Project3 Preemptive Kernel 设计文档

中国科学院大学
张旭
2017/11/15

1. 时钟中断与 blocking sleep 设计流程

（1） 中断处理的一般流程

采集时钟中断信号 → CPU 跳到 BFC00380 处执行 general_exception_handler → 关中断 → 查询中断服务程序表并跳到中断服务程序执行 → 保存上下文 →清中断 → 处理中断 → 恢复上下文 → 开中断 → 返回

（2） 你所实现的时钟中断的处理流程，如何处理 blocking sleep 的 tasks；如何处理用户态 task 和内核态 task

本次实现的时钟中断服务函数共考虑了三种情况：

1. 所有的进程都睡眠。此时只将 time_elapse 加一，然后返回，不进行调度。
2. 用户态 task：handle_int → SAVE_CONTEXT (USER) → 关中断（CLI）→查询 int IP → 为时钟中断 → 跳到时钟中断处理函数执行 → ENTER_CRITICAL → 清中断 →退出核心态 → LEAVE_CRITICAL → RESTORE_CONTEXT（USER） → return_from_exception
3. 核心态 task：handle_int → 关中断（CLI）→查询 int IP → 为时钟中断 → 跳到时钟中断处理函数执行 → ENTER_CRITICAL → 清中断 → LEAVE_CRITICAL → return_from_exception

（3） blocking sleep 的含义，task 调用 blocking sleep 时做什么处理？什么时候唤醒 sleep 的 task？

task 调用 blocking sleep 时，修改 task 的 PCB 块中的 deadline 变量，把 task 的状态改为 SLEEPING，然后放到 sleeping_queue 中去。

当 time_elapse 等于 deadline 时，唤醒 task。

（4） 设计或实现过程中遇到的问题和得到的经验（如果有的话可以写下来，不是必需项）

绝不能唯任务书论，还是看官方文件吧。

2. 基于优先级的调度器设计

（1） priority-based scheduler 的设计思路，包括在你实现的调度策略中优先级是怎么定义的，如何给 task 赋予优先级，调度与测试用例如何体现优先级的差别

我在 PCB 块中加入优先级（priority）变量和轮（round）变量，按优先级及轮的大小给 ready_queue 里的进程排序，优先级高，进行的轮数少的进程排在前列。每次执行时，将进程的优先级减 1，当优先级减为 0 时，将 round 加 1，优先级恢复初始值。每次将进程加入 ready_queue 中时，均要

对队列进行一次排序。

Task 的优先级设为 PID+1，在 print_status 函数中打印出进程的优先级，可以看到优先级的动态变化，且进程的 entry_count 之比等于优先级之比。

3. 关键函数功能

```
void sortqueue(node_t *queue){
    if(is_empty(queue));
    else{
        pcb_t *head;
        pcb_t   *test;
        node_t *nhead;
        node_t  *ntest;
        nhead = queue->prev;
        ntest = nhead->prev;
        head = (pcb_t*)nhead;
        while(ntest != queue){
            test = (pcb_t*)ntest;
            if((head->priority > test->priority && head->round == test->round) || head->round < test->round){
                nhead->next->prev = ntest;
                ntest->next = nhead->next;
                nhead->next = ntest;
                nhead->prev = ntest->prev;
                ntest->prev = nhead;
                nhead->prev->next = nhead;
                ntest = nhead->prev;
            }
            else    break;
        }
    }
}
```

该函数根据插入队列元素的优先级和轮大小，将该元素插入队列的合适位置。

参考文献

[1] SEE MIPS RUN

■