操作系统实验报告：实验二

主负责人：12241014 何 雨

组员：13061137沈宏泽，13061144 李超，13061127 纪云钊

*备注：本次实验，我们整理了一份详尽的实验总结，所以详细内容请查看“第二次实验总结”文档。*

# 1需求说明

## 1.1基本要求和提高要求

本次实验的基本要求是完成十个debug的调试任务。

提高要求是实现一个多级轮转反馈的作业调度算法。

## 1.2完成情况

圆满完成了全部的基本要求和提高要求。

# 2设计说明

## 2.1所使用的系统调用的列表

Stat,remove,mkfifo,open,close,read,write,creat,kill,sihnal,sigaction,wait,waitpid,raise,bzero,getpid,getpuid,atoi等。

关于这些函数的详细说明在“第二次实验总结”文档中。

## 2.2提高要求实现说明

采用多级轮转反馈调度算法。

三个优先级：1，2，3，分别对应3个队列，运行时间分别为5s，2s，1s。

高优先级队列没有执行完毕的话，不会去轮转低优先级队列。

低优先级队列等待时间超过10s会升级，从而得到运行。

新加入的作业优先级高于当前作业的话，会进行抢占。

## 2.3实验分组说明

1．调试工作：未进行分配，要求所有组员都必须各自完成。

2．提高要求：

调度算法设计，具体工作分配--------------------何雨

修改do\_enq函数，设置三个链表队列------------------由纪云钊完成

修改jobselect函数------------------由李超完成

修改updatell函数----------------沈鸿泽，纪云钊

修改do\_stat函数--------------沈鸿泽

修改do\_deq函数---------------李超

修改jobswitch函数--------------------何雨

调度策略以及算法数据结构-----------------何雨

整合所有修改，统一处理------------------何雨

调试bug，改正错误---------------------何雨

会议记录、实验报告以及实验总结的撰写-----------------何雨

# 3收获与感想

## 3.1给予你帮助的人

感谢助教老师帮助我解决了为什么只加入一个作业无法运行的bug。

## 3.2从实验中学到的东西

1．并发程序执行时序的不确定性

第一个地方就是多进程运行的时候，进行执行顺序的重要性。而且这个错误通常很难被发现，因为进程的执行顺序是不可预测的，这就要求我们在写这样的并发程序的时候，尤其需要关注这种时序问题，并且充分利用号信号量来使得顺序可控。

2．指针虽好，却容易滥用

这个调度程序当中使用了大量的指针，尤其是提高要求的时候，设置了三个链表队列，整个程序到处都充斥了指针操作。

刚开始的时候，感觉C语言的指针真是太好用了，方便简单，还省事。

是的，写程序的时候确实省事啊，但是写完之后出错也是so easy啊！可以这么说吧，我们组用了大约6个小时的时间完成了提高要求的代码，然后，至少花了两倍的时间进行调试才使得程序运行正常。在这近乎让人崩溃的调试，所有找到的错误，全部都与指针有关。感觉，这真是一种讽刺！

通过函数调用传递一个指针，改变指针指向的内容，原函数也会跟着变化，但如果改变的是指针本身的内容，那原函数是肯定不会变化的。这应该是指针最常犯的错误吧。

所以，如果传指针的话，能不用子函数就不要用子函数了，调用的层数越多，指针越不可控。