**操作系统课程设计实验报告**

——实验二：schedule实验

负责人姓名：冯炜韬

学号：14061197

日期：2016.4.15

**小组成员**

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 姓名 | 学号 | 实验分工 |
| 1 | 张巍 | 14061186 | 实验一/三 |
| 2 | 冯炜韬 | 14061197 | 实验二/四 |

目录

目录 3

1.实验目的 4

2.需求说明 4

2.1基本要求 4

2.2 提高要求 4

2.3 完成情况 5

3.设计说明 6

3.1 程序流程图 6

3.2基本要求实现说明 6

3.3 提高要求实现说明 7

4.收获和感想 10

5.实验遇到的问题及解决方法 10

# 1.实验目的

1.理解操作系统中调度的概念与调度策略

2.学习Linux系统进程控制以及进程间通信的概念与方法

3.理解并掌握几种常用的调度算法，能分析各算法的特征和优劣

# 2.需求说明

## 2.1基本要求

本实验要求实现一个作业调度程序，通过该程序可以完成作业入队、出队、查看和调度。具体要求如下：

1）实现作业调度程序scheduler，负责整个系统的运行

这是一个无限循环运行的进程，其任务是响应作业的入队、出队及状态查看请求，采用适当的算法调度各作业运行。

2）实现作业入队命令。用户通过该命令给scheduler发送入队请求，将作业提交给系统运行。每个作业提交以后，若创建成功，scheduler都将为其分配一个唯一标识jid。Scheduler调度程序为每个作业创建一个进程，并将其状态置为READY，然后放入就绪队列中，打印作业信息。

3）实现作业出队命令。用户通过该命令给scheduler发送出队请求，scheduler将使改作业出队，然后清除相关的数据结构。若该作业正在运行，则需先终止其运行。每个用户都将只能杀掉自己提交的作业。

4）实现作业状态查看命令。在标准输出上打印就绪队列中各作业的信息。

5）实现多级反馈的轮转调度算法。

每个作业都有其动态的优先级，在用完分配的时间片之后，可以被优先级更高的作业抢占运行。就绪队列中的进程等待时间越长，其优先级越高。

## 2.2 提高要求

1）改进源码中的调度算法，实现更高效更合理的调度

2）将stat打印输出通过fifo传输给命令程序

## 2.3 完成情况

完成了以上提出的所有要求

# 3.设计说明

## 3.1 程序流程图

时间片用完

向多级队列查询

多级队列发现的最高优先级任务

大于等于当前任务优先级？

将该任务拿出来，并将当前任务放回队列

执行下一个任务

是

否

## 3.2基本要求实现说明

源程序给的代码基本已经实现基本要求，这里仅仅是修改了一些bug，比如第一个任务不会自动进行，切换任务时导致队列内容消失等。

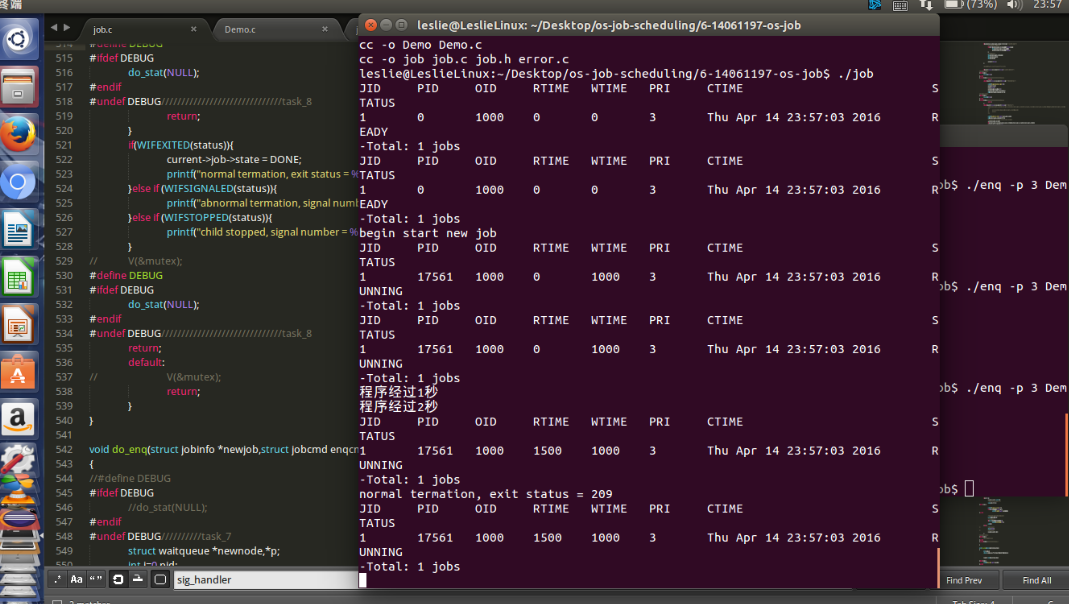
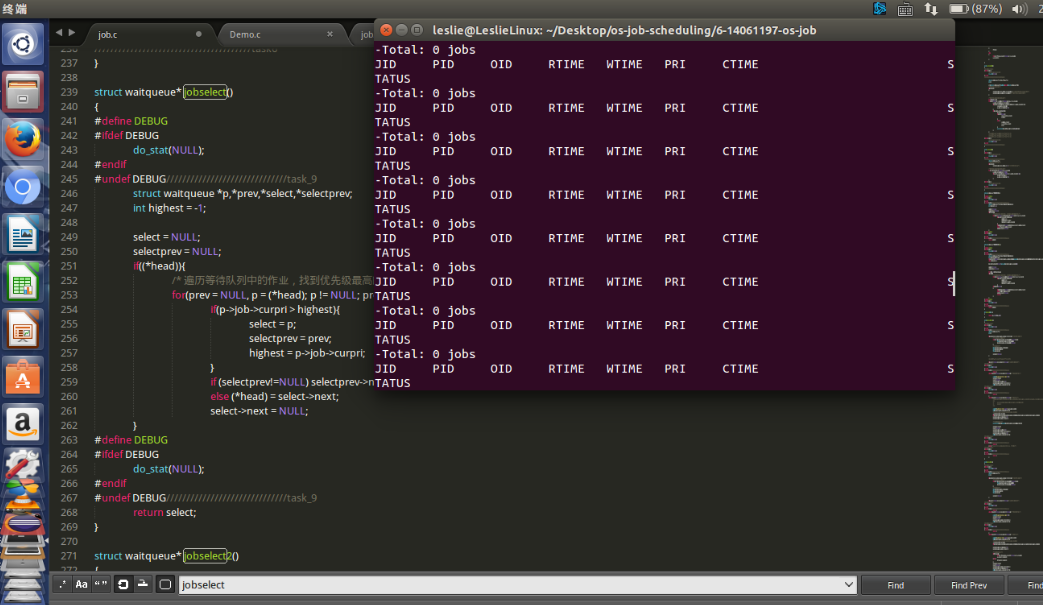
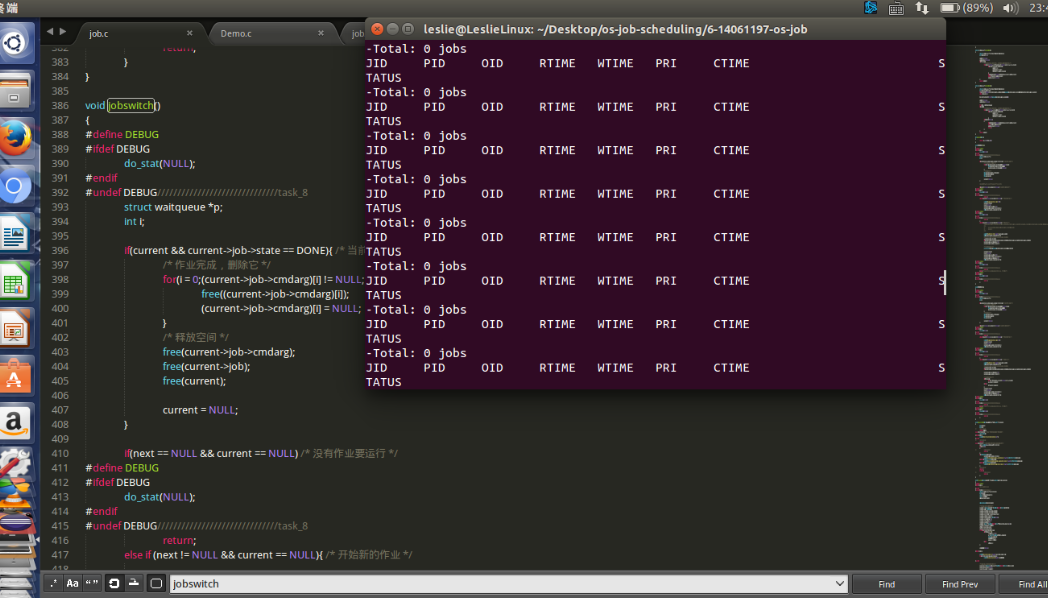
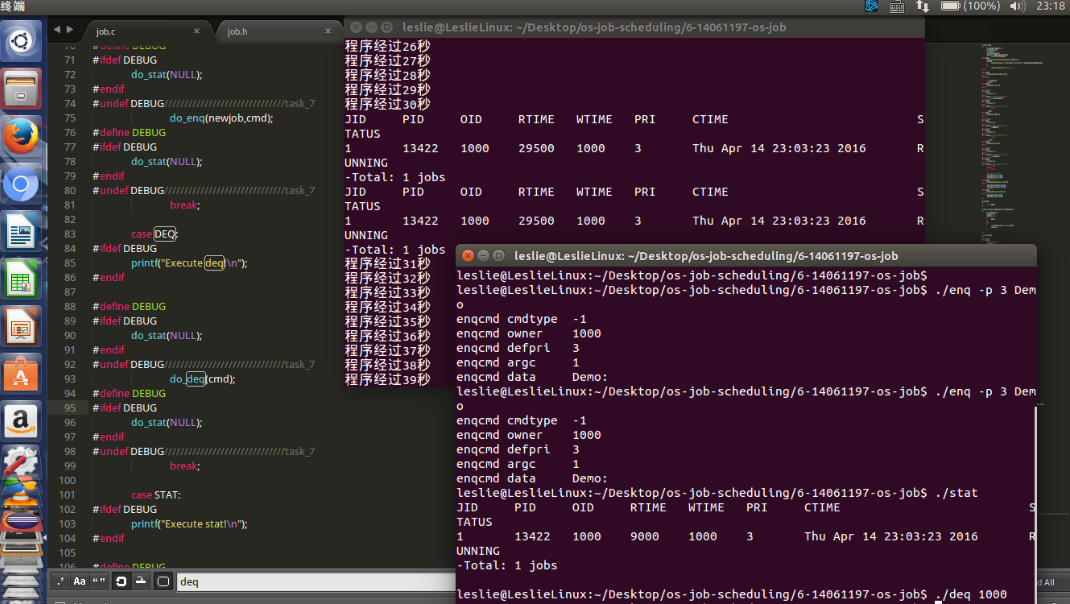
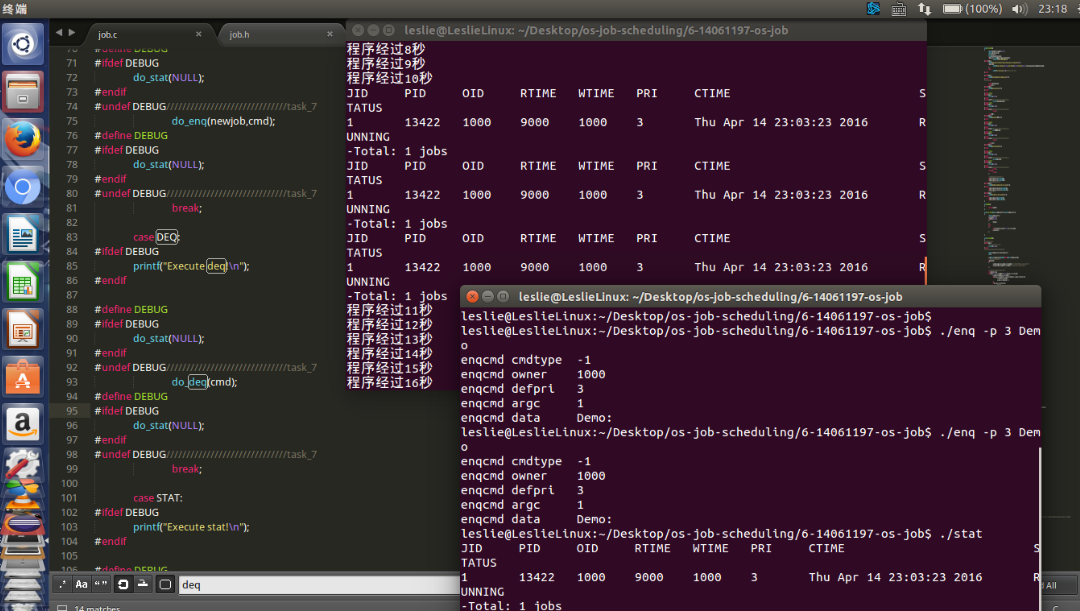
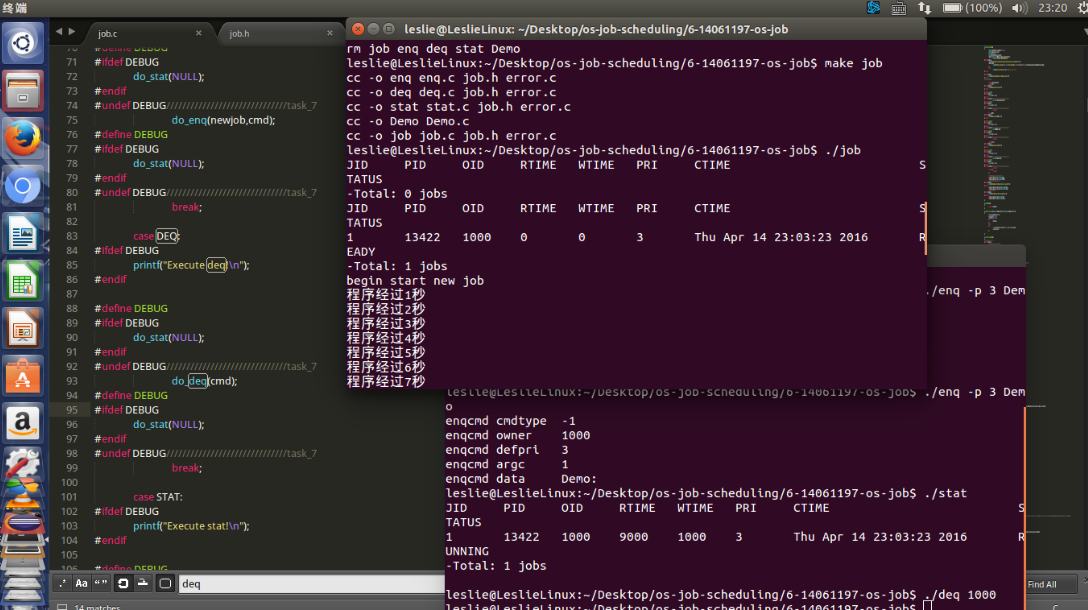
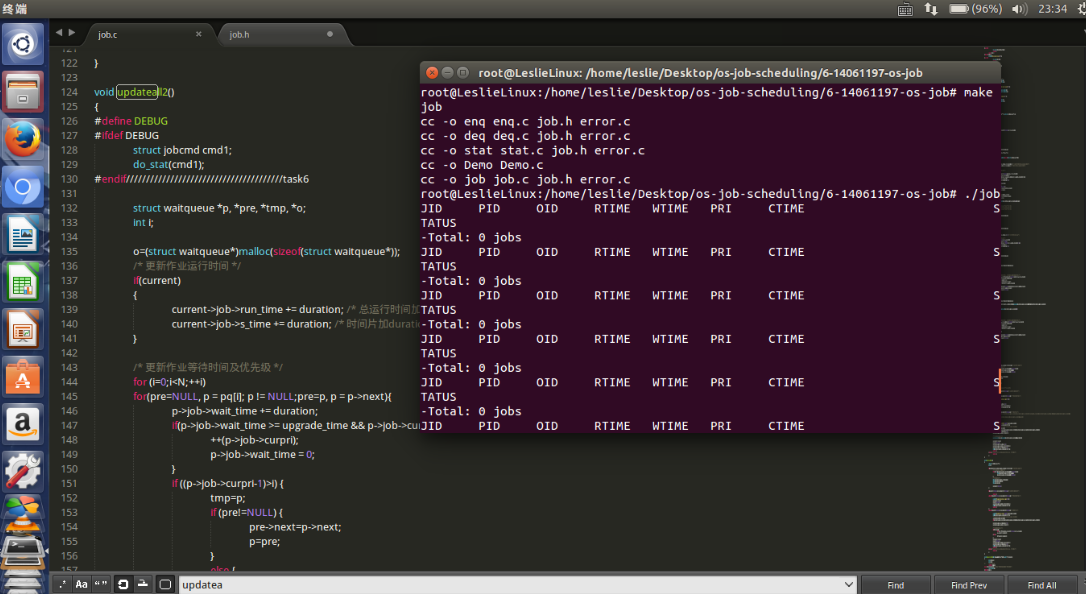
## 3.3 提高要求实现说明

1）增加另外两个任务队列，构成三队列系统，高级队列优先轮转，高级队列为空才轮转低级队列。抢占式运行，当新任务到达，如果优先级比当前优先级高，或者等于当前优先级但当前时间片已用完，则抢占运行机会。

2）scheduler收到stat指令，将输出用springf函数转为字符串，再写入fifo中，传回给指令程序

3）增加调试信息，这个比较简单和重复，不赘述

4）调试结果及代码截屏



# 4.收获和感想

这个实验比上次的简单许多，但是不明白为什么调试信息输出竟变成硬性的提高要求，这个问题是本末倒置的，调试就是为了在程序出错的时候进行检测以便修改，现在程序做好了还要按照规定输出调试信息，基本没有意义。

# 5.实验遇到的问题及解决方法

没有什么问题或者困难，FIFO操作不会的，上网找一下系统调用的用法就好了。