**武汉大学计算机学院**

**本科生实验报告**

**处理器调度**

专 业 名 称 ：计算机科学与技术

课 程 名 称 ：操作系统设计

指 导 教 师 ：宋伟

学 生 学 号 ：

学 生 姓 名 ：

二○二一年七月

**郑 重 声 明**

本人呈交的实验报告，是在指导老师的指导下，独立进行实验工作所取得的成果，所有数据、图片资料真实可靠。尽我所知，除文中已经注明引用的内容外，本实验报告不包含他人享有著作权的内容。对本实验报告做出贡献的其他个人和集体，均已在文中以明确的方式标明。本实验报告的知识产权归属于培养单位。

本人签名： 日期： 2021.7.2

摘 要

处理器调度实验的实验目的是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。

实验设计主要遵循实验设计原理。

实验内容主要包括：选择一个调度算法，实现处理器调度。

实验结论为该实验让我重温了时间片轮转法，加深了我对处理器调度的理解，让我更清晰地了解每个进程在处理器中的执行情况。

**关键词：**处理器调度

**目 录**

**1** **处理器调度**

1.1 实验内容及上机实验所用平台 ……………………………………………………… 1

1.2 数据结构及代码段分析 ……………………………………………………………… 1

1.3 调试过程 ………………………………………………………………………………… 3

1.4 实验总结 ………………………………………………………………………………… 5

**参考文献** …………………………………………………………………………………… 6

**1** **处理器调度**

**1.1 实验内容及上机实验所用平台**

本实验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。

**1.1.1 实验内容**

设计一个按时间片轮转法实现处理器调度的程序。

**1.1.2 实验设计**

（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——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时间单位数。

·已运行时间——进程已运行的时间单位数，初始值为0。

·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）运行之前，为每个进程确定它的“要求运行时间”。通过键盘输入这些参数。

（3）把5个进程按顺序排成循环队列，用指针指出队列连接情况。用一个标志单元记录轮到运行的进程。处理器调度总是选择标志单元指示的进程运行，对所指的进程，将其“已运行时间”加1。

（4）进程运行一次后，若“要求运行时间”等于“已运行时间”，则将状态改为“结束”，退出队列，否则将继续轮转。

（5）若就绪队列为空，结束，否则转到（3）重复。

**1.1.3 上机实验所用平台**

开发软件采用VSCode，编程语言使用C++。

**1.2 数据结构及代码段分析**

**1.2.1 数据结构**

为PCB定义一个数据结构，其中包括：

·进程名

·指向下一个PCB的指针

·要求运行时间

·已运行时间

·状态

根据上述数据结构创建5个PCB实例，并构成一个循环队列。

表**1.1** 每个PCB的指针指向的PCB

|  |  |
| --- | --- |
| PCB进程名 | 指向的PCB进程名 |
| P1 | P2 |
| P2 | P3 |
| P3 | P4 |
| P4 | P5 |
| P5 | P1 |

**1.2.2 代码段分析**

重点代码段如下：

PCB \*former = &P5, \*now = &P1;

int time = 0;

while(1){

now->runTime++;

cout << "在时间段" << time << "-" << ++time << "内" << endl;

cout << "进程" << now->name << "在运行" << endl;

if(now->requestTime <= now->runTime){

if(now->next == now){

cout << "运行后所有进程执行完毕" <<endl;

break;

}

now->condition = 'E';

former->next = now->next;

}

else{

former = former->next;

}

now = now->next;

PCB \*temp = now;

cout << "运行后就绪队列情况为";

cout << temp->name;

while(temp->next != now){

temp = temp->next;

cout << " -> " << temp->name;

}

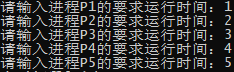
cout << endl << endl;

}

在该代码中设置了两个指针，now指针指向当前需要运行的进程，former指针指向now在循环队列中当前需要运行的进程的前一个进程。在循环中每次将now指针指向的进程的已运行时间加1，然后判断当前所有进程是否都执行完毕，如果是，则循环队列中应该只有一项，而且该项的已运行时间大于等于要求运行时间。如果有进程尚未执行完，则要判断当前now指针指向的进程是否执行完毕，若执行完毕则将其移出循环队列，并将now指针指向下一个要运行的进程。如果当前now指针指向的进程未执行完，则将former指针和now指针均向后移一个进程。

**1.3 调试过程**

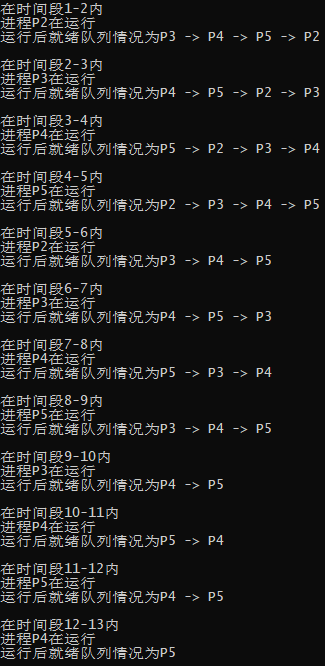
我们将P1-P5的要求运行时间分别设置为1、2、3、4、5，得到的输出结果见以下图片。



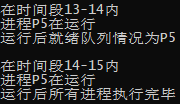
图**1.1** 输入五个进程的要求运行时间



图**1.2** 时间段0-1运行结果



图**1.3** 时间段1-2至时间段12-13运行结果



图**1.4** 时间段13-14和时间段14-15运行结果

**1.4 实验总结**

该实验让我重温了时间片轮转法，加深了我对处理器调度的理解，让我更清晰地了解每个进程在处理器中的执行情况。在编写代码的过程中我也遇到了一些问题，例如在定义结构体时忘记将构造函数里的进程名加引号，导致拖延了一些时间；在former指针和now指针的变化上最初也没有考虑的太清楚，导致得到了错误的结果。但这些问题在经过我的仔细检查和更加谨慎的思考下都得到了有效的解决。总体来说我认为该实验原理并不是特别复杂，但能够加强我对处理器调度算法的认知，让我有许多收获！

**参考文献**

[1] 郑鹏，曾平，金晶.计算机操作系统[M].武汉:武汉大学出版社，2014.7

教师评语评分

评语：

评分：

评阅人：

年 月 日

（备注：对该实验报告给予优点和不足的评价，并给出百分之评分。）