LAB6 田新韬 2012011268

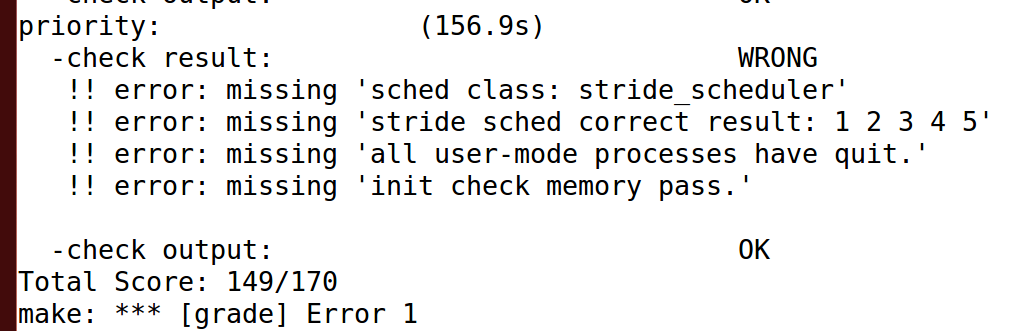
Ex0

先diff修改后的lab5与原lab5，再patch到lab6中

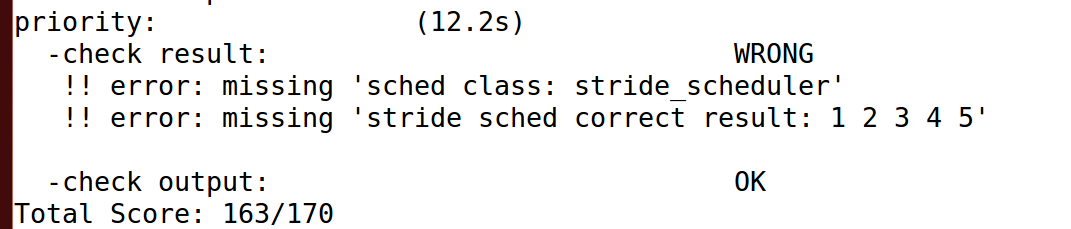
Ex1 使用Round-Robin调度算法

Patch过后还需要修改proc.c中初始化proc\_struct类的代码，以及trap.c处理时钟中断的代码，不然跑不通…

Make grade，可以通过大部分测例，spin/waitkill/priority无法通过，149/170



然后发现还需要修改trap.c处理时钟中断的代码…修改过后可以通过除priority外其他测例：



* 请理解并分析sched\_class中各个函数指针的用法，并结合Round Robin 调度算法描ucore的调度执行过程

Init函数用于初始化，enqueue进程入队/计数器加一，dequeue进程出队/计数器减一，pick\_next选择要执行的下一个进程，proc\_tick响应时钟中断，进行调度。

Ucore的调度执行过程：当需要进程入队，对其使用的时间片初始化并插入至队尾；需要进程出队，直接删除；取出执行的进程时，取就绪队列的头；出现时钟中断，将当前进程的剩余可执行时间减一，至0时标记为need\_resched，在之后调用schedule函数调度。

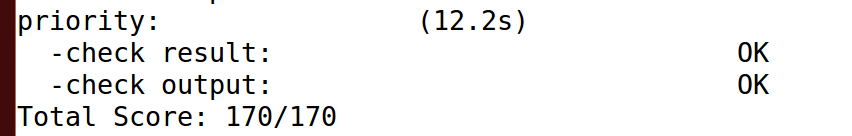
* 请在实验报告中简要说明如何设计实现”多级反馈队列调度算法“，给出概要设计，鼓励给出详细设计

多级反馈队列设计：在proc\_struct中添加多个多级反馈队列，每个队列分别初始化并配置不同的优先级，在入队时先check进程的剩余时间片，不为0直接入队，若为0则降低优先级再入对应的队并重新设置时间片，出队过程不变，pick\_next按照优先级取进程，proc\_tick不变。

Ex2 实现Stride调度算法

按照注释提示实现了stride调度算法。

Make grade测试通过：



测试时需要将代码包中default\_sched\_stride\_c中的代码覆盖掉default\_sched.c

实现过程：

BigStride设置成我的学号..

初始化，入队，出队与rr算法类似

Pick\_next取斜堆顶即可，并且计算stride