

ChCoreLab4

1 思考题1

在start.s文件中，首先将系统寄存器mpidr_e11中的CPU ID存储到寄存器x8中。然后通过将该值与0xFF进行按位与运算，判断是否为0号CPU（即主核）。如果是0号CPU，程序将跳转到主核的初始化代码primary执行。而对于其他CPU，它们将继续执行，并被阻塞在wait_until_smp_enabled代码处，等待主核完成初始化并设置secondary-boot_flag标志后才继续执行secondary_init_c，进行其他CPU的初始化操作。

2 思考题2

- 是虚拟地址, 因为 MMU 已经启动;
- 通过 `main` 函数的参数 `boot_flag` 传入 `enable_smp_cores`
- 由kernel/arch/aarch64/main.c文件中的main.c函数中可知, `boot_flag` 是smp的boot flag地址, 是物理地址, 在 `enable_smp_cores` 中通过调用 `phys_to_virt` 将其转换为虚拟地址, 即为 `secondary_boot_flag`。

```
void enable_smp_cores(paddr_t boot_flag)
{
    int i = 0;
    long *secondary_boot_flag;

    /* Set current cpu status */
    cpu_status[smp_get_cpu_id()] = cpu_run;
    secondary_boot_flag = (long *)phys_to_virt(boot_flag);
    for (i = 0; i < PLAT_CPU_NUM; i++) {
        secondary_boot_flag[i] = 1;
        flush_dcache_area((u64) secondary_boot_flag,
                           (u64) sizeof(u64) * PLAT_CPU_NUM);
        asm volatile ("dsb sy");
        while (cpu_status[i] == cpu_hang)
            ;
        kinfo("CPU %d is active\n", i);
    }
    /* wait all cpu to boot */
    kinfo("All %d CPUs are active\n", PLAT_CPU_NUM);
    init_ipi_data();
}
```