实验三 进程调度

一、 实验类型

本实验为综合性实验

二、 实验目的与任务

在采用多道程序设计的系统中，往往有若干个进程同时处于就绪状态。当就绪进程个数大于处理机数时，就必须依照某种策略来决定那些进程优先占用处理机。本实验模拟在单处理机情况下的处理机调度，帮助学生加深了解处理机调度的工作。

三、 预习要求

1）熟悉进程控制块和进程组织方式

2）熟悉进程调度的概念

3）熟悉时间片轮转调度算法等

4）熟悉 c 语言编程，指针及结构体等知识

5）数据结构中的链表的建立及基本操作

四、 实验基本原理

进程控制块通过链表队列的方式组织起来，系统中存在运行队列和就绪队列（为简单起见，不设阻塞队列），进程的调度就是进程控制块在运行队列和就绪队列之间的切换。当需要调度时，从就绪队列中挑选一个进程占用处理机，即从就绪队列中删除一个进程，插入到运行队列中，当占用处理机的进程运行的时间片完成后，放弃处理机，即在运行队列中的进程控制块等待一段时间（时间片）后，从该队列上删除，如果该进程运行完毕，则删除该进程（节点）；否则，则插入到就绪队列中。

五、 实验仪器与设备（或工具软件）

实验设备：计算机一台，软件环境要求：安装 Red Hat Linux 操作系统和 gcc 编译器。

六、 实验内容

设计一个时间片轮转调度算法实现处理机调度的程序，具体内容如下

1）实验中使用的数据结构

(1) PCB 进程控制块

内容包括参数①进程名 name；②要求运行时间 runtime；③优先数 prior；④状态

state；⑤已运行时间 runedtime。

(2) 为简单起见，只设运行队列，就绪链表两种数据结构，进程的调度在这两个队列中切换，如图 3.1 所示

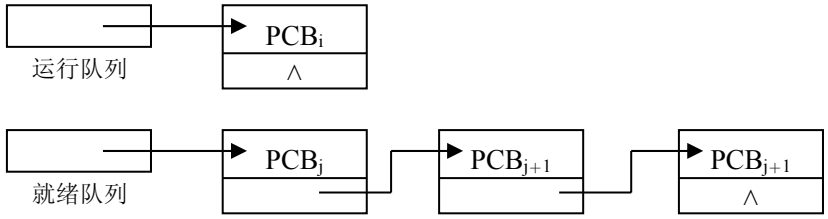

图 3.1PCB 链表

2）每个进程运行时间随机产生，为 1~20 之间的整数。

3）时间片的大小由实验者自己定义，可为 3 或 5

4）可参考的程序流程图如图 3.2

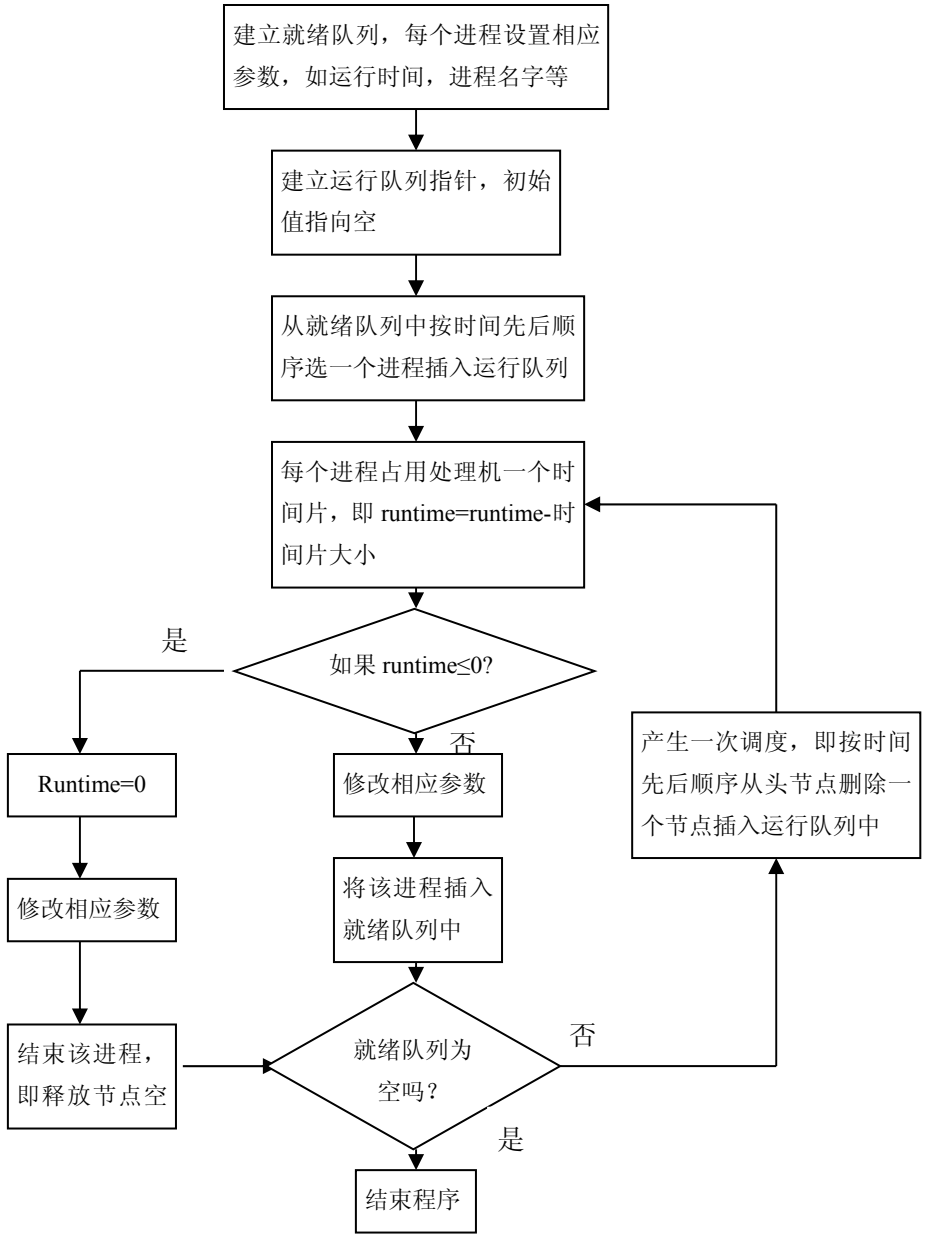

图 3.2 模拟进程调度的流程图

5）参考程序

```
#include "stdio.h"
#include "stdlib.h"

typedef struct PCB
{
        int name;
        int runtime;
        int runedtime;
        int state;
        int killtime;
   struct PCB *next;
}PCB;
#define NUM 10

void main()
{
      int timeslice=3;
      PCB *runqueue;
      PCB *top,*tail,*temp;
      int i;
      srand(10);
    for(i=0;i<NUM;i++)
      {
            temp=new PCB;
            temp->name=i;
            temp->runtime=rand()%20;
            temp->runedtime=0;
            temp->next=NULL;
            temp->killtime=0;
           if(i==0) {top=temp; tail=temp;}
           else{
                tail->next=temp;
                tail=temp;
           }
           printf("process name %d, runtime=%d, runedtime=%d,killtime=%d\n"
           ,tail->name,tail->runtime,tail->runedtime,tail->killtime);
      }

      while(top!=NULL)
      {
        runqueue=top;
        top=top->next;
        runqueue->next=NULL;
        runqueue->runtime= runqueue->runtime-timeslice;
        if(runqueue->runtime<=0)
```

```
        {
           runqueue->killtime=runqueue->runtime+timeslice;
           runqueue->runedtime=runqueue->runedtime+runqueue->killtime;
           runqueue->runtime=0;
           printf("process name %d, runtime=%d, runedtime=%d,killtime=%d\n"
           ,runqueue->name,runqueue->runtime,runqueue->runedtime,runqueue->killtime);
        }
        else{
           runqueue->killtime=timeslice;
           runqueue->runedtime=runqueue->runedtime+runqueue->killtime;
           printf("process name %d, runtime=%d, runedtime=%d,killtime=%d\n"
           ,runqueue->name,runqueue->runtime,runqueue->runedtime,runqueue->killtime);
           tail->next=runqueue;
           tail=tail->next;
        }
     }
}
```

6）运行结果，包括各个进程的运行顺序，每次占用处理机的运行时间，可以参考下列输出如图 4.3。

```
process name=1, run=8, runed=3,  killtime=3
process name=2, run=0, runed=3,  killtime=3
process name=3, run=13, runed=3,  killtime=3
process name=4, run=6, runed=3,  killtime=3
process name=5, run=0, runed=1,  killtime=1
process name=6, run=11, runed=3,  killtime=3
process name=7, run=4, runed=3,  killtime=3
process name=8, run=16, runed=3,  killtime=3
process name=9, run=9, runed=3,  killtime=3
process name=10, run=2, runed=3,  killtime=3
process name=1, run=5, runed=6,  killtime=3
process name=3, run=10, runed=6,  killtime=3
process name=4, run=3, runed=6,  killtime=3
process name=6, run=8, runed=6,  killtime=3
process name=7, run=1, runed=6,  killtime=3
process name=8, run=13, runed=6,  killtime=3
process name=9, run=6, runed=6,  killtime=3
process name=10, run=0, runed=5,  killtime=2
process name=1, run=2, runed=9,  killtime=3
process name=3, run=7, runed=9,  killtime=3
process name=4, run=0, runed=9,  killtime=3
process name=6, run=5, runed=9,  killtime=3
process name=7, run=0, runed=7,  killtime=1
process name=8, run=10, runed=9,  killtime=3
process name=9, run=3, runed=9,  killtime=3
process name=1, run=0, runed=11,  killtime=2
```

图 4.3 输出结果示意图

七、 实验步骤

1）进入 vi 编辑器

2）在编译器中输入所要运行的程序代码

3）退出编辑器，返回命令行输入方式，使用 gcc 编译器编译程序，获得能运行的目标程序。

4）运行目标程序，查看运行结果。

八、 注意事项

1）修改时间片大小，查看对实验结果的影响。

2）随机数的产生由 rand()函数实现，rand()的输出随机数范围在 0~2^{15} 之间，需要转换到 0~20 范围。

3）注意链表节点的插入，删除方法。

九、 实验报告要求

需要列出运行了的程序清单及相应结果，并对结果进行分析和讨论。对结果的分析主要讨论时间片大小对程序执行的影响？