操 作 系 统 原 理

实 验 报 告

实验题目： 实验 2：时间片轮转进程调度

实验时间： 2022 年 11 月 9 日

学号姓名： 2022129390 吕强

实验目的和要求

1．掌握时间片轮转进程调度的概念和算法。

2．加深对处理机分配的理解。

实验内容及步骤

在 Visual C++ 6.0 集成开发环境下使用 C 或 C++语言，利用相应的 WIN32 API 函数，编写程序实现时间片轮转进程调度算法，学会运行程序和中断当前程序的运行。

1．在源程序上写出注释。

2．画出程序流程图。

3．调试程序并写出运行结果。

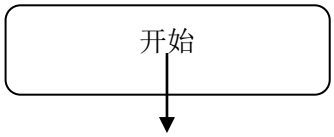
实验程序

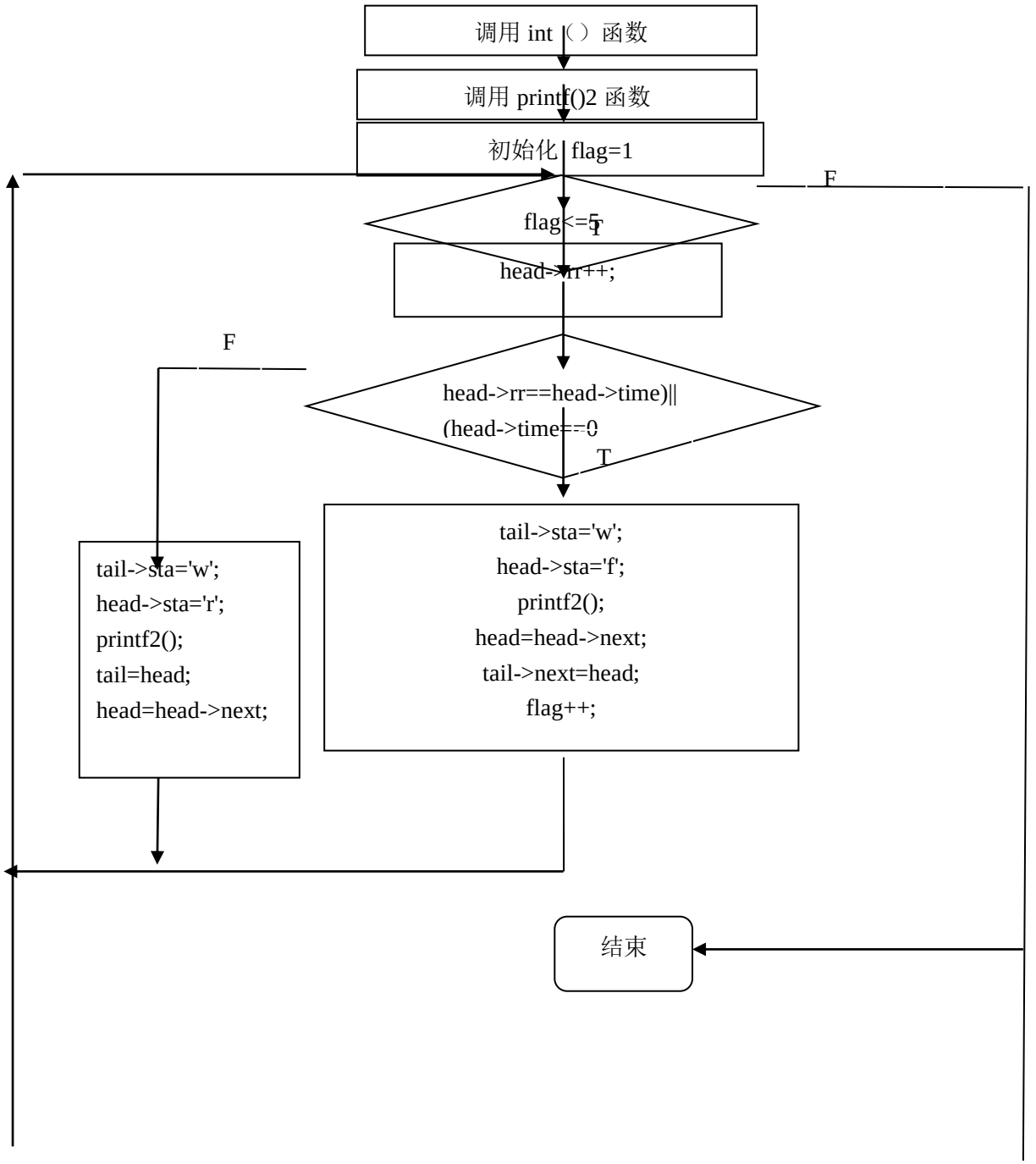
```
#include <stdio.h>
    #include <stdlib.h>
    struct PCB
    {
        int pid;//进程标识符
        int rr;//已运行时间
        int time;//进程要求运行时间
        char sta;//进程的状态
        struct PCB *next;//链接指针
    };
    struct PCB pcb1,pcb2,pcb3,pcb4,pcb5,*tail,*head,*rp;
    void init()//初始化各个进程的运行时间
    {
        int time;
        pcb1.pid = 1;
        pcb2.pid = 2;
        pcb3.pid = 3;
```

```
    pcb4.pid = 4;
    pcb5.pid = 5;
    pcb1.rr =pcb2.rr =pcb3.rr =pcb4.rr =pcb5.rr = 0;
    pcb1.sta = pcb2.sta = pcb3.sta = pcb4.sta = pcb5.sta = 'w';
    printf("请输入进程 p1 需要运行的时间:");
    scanf("%d",&time);
    pcb1.time =time;
    printf("请输入进程 p2 需要运行的时间:");
    scanf("%d",&time);
    pcb2.time=time;
    printf("请输入进程 p3 需要运行的时间:");
    scanf("%d",&time);
    pcb3.time=time;
    printf("请输入进程 p4 需要运行的时间:");
    scanf("%d",&time);
    pcb4.time =time;
    printf("请输入进程 p5 需要运行的时间:");
    scanf("%d",&time);
    pcb5.time=time;
    pcb1.next=&pcb2;
    pcb2.next=&pcb3;
    pcb3.next=&pcb4;
    pcb4.next=&pcb5;
    pcb5.next=&pcb1;
    head=&pcb1;
    tail=&pcb5;
}
void printf1()//显示表头
{
    printf("+---------------|---------------|---------------|---------------+\n");
    printf("|\tpid\t|\trr\t|\ttime\t|\tSTA\t|\n");
    printf("|---------------|---------------|---------------|---------------|\n");
}
void printf2()//显示各个进程的初始状态
{
    printf("processes p%d running\n",head->pid);
    printf1();
    printf("|\t%d\t|\t%d\t|\t%d\t|\t%c\t|\n",head->pid,head->rr,head->time,head->sta);
    printf("|---------------|---------------|---------------|---------------|\n");
    rp=head;
    while(rp!=tail)
    {
        rp=rp->next;
        printf("|\t%d\t|\t%d\t|\t%d\t|\t%c\t|\n",rp->pid,rp->rr,rp->time,rp->sta);
```

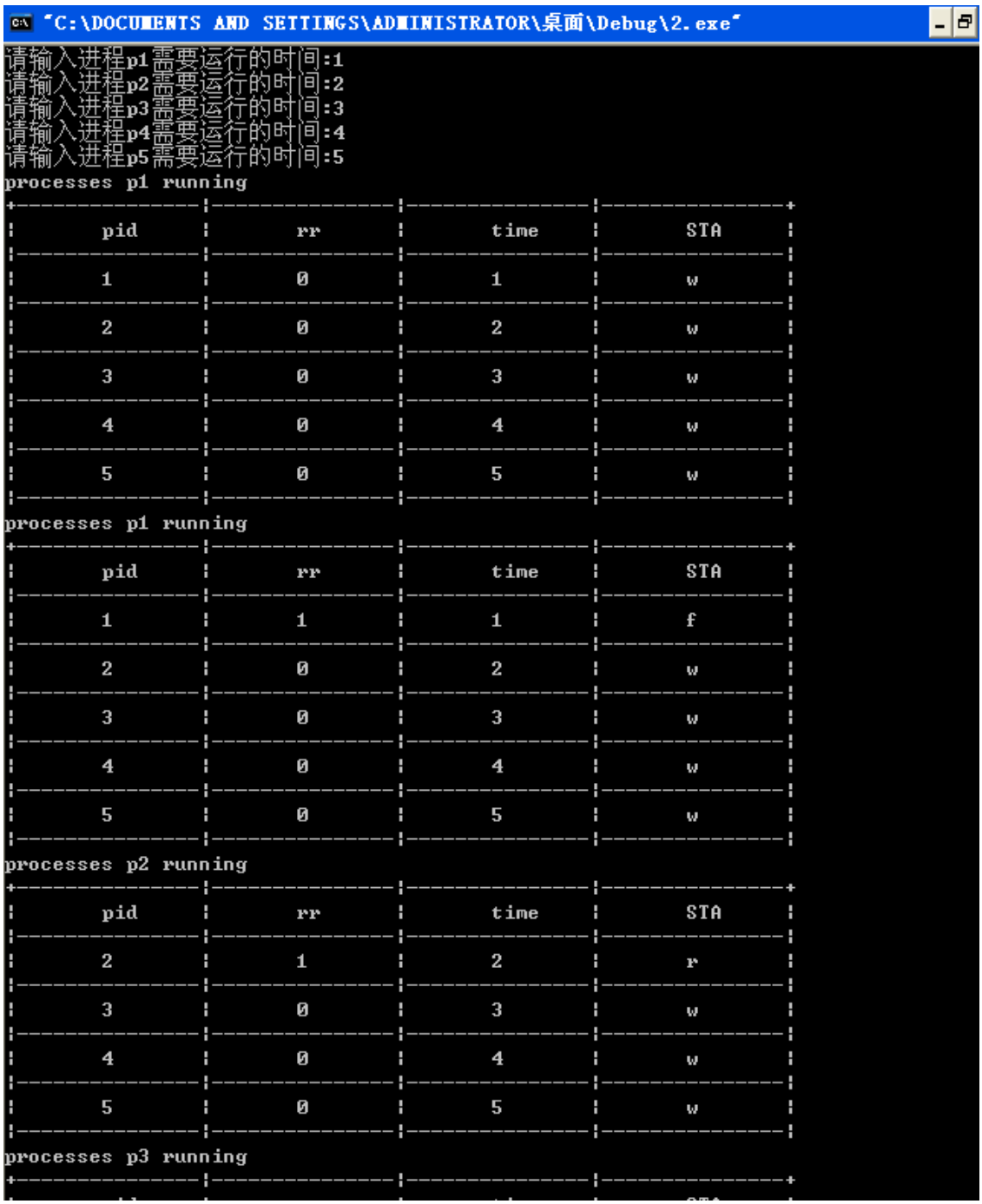
```
            printf("|---------------|---------------|---------------|---------------|\n");
        }
    }
    void operation()//运行
    {
        int flag=1; //定义标志位
        while(flag<=5) //定义 while 循环
        {
            head->rr++;//头指针进行下一位位移
            if((head->rr==head->time)||(head->time==0))
             //if 语句判断如果头部位等于执行时间
            {
                tail->sta='w';//将进程状态设置为等待态
                head->sta='f';//将进程状态设置为执行态
                printf2();
                head=head->next;
                tail->next=head;
                flag++;
            }
            else
            {
                tail->sta='w';//将进程状态设置为等待态
                head->sta='r';//将进程状态设置为就绪态
                printf2();
                tail=head;
                head=head->next;
            }
        }
    }
void main()
    {
        init();
        printf2();
        operation();
    }
```

程序流程图




调用 int（）函数
调用 printf()2 函数
初始化 flag=1
F
flag<=5
T
head->rr++;
F
head->rr==head->time)||
(head->time==0
T
tail->sta='w';
head->sta='r';
printf2();
tail=head;
head=head->next;
tail->sta='w';
head->sta='f';
printf2();
head=head->next;
tail->next=head;
flag++;
结束

运行结果


"C:\DOCUMENTS AND SETTINGS\ADMINISTRATOR\桌面\Debug\2.exe"
请输入进程p1需要运行的时间:1
请输入进程p2需要运行的时间:2
请输入进程p3需要运行的时间:3
请输入进程p4需要运行的时间:4
请输入进程p5需要运行的时间:5
processes p1 running
pid rr time STA
1 0 1 w
2 0 2 w
3 0 3 w
4 0 4 w
5 0 5 w
processes p1 running
pid rr time STA
1 1 1 f
2 0 2 w
3 0 3 w
4 0 4 w
5 0 5 w
processes p2 running
pid rr time STA
2 1 2 r
3 0 3 w
4 0 4 w
5 0 5 w
processes p3 running

分析讨论

在运行过程中需要将 viod 改为 int 才可以正常运行，以及有关流程图的走向，画法及 operation()的展开介绍。if-else 语句的判断以及后面条件的执行过程。

教师评语及成绩