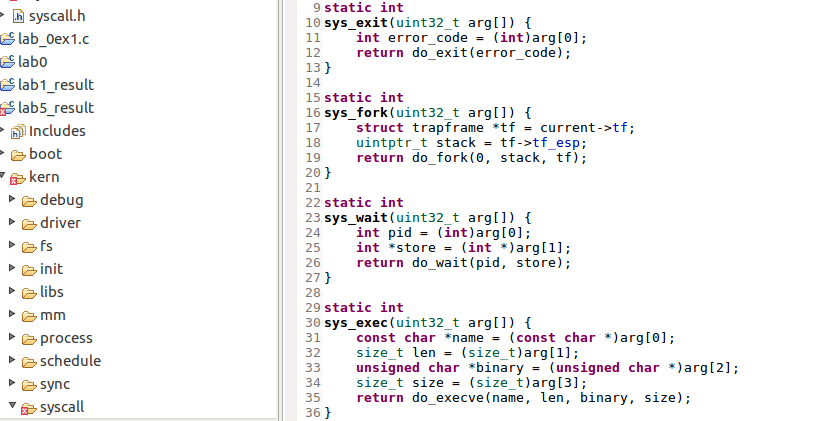
# Lab3

**Lab5:**

1.请分析fork/exec/wait/exit在实现中是如何影响进程的执行状态的？



fork：如果创建新进程成功，则出现一个子进程一个父进程。在子进程中，fork函数返回0，在父进程中，fork返回新创建子进程的进程ID。

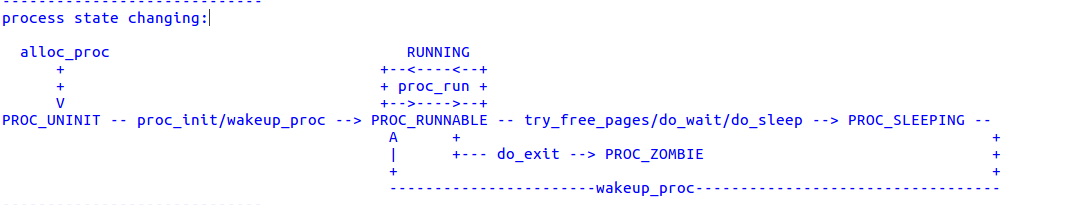
exit：会把一个退出码error\_code传递给操作系统，操作系统通过执行内核函数do\_exit来完成对当前进程的退出处理，工作是回收当前进程所占内存资源。

execve：完成用户进程的创建工作。首先为加载新的执行码做好用户态内存空间清空准备。再加载应用程序执行码到当前进程的新创建的用户态虚拟空间中。

wait：等待子进程的结束通知。wait\_pid函数等待进程id号为pid的子进程结束通知，让ucore来完成对子进程的最后回收工作。

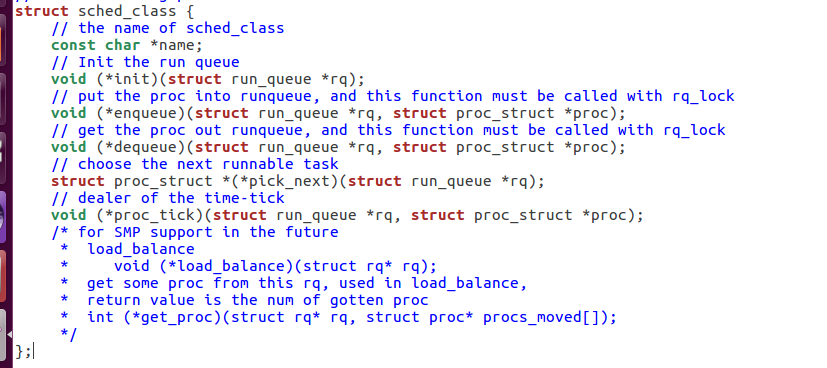
fork不会影响当前进程的执行状态，但是会将子进程的状态标记为RUNNALB，使得可以在后续的调度中运行起来；exec不会影响当前进程的执行状态，但是会修改当前进程中执行的程序；wait系统调用取决于是否存在可以释放资源（ZOMBIE）的子进程，如果有的话不会发生状态的改变，如果没有的话会将当前进程置为SLEEPING态，等待执行了exit的子进程将其唤醒；exit会将当前进程的状态修改为ZOMBIE态，并且会将父进程唤醒（修改为RUNNABLE），然后主动让出CPU使用权

2.请给出ucore中一个用户态进程的执行状态生命周期图（包执行状态，执行状态之间的变换关系，以及产生变换的事件或函数调用）。（字符方式画即可）



**Lab6:**

1.1 请理解并分析sched\_calss中各个函数指针的用法，并接合Round Robin 调度算法描述ucore的调度执行过程



 init：初始化运行队列

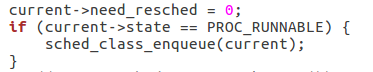
 enqueue：将进程p加入到运行队列rq中

 dequeue：将进程p从运行队列rq中删除

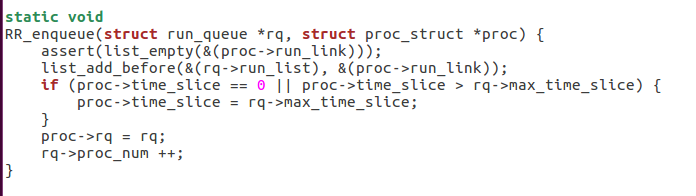
 pick\_next：返回运行队列中下一个可执行的进程

 proc\_tick：time tick的处理函数

当调用schedule函数时发生调度,清除当前进程需要调度的标记，把当前进程放进运行队列中去



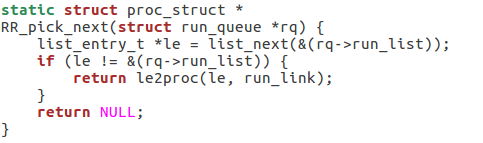
sched\_class\_enqueue函数调用Round Robin算法的RR\_enqueue，也就是把当前进程加入到运行队列的最后，将当前进程的时间片时间重置为最大，更新队列中进程数目

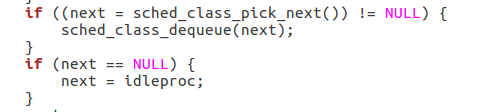


随后，通过Round Robin调度算法选择下一个要运行的进程。具体地，sched\_class\_pick\_next函数调用Round Robin算法的RR\_pick\_next接口

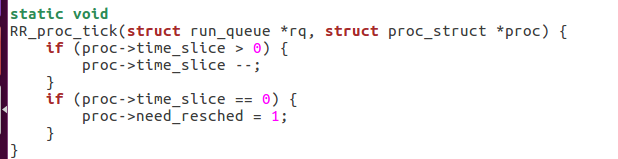


Round Robin算法的RR\_pick\_next将返回运行队列中的第一个进程交给schedule作为下一个要运行的进程



如果可以选出下一个要运行的进程，则将这个进程从运行队列中删除。否则，运行idleproc进程，即返回查询是否有新的进程需要运行。sched\_class\_dequeue函数会调用Round Robin算法的RR\_dequeue函数。  
  
随后通过proc\_run函数切换到新进程并进入新进程执行。proc\_run的具体切换过程在Lab5中已经完成。

此外，在每一次时钟tick的时候，都会调用Round Robin的RR\_proc\_tick函数，将当前进程的所占用的时间片剩余时间减一，当时间片耗尽时，设置为需要调度并等待调度中已经完成



1.2 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

在proc\_struct中添加总共N个多级反馈队列的入口，每个队列都有着各自的优先级，编号越大的队列优先级约低，并且优先级越低的队列上时间片的长度越大，为其上一个优先级队列的两倍；并且在PCB中记录当前进程所处的队列的优先级；

处理调度算法初始化的时候需要同时对N个队列进行初始化；

在处理将进程加入到就绪进程集合的时候，观察这个进程的时间片有没有使用完，如果使用完了，就将所在队列的优先级调低，加入到优先级低1级的队列中去，如果没有使用完时间片，则加入到当前优先级的队列中去；

在同一个优先级的队列内使用时间片轮转算法；

在选择下一个执行的进程的时候，有限考虑高优先级的队列中是否存在任务，如果不存在才转而寻找较低优先级的队列；

从就绪进程集合中删除某一个进程就只需要在对应队列中删除即可；

处理时间中断的函数不需要改变；

2. 实现 Stride Scheduling 调度算法,请在实验报告中简要说明你的设计实现过程

设置一个大整数，将来的所有pass值都由这个大整数除以进程的优先级得到

将进程加入运行队列时，插入到斜堆中，并将运行队列的进程计数加一，同时需要在进程的数据结构中关联运行队列

将进程从运行队列移走时，需要将进程从斜堆中删除，并将运行队列的进程计数减一

选择接下来调度的进程时，如果使用斜堆只需要取出堆顶，随后要更新进程的stride值。

处理时钟和RR算法一样，如果time slice大于0，则将值减一。否则以为着时间片用完，需要调度