编号：

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实验 | 一 | 二 | 三 | 四 | 五 | 六 | 七 | 八 | 九 | 十 | 总评 | 教师签名 |
| 成绩 |  |  |  |  |  |  |  |  |  |  |  |  |

武汉大学计算机学院

课程实验(设计)报告

专业(班)： 计算机科学与技术5班

学 号： 2016301500187

姓 名： 董夏磊

课程名称： 操作系统设计

任课教师： 宋伟

2018年 11 月 17 日

**处理器调度**

1. **实习题目**
2. 实习内容

选择一个调度算法，实现处理器调度。

1. 实习目的

本实习模拟在单处理器环境下的处理器调度，加深了解处理器调度的工作。

1. 实习题目

设计一个按优先数调度算法实现处理器调度的程序。

(1)假定系统有5个进程，每个进程用一个PCB来代表。PCB的结构为：

·进程名——如P1~P5。

·指针——按优先数的大小把5个进程连成队列，用指针指出下一个进程PCB的首地址。

·要求运行时间——假设进程需要运行的单位时间数。

·优先数——赋予进程的优先数，调度时总是选取优先数大的进程先执行。

·状态——假设两种状态：就绪和结束，用R表示就绪，用E表示结束。初始状态都为就绪状态。

(2) 每次运行之前，为每个进程任意确定它的“优先数”和“要求运行时间”。

(3) 处理器总是选择队首进程运行。采用动态改变优先数的办法，进程每运行1次，优先数减1，要求运行时间减1。

(4) 进程运行一次后，若要求运行时间不等于0，则将它加入就绪队列，否则，将状态改为“结束”，退出队列。

(5) 若就绪队列为空，结束，否则转到(3)重复。

要求能接受键盘输入的进程优先数及要求运行时间，能显示每次进程调度的情况，如哪个进程在运行，哪些进程就绪，就绪进程的排列情况。

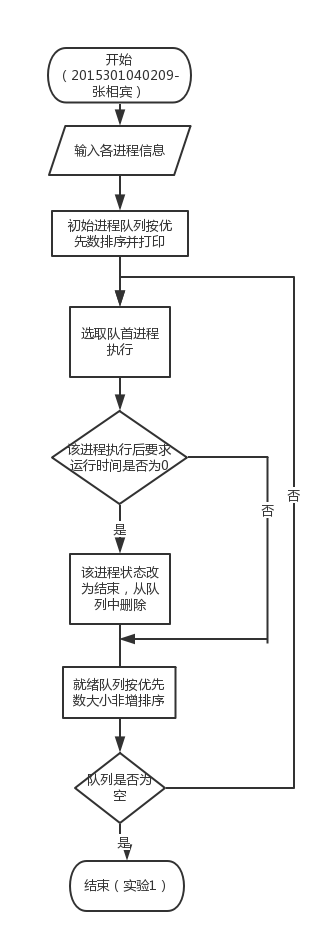
1. **上机实习所用平台及相关软件及语言**

Visual Studioo 2017 + Windows10

C++编写

1. **实习内容及设计思想**
2. 设计思路

程序流程图如下：



1. 主要数据结构

Class process{ //进程类

Void run(); //进程执行

Bool check(); //检查进程状态

String id; //进程名

Int asktime; //要求运行时间

Int pri; //优先数

Char state; //两种状态，E代表结束，R代表就绪

}

Deque<process> pro; //队列容器，初始进程队列

Deque<process> proR; //队列进程，就绪进程队列

Void runFir(); //执行队首进程

Void checkIter(); //检查迭代器指向进状态并执行对应操作

Bool isPriGreater(); //模板排序函数Sort自定义排序方式按优先数非增排序

1. 主要代码结构

//当就绪队列不为空时

while (!proR.empty()) {

//运行队首进程

runFir(proR);

//迭代器指向队首进程

it = proR.begin();

cout << "正在运行程序：";

cout << "\t\t进程" ;

cout << it->id << endl;

cout << "运行中......." << endl;

Sleep(2 \* 1000);

//该次运行结束

//检查队首进程要求运行时间是否为0

checkIter(proR.begin(), proR);

//就绪队列排序

sort(proR.begin(), proR.end(),isPriGreater);

//打印就绪队列

cout << "第" << num + 1 << "次运行结束后就绪队列：" << endl;

num++;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "排列顺序" << "\t进程ID" << "\t\t剩余运行时间" << "\t优先级" << endl;

it = proR.begin();

for (unsigned i = 0; i < proR.size(); i++) {

cout << (i + 1) << "-->" << "\t\t" << it->id << "\t\t"

<< it->asktime << "\t\t" << it->pri << endl;

it++;

}

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

}

cout << "就绪队列为空，结束..." << endl;

1. 代码段分析

见上述注释及流程图

1. **调试过程**
2. 测试数据设计

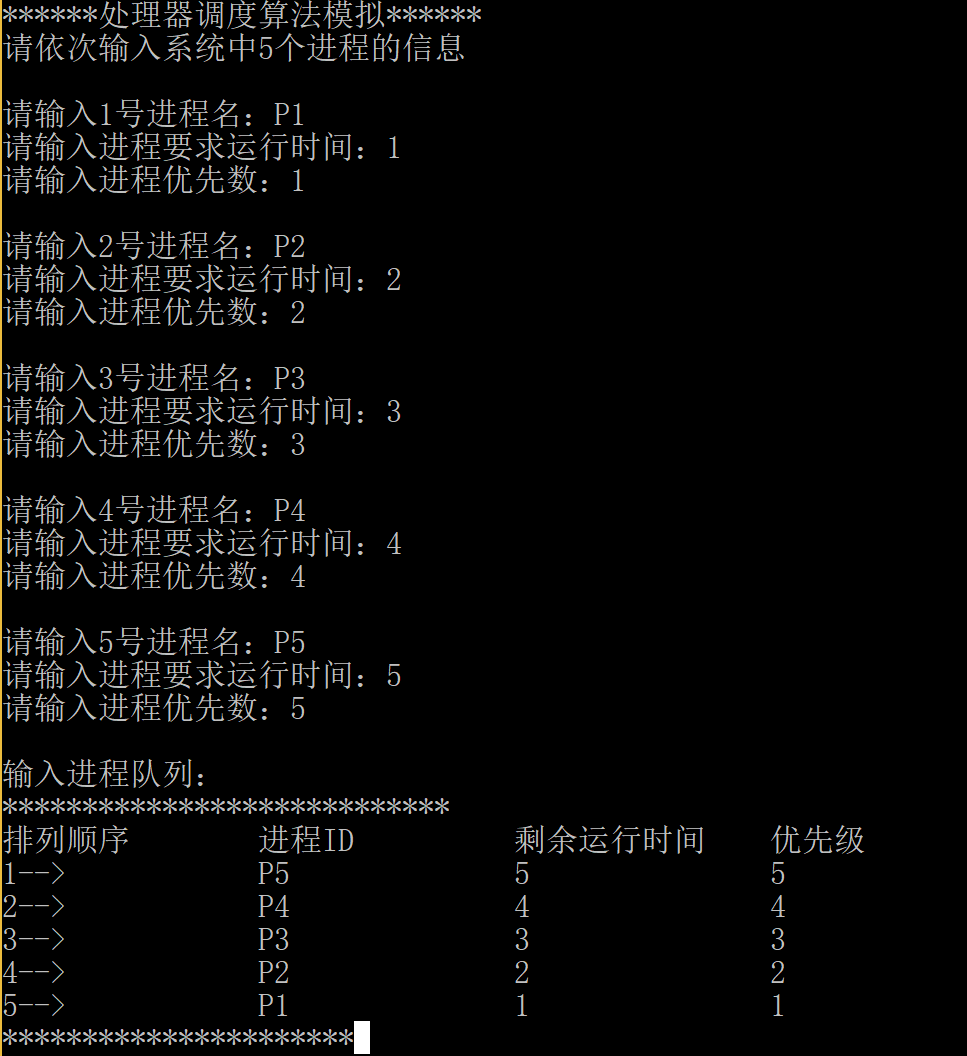
初始进程：

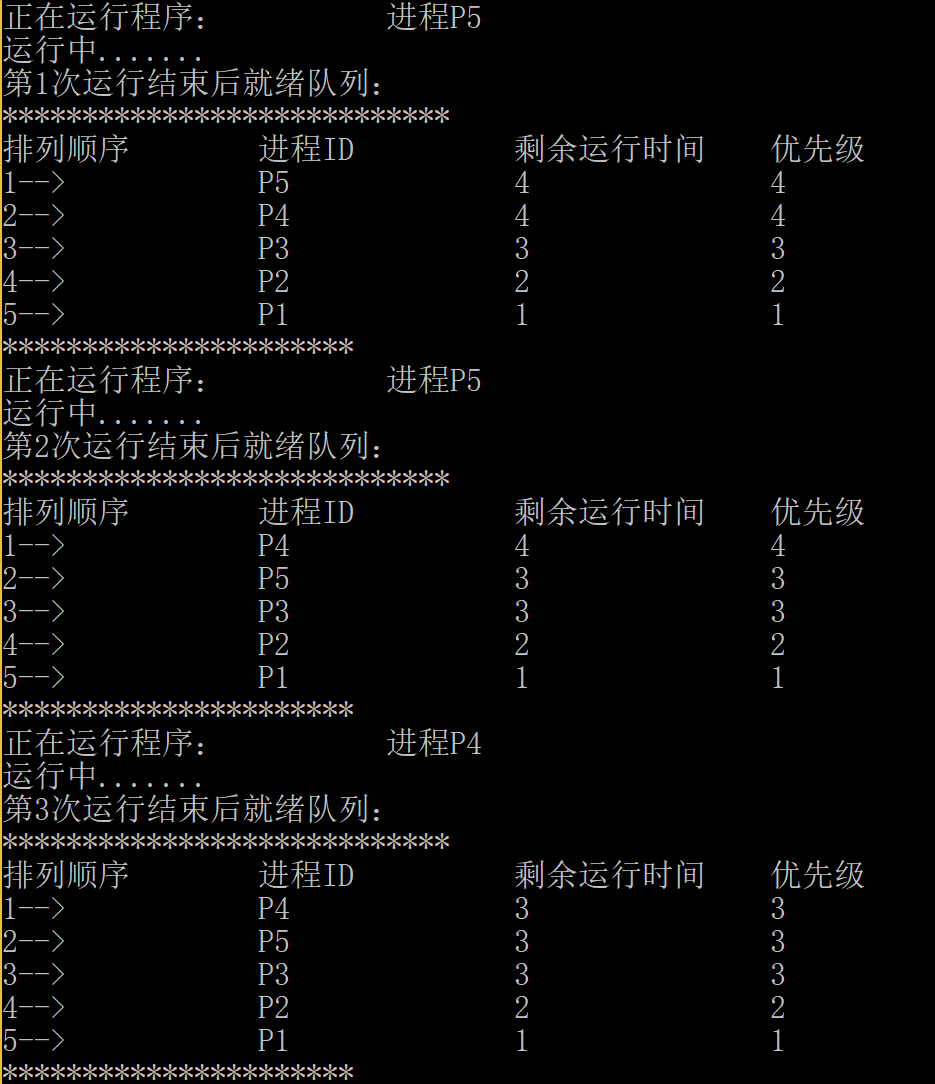
|  |  |  |
| --- | --- | --- |
| 进程名 | 要求运行时间 | 优先数 |
| P1 | 1 | 1 |
| P2 | 2 | 2 |
| P3 | 3 | 3 |
| P4 | 4 | 4 |
| P5 | 5 | 5 |

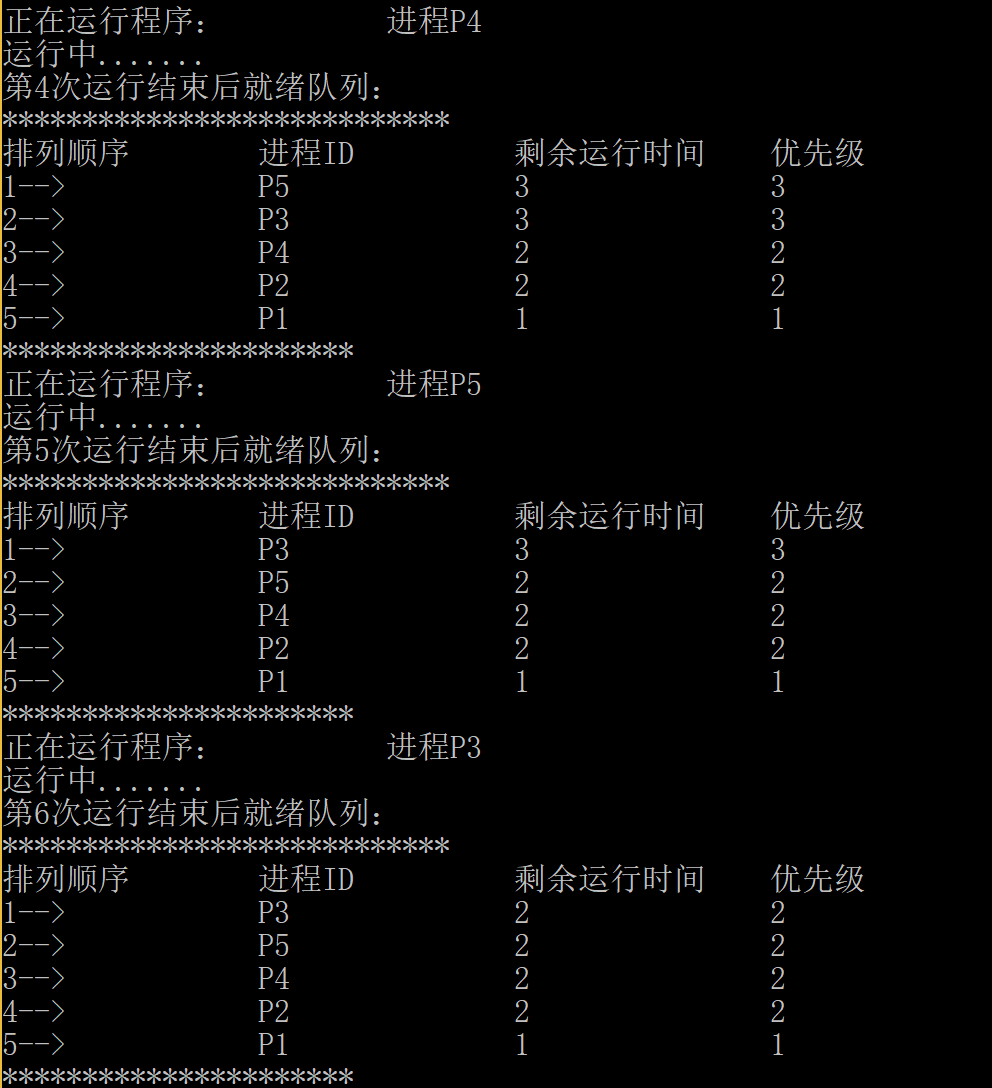
预测执行序列：

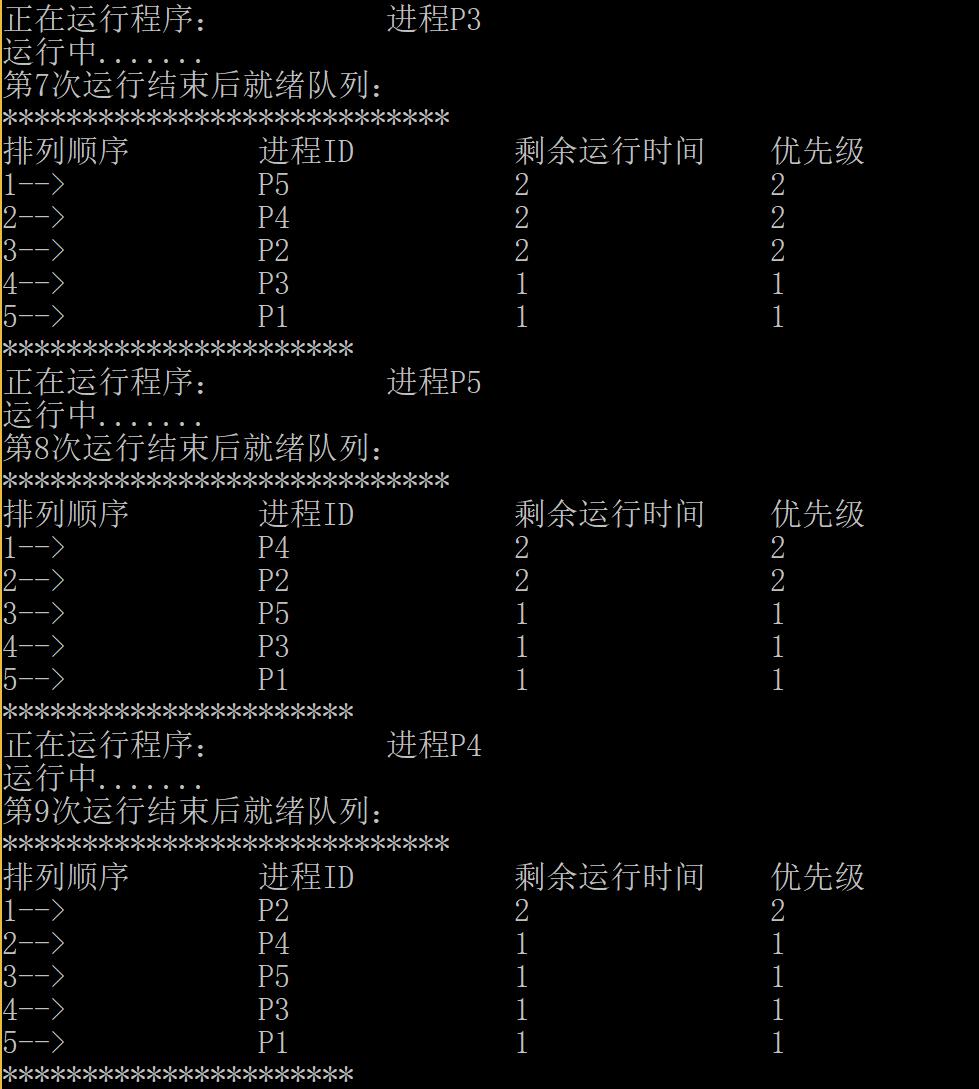
P5->P5->P4->P4->P5->P3->P3->P5->P4->P2->P2->P4->P5->P3->P1

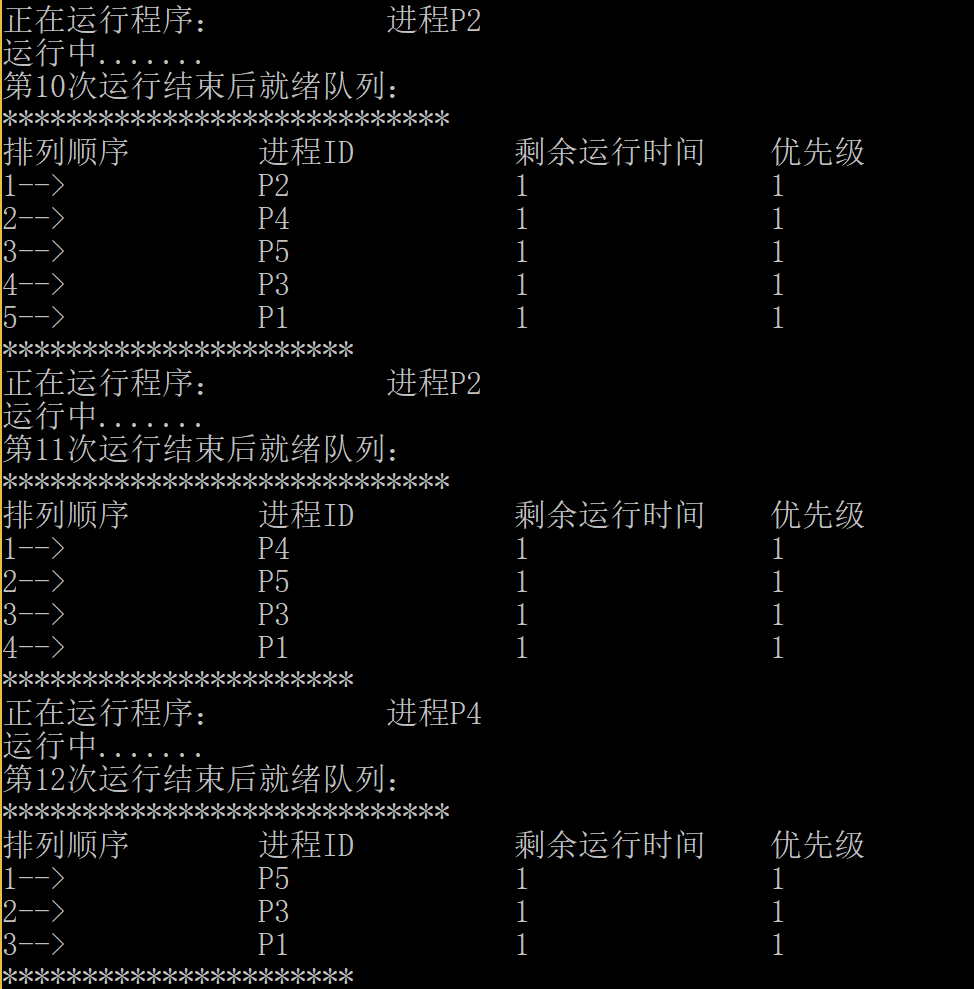
1. 测试结果分析













与预测相符

1. **总结**
2. 实习中遇到的问题及解决方法

（1）采用deque容器表示进程队列，使用模板函数sort进行排序并自己编写，没有使用list，因为其sort算法较特殊为list.sort();

（2）关于容器增删后迭代器失效的问题在编写时尤为注意

1. 实习中产生的错误及原因分析
2. 实习体会及收获

在刚刚学习过c++后，将其中的容器和模板函数应用到操作系统实验中，加深了对操作系统对进程调度的理解，巩固了c++语言的掌握。