Report of ChCore-Lab4

杨乐天

思考题1

参考函数`_start`的代码：

- `mrs x8, mpidr_el1`函数将CPU的ID读入寄存器x8。
- `and x8, x8, #0xFF`读取CPU ID的后8个bit。
- `cbz x8, primary`在CPU ID的后8个bit都为0的情况下跳转到`primary`，也就是主CPU的执行逻辑。其他CPU按顺序进入`wait_for_bss_clear`被阻塞。

阻塞其他CPU的方式有两层：

- 第一层通过读取`clear_bss_flag`，在结果小于等于0时无限循环，实现了对`clear_bss_flag`的监测工作。

```
wait_for_bss_clear:
    adr x0, clear_bss_flag
    ldr x1, [x0]
    cmp     x1, #0
    bne wait_for_bss_clear
```

- 第二层以类似的方式无限循环，监测`secondary_boot_flag`。

```
wait_until_smp_enabled:
    /* CPU ID should be stored in x8 from the first line */
    mov x1, #8
    mul x2, x8, x1
    ldr x1, =secondary_boot_flag
    add x1, x1, x2
    ldr x3, [x1]
    cbz x3, wait_until_smp_enabled
```

主CPU在完成多核设置之后将`clear_bss_flag`和`secondary_boot_flag`设置为1，于是其他CPU不再阻塞，与主CPU进行类似的操作，进入特权级EL1，设置栈指针，并跳转到C语言，开始执行内核初始化。

思考题2

在`start.S`中调用的`secondary_boot_flag`是物理地址，因为此时MMU尚未启动。

传入过程：在`init_c`中调用`start_kernel`函数时传入`secondary_boot_flag`，虽然此时MMU已经启动，但是其仍然是物理地址。`start_kernel`函数执行一些代码后进入`main`函数中，`main`函数再将物理地址`secondary_boot_flag`传入`enable_smp_cores`中。

赋值过程：在`smp.c`文件的`enable_smp_cores`函数中，首先调用了`phys_to_virt`函数将物理地址转化为虚拟地址，从而可以进行访问。在`for`循环中，将每个CPU对应的`secondary_boot_flag`都设置为1，等待该CPU启动之后打印该CPU已启动。

练习1

根据提示的内容，我们遍历每个CPU核心的就绪队列，通过`init_list_head`函数初始化每个CPU核心的`queue_meta`结构体，同时初始化对应的`lock`。

练习2

根据提示写两行代码。使用`list_append`函数即可实现`thread->ready_queue_node`的插入。注意到`list_append`函数没有对`queue_len`属性进行修改，因此剩下一行就是对`queue_len`进行加一操作。

练习3

`find_runnable_thread`函数：根据提示使用`for_each_in_list`宏，并且`if`语句的条件也已经给出，我们只需要在遍历中满足`if`条件后`break`即可。

`__rr_sched_dequeue`函数：与练习2类似，使用`list_del`函数再对`queue_len`减一即可。

练习4

系统调用`sys_yield`中只需要调用`sched()`函数即可。

`rr_sched`函数中只需要根据提示写出如下代码即可：

```
rr_sched_enqueue(old);
```

练习5

我们可以根据上下文得知使用`asm`调用汇编语言读取、写入的格式分别为：

```
asm volatile ("..."::"=r" (var));
asm volatile ("..."::"r" (var));
```

然后根据步骤的要求读取、写入`cntfrq_el0`寄存器和`cntp_tval_el0`寄存器即可。

练习6

在`switch`代码块中写入`case INT_SRC_TIMER1`即可判断中断号，剩余内容根据提示写出即可。

`kernel/irq/timer.c`中的减少budget后还需要调用一次`sched()`函数。

在`policy_rr.c`中修改budget即修改为`DEFAULT_BUDGET`。

```
old->thread_ctx->sc->budget = DEFAULT_BUDGET;
```

练习7

根据注释提示，我们只需要设置属性值即可，而这些属性值均在函数头定义中给出，因此只要按顺序填入即可。