JOS Lab4 文档

杨健邦 515030910223

本文档描述了本次 LAB 各个 Exercise 实现的方法或者我自己的理解 (只陈序一部分相对有难度比较复杂的设计)。至于对 Question 的解答以及以及 Challenge 的实现方法，请阅读 answer-lab4.txt。

Part A: Multiprocessor Support and Cooperative Multitasking

Exercise 1-3.

- 现在 page_init 为物理地址 MPENTRY_PADDR 预留一个物理页，然后将 AP 的启动代码映射到此处。在 trap_init_percpu 使用 thiscpu->ts_cpu0 设置好内核栈以及段。由于之前的 lab 使用的是 sysenter 来进行系统调用，在 trap_init_percpu 的时候同时也要给每个 CPU 注册好它相应的 MSR，为每个 CPU 开启 systenter 系统调用的方式。

- 除了 BSP，其它 AP 都没有开启大页，因此修改 pmac.c 中的 mem_init，全部采用小页映射的方式。

Exercise 4-4.1.

- 在相应的地方拿锁放锁。要注意的是，在 systenter 的时候也要拿锁，sysexit 之前要放锁。在实现 ticket spin lock 的时候，为了防止乱序或者相应的变量被存放到寄存器中读（但是在内存里的值被其它 CPU 所更改），有两种解决方法，一种是在定义 spinlock 结构体中，own 变量要加上 volatile 修饰词，强制每次读的时候都要从内存中读取；另一种方法是读操作也使用原子指令 atomic_return_and_add 进行读取。

- 由于用 sysenter 的方法进行系统调用，不会自动保存更新 env 的 trapframe，调用 sys_yield 之后，当前 env 的环境没有正常保存，因此对于 sys_yield 的情况需要更新当前 env 的 trapframe 中的 eip 和 esp。

Exercise 5.

- Round-robin 调度方法。分为两种情况，第一种是 CPU 之前没有运行的 CPU（thisenv）为空，那么从 0 到 NENV 遍历，找到第一个可运行的 env 进行调度；第二种是 CPU 之前有运行过的 env，那么从当前 env 的后一个开始环形遍历，找到第一个可运行的 env 进行调度。

Exercise 6.

- Lab3 的 sysenter 的方法只能传递 4 个参数，而 sys_page_map 需要使用 5 个参数，因此需要对 lib/syscall 进行修改，用%esi 来传递第五个参数。在 systenter 之前，先将返回%eip 入栈，因为用户的%esp 保存在%ebp 中，因此在 systenter 之后，可以通过%ebp 来找到用户的%eip。

Part B: Copy-on-Write Fork

Exercise 7-10.

- 为了使得处理完 page fault 之后，用户 env 的 trapframe 能够恢复正常，需要将 trameframe 的 eip 压进用户 env 运行的栈，然后将 esp 设置成 trapframe 中的 esp – 4 的值，此时使用 ret 指令便可以同时使得 eip 和 esp 恢复正常。

- 要理解 vpt 和 vdt 为什么分别对应着所有的 page table 和 directory table。虽然在映射的时候，只做了一个将 UVPT 映射到 pgdir 的物理地址的映射。但是，根据地址翻译的方法，相当于当前环境的所有页表就在 UVPT-UVPT+PTSIZE 的这一段虚拟地址上（一共 1024 个页表，正好等于 PTSIZE），页表按顺序排列，从 UVPT 排到 UVPT+PTSIZE。Page directory table 也被看成是一个 page table，在(UVPT+(UVPT>>12)*4)的虚拟地址处，它相当于第 UVPT>>22 个页表。相关链接：https://pdos.csail.mit.edu/6.828/2014/lec/l-josmem.html

Exercise 11.

- 要将 parent 和 child 的 UTOP 以下的所有页，除了 user exception stack，在 parent 和 child 的页表中映射成 COW 的方式。这样他们对页数据的读取和修改就不会互相影响。
- 由于 user exception stack 就是用来处理这些 COW 页的，因此必须给 parent 和 child 分配不用的页，不能是 COW 的形式。

Part C: Preemptive Multitasking and Inter-Process communication (IPC)

Exercise 12-14.

- JOS 采用了一个 CPU 一次只会处理一个中断的方式，这就意味着，在进入内核态处理中断的时候要关中断。在设置 trapgate 的时候有一个参数是 istrap，当 istrap 参数是 0 的时候，CPU 处理该中断时会自动关闭中断，在 JOS 中，应该要将所有的 interrupt/trap gate 设置成 interrupt gate，也就是 istrap 应总是设置为 0。

- 我们的系统调用采用的是 systenter 的方法，执行 systenter 指令会自动关中断，但是 sysexit 并不会打开中断，因此，在 sysexit 之前还需要使用 sti 指令来打开中断。