Project2 Non-Preemptive Kernel 设计文档

中国科学院大学
张旭
2017/10/16

1. Context Switching 设计流程

1. PCB 包含的信息
   PCB 包含$16～$23，$30，sp，ra 寄存器的值，进程或线程的 ID、状态信息。

2. 如何启动第一个 task
   当把内核加载到内存后，首先从_stat 函数开始执行，进行 PCB、ready_array、block_array 的初始化工作，然后调用 scheduler_entry 函数，在该函数中跳转到 scheduler 函数，取出 ready_array 中第一个进程 PCB，其地址赋给 current_running，再跳转回 scheduler_entry 函数，将 current_running 中的寄存器值赋给各寄存器，然后通过"jr ra" 跳转到第一个进程的入口。

3. scheduler 的调用和执行流程
   通过 scheduler_entry 函数进入 scheduler 函数。在 scheduler 函数中取出 ready_array 中第一个进程 PCB，把地址赋给 current_running。

4. context switching 时如何保存 PCB，使得进程再切换回来后能正常运行
   除 sp 以及 ra 寄存器，其余寄存器的值保存在 PCB 中相应位置。由于调用 do_yeild 函数时，sp 减 24，且原 ra 的值保存在 16(sp)处。故将 sp+24 以及 16(sp) 的值作为 sp 和 ra 的值保存在 PCB 的相应位置处。

5. 任何在设计、开发和调试 bootblock 时遇到的问题和解决方法
   保存 sp 和 ra 寄存器的值一值不对，后来反汇编 kernal 函数，看了 save_pcb 函数源码后，才发现 sp 的值在调用 save_pcb 时已经发生了变化。

2. Context Switching 开销测量设计流程

1. 如何测量线程切换到线程时的开销
   在上游线程调用 do_yeild 函数时，用全局变量记录 CPU 的 cycle 数。在下游线程开始执行前再用全局变量记录 CPU 的 cycle 数，前后两者的差值除以 MHZ，即可得到线程切换到线程时的开销时间。

2. 如何测量线程切换到进程时的开销
   排列执行顺序：线程 → 进程 → 线程。在上游线程调用 do_yeild 函数时，用全局变量记录 CPU 的 cycle 数。进程中只做 yeild 操作。在下游线程开始执行前再用全局变量记录 CPU 的 cycle 数，前后两者的差值除以 2 倍的 MHZ，即可得到线程切换到进程时的开销时间。

3. 任何在设计、开发和调试 bootblock 时遇到的问题和解决方法
   无

3. Mutual lock 设计流程

1. spin-lock 和 mutual lock 的区别
   spin-lock： 当一个进程不能得到锁时，他会一直循环，每次循环都去尝试得到锁。
   mutual lock： 当一个进程不能得到锁时，他会进入阻塞态，直到锁被释放。

2. 能获取到锁和获取不到锁时各自的处理流程
   进程获取到锁： 将锁的状态置为 locked。
   进程获取不到锁： 调用 block 函数，将进程状态设为 blocked，保存 PCB 状态，并存到 block_array 中，再从 ready_array 中取出新的 PCB 执行。

3. 被阻塞的 task 何时再次执行
   占用锁的进程释放锁后，调用 unblock 函数，从 block_array 取出被阻塞的进程，修改进程状态为 ready，并将该 PCB 放到 ready_array 的队尾。

4. 任何在设计、开发和调试 bootblock 时遇到的问题和解决方法
   无

4. 关键函数功能

此次试验难度不大，主要是实验一的 save_pcb 函数耗费了我很长时间，现在想来也没有特别重要的，但不写又不太好，故把 save_pcb 函数贴在下面。

```
save_pcb:
    # save the pcb of the currently running process
    # need student add
    lw t1,(current_running)
    lw t2,16(sp)
    sw t2, 40(t1)
    addiu sp,sp,24
    sw sp, 32(t1)
4:
    sw $16, 0(t1)
    sw $17, 4(t1)
    sw $18, 8(t1)
    sw $19, 12(t1)
    sw $20, 16(t1)
    sw $21, 20(t1)
    sw $22, 24(t1)
    sw $23, 28(t1)
    sw $30, 36(t1)
    jr ra
```