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

```
priority:                 (156.9s)
  -check result:                                WRONG
   !! error: missing 'sched class: stride_scheduler'
   !! error: missing 'stride sched correct result: 1 2 3 4 5'
   !! error: missing 'all user-mode processes have quit.'
   !! error: missing 'init check memory pass.'

  -check output:                                OK
Total Score: 149/170
make: *** [grade] Error 1
```

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：

```
priority:                 (12.2s)
  -check result:                                WRONG
   !! error: missing 'sched class: stride_scheduler'
   !! error: missing 'stride sched correct result: 1 2 3 4 5'

  -check output:                                OK
Total Score: 163/170
```

- 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：

```
priority:                   (12.2s)
  -check result:                                OK
  -check output:                                OK
Total Score: 170/170
```

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