r0
```

```
// Invalidate all instruction caches to point of unification. Also flushes branch target cache.
mov     r0, #0x00
mcr     p15, 0, r0, c7, c5, 0

// Control Register Setting
mrc     p15, 0, r0, c1, c0, 0
bic     r0, r0, #0x01
bic     r0, r0, #0x04
bic     r0, r0, #0x1000
bic     r0, r0, #0x2000
orr     r0, r0, #0x02
orr     r0, r0, #0x800
mcr     p15, 0, r0, c1, c0, 0

// change vector table base address
ldr       r0, =vh_vector_start
mcr     p15, 0, r0, c12, c0, 0
...
```