Project3 Preemptive Kernel设计文档

中国科学院大学

张旭

2017/11/15

# 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

1. 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
4. 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。

1. 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。

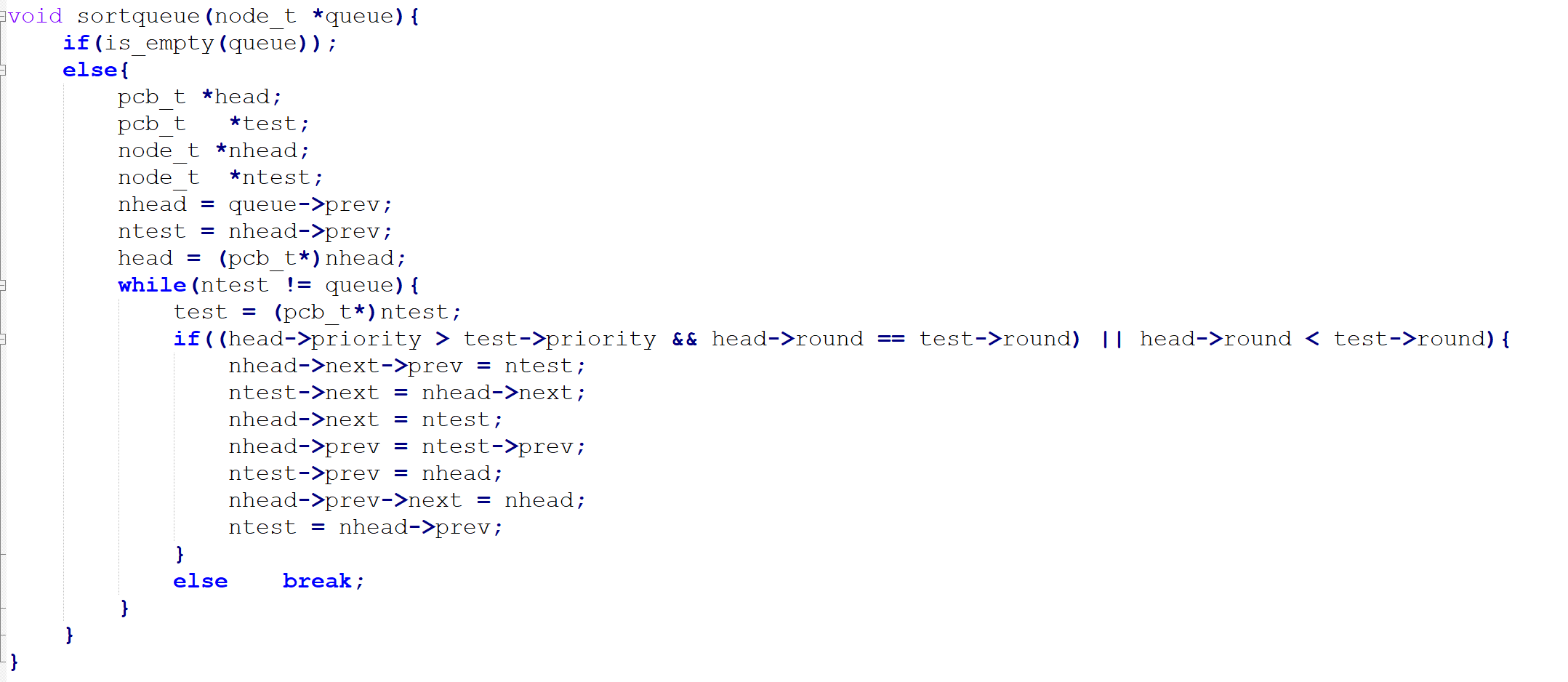
# 基于优先级的调度器设计

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。

# 关键函数功能



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

▄