# 实验__一__题目__进程调度___第___周星期_____

## 一、实验目的

用高级语言编写和调试一个进程调度程序，以加深对进程的概念及进程调度算法的理解。

## 二、实验内容和要求

编写并调试一个模拟的进程调度程序，采用“短进程优先”调度算法对五个进程进行调度。以加深对进程的概念及进程调度算法的理解.

每个进程有一个进程控制块（PCB）表示。进程控制块可以包含如下信息：进程名、优先数、到达时间、需要运行时间、已用 CPU 时间、进程状态等等。

进程的优先数及需要的运行时间可以事先人为地指定（也可以由随机数产生）。进程的到达时间为进程输入的时间。

进程的运行时间以时间片为单位进行计算。

每个进程的状态可以是就绪 W（Wait）、运行 R（Run）、或完成 F（Finish）三种状态之一。

就绪进程获得 CPU 后都只能运行一个时间片。用已占用 CPU 时间加 1 来表示。

如果运行一个时间片后，进程的已占用 CPU 时间已达到所需要的运行时间，则撤消该进程，如果运行一个时间片后进程的已占用 CPU 时间还未达所需要的运行时间，也就是进程还需要继续运行，此时应将进程的优先数减 1（即降低一级），然后把它插入就绪队列等待 CPU.

每进行一次调度程序都打印一次运行进程、就绪队列、以及各个进程的 PCB，以便进行检查。重复以上过程，直到所要进程都完成为止。

## 三、实验主要仪器设备和材料

实验环境

硬件环境：个人台式机 Microsoft Windows7

软件环境：C 语言编程环境，VS 2010

## 四、实验原理及设计方案

1、实验原理

进程作为基本数据处理单元，需要对进程的基本信息进行相关的描述。进程的基本

信息包括进程进程名、到达的时间、预计的进程运行时间、进程开始运行时间、进程仍需运行的时间、进程完成的时间、进程运行的次数等。在此，可以定义一个结构体链表来储存进程信息。并在此基础上进行其他操作

短进程优先调度算法　对强占式短进程优先调度算法而言，其本质特征便是按进程的预计运行时间长短进行排序，先执行短进程。 若内存中运行的进程优先级比就绪队列中的某进程优先级低（即运行的进程预计运行时间比就绪队列中的某进程长），此运行的进程让出内存并进入就绪队列，优先级更高的短进程强占内存资源并运行直到结束或者遇到优先级更高的进程强占为止。

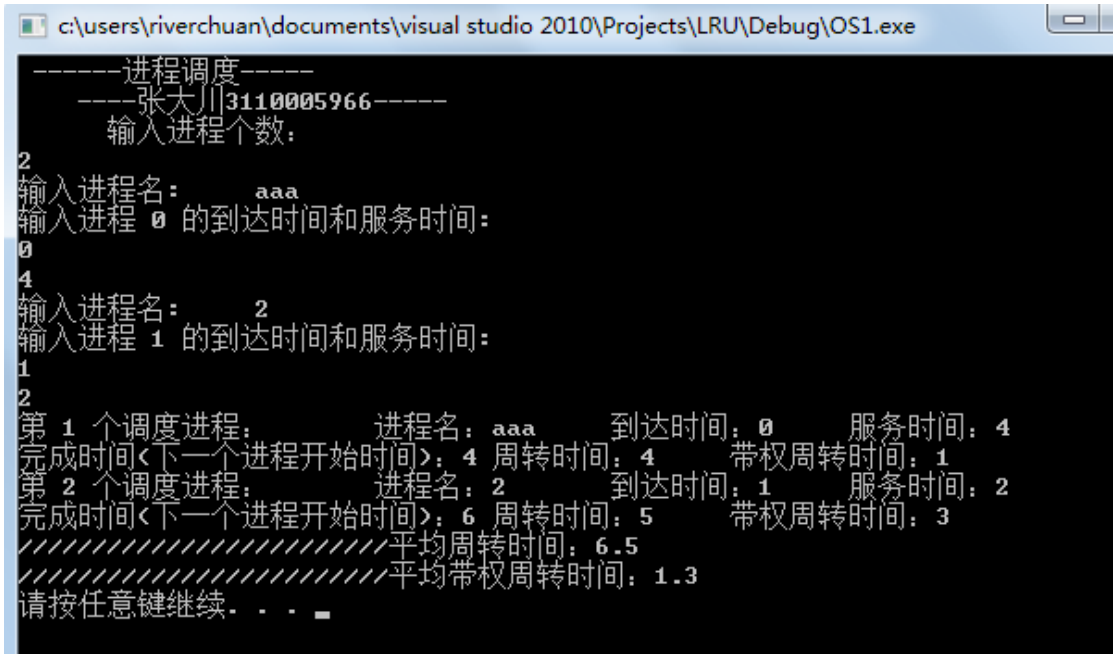
2、 设计方案

```
  struct pcb {
            char name[10];   //进程名
            char state;      //进程状态 w：等待 r：运行
            int queue;       //进程队列 1：2：3
            int priority;    // 数字越小优先级越高
            int needtime;    //需运行时间
            int runtime;     //已经运行时间
            pcb *link;       //进程队列下一指针
 }*ready=NULL,*run;
typedef struct pcb PCB;
//分别设置三条就绪队列头指针和尾指针
PCB *First=NULL,*Second=NULL,*Third=NULL,*end1=NULL,*end2=NULL,*end3=NULL;
//为三条就绪队列设置三个时间片：1：2：3
int time[3];
```

3、 程序流程图

**五、实验结果及分析**

短进程优先调度算法.exe,出现进程初始化界面，输入初始进程数目，接着必须按提示输入所有初始化进程，如下图所示：




**六、调试总结及心得体会**

第一次实验相对比较简单，依照教材的例子，照设计思想，不断调试不断完善，总算完成了，加深了进程调度的理解，搞清楚了各个调度的优缺点，收获不少。

**七、源程序代码：**

```
#include<iostream>
#include<string>
#include<stdio.h>
using namespace std;
//进程结构
class JCB
{
public:
    string name;
    int cometime;
    int sertime;
    bool run;
    JCB * next;
public:
    JCB(string tn="",int tc=-1,int ser=0,bool tr=false,JCB *nex=NULL)
        :name(tn),cometime(tc),sertime(ser),run(tr),next(nex)
    {}
    void show()
```

```
    {
        cout<<"\t 进 程 名 ： "<<name<<"\t 到 达 时 间 ： "<<cometime<<"\t 服 务 时 间 ：
"<<sertime<<endl;
    }
};

//进程的排序，按照进程的提交时间先后排序
void sort(JCB *head)
{
    JCB *q,*s,*p=head->next;
    head->next=NULL;
    while(p!=NULL)
    {
        q=p->next;
        if(head->next==NULL)
        {
            p->next=head->next;
            head->next=p;
        }
        else
        {
            s=head;
            while(p->cometime>s->next->cometime)
            {
                s=s->next;
            }
            p->next=s->next;
            s->next=p;
        }
        p=q;
    }
}
//辅助函数，查找JCB
JCB * getJCB(JCB *q,JCB *p)
{
    JCB *find=q;
    int minsertime=q->sertime;
    while(q!=p)
    {
        if(q->sertime<minsertime&&q->run==false)
        {
            minsertime=q->sertime;
            find=q;
        }
```

```
            q=q->next;
        }
        return find;
}
//shadule调度函数
//
void shadule(JCB *head,int n)
{
        JCB *s,*q,*p=head->next;
        JCB *sp;
        int sum_rolltime=p->sertime-p->cometime;
        int waittime=p->cometime;
        int sum_rollsertime=p->sertime;
        int i=1;
        sp=p;
        for(int num=0;num<n;num++)
        {
            while(p->run==true)
                p=p->next;
            q=p;
            while(p!=NULL&&p->cometime<=waittime)
                p=p->next;
            s=getJCB(q,p);
            cout<<"第 "<<i++<<" 个调度进程： ";
            s->show();
            s->run=true;
            waittime+=s->sertime;
            cout<<"完成时间（下一个进程开始时间）： "<<waittime<<"\t周转时间：
"<<waittime-(s->cometime)
                <<"\t带权周转时间： "<<waittime/float(s->sertime)<<endl;
            sum_rolltime+=(waittime-(s->cometime));
            sum_rollsertime+=s->sertime;
            p=sp;
        }
        cout<<"//////////////////////平均周转时间： "<<sum_rolltime/float(n)<<endl;
        cout<<"//////////////////////  平  均  带  权  周  转  时  间  :
"<<sum_rolltime/float(sum_rollsertime)<<endl;
        system("pause");
}



void main()
{
```

```
    int num;
    string name;
    int cometime,sertime;
    cout<<" ------进程调度-----"<<endl;
    cout<<"    ----张大川3110005966-----"<<endl;

    cout<<"      输入进程个数："<<endl;
    cin>>num;
    JCB *head=new JCB();

    for(int i=0;i<num;++i){
        cout<<"输入进程名:\t";
        cin>>name;
        cout<<"输入进程 "<<i<<" 的到达时间和服务时间:"<<endl;
        cin>>cometime>>sertime;
        JCB *p=new JCB(name,cometime,sertime,false,head->next);
        head->next=p;
    }
    sort(head);
    shadule(head,num);
}
```