

## 一. 实验题目

设计一个有 N 个进程并行的进程调度程序。

进程调度算法：采用最高优先数优先的调度算法（即把处理机分配给优先数最高的进程）和同优先级条件下先来先服务算法。

## 二. 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。

## 三. 实验步骤

步骤：

首先定义进程控制块 PCB：

```
struct pcb {
    char name[10];
    char state;//进程状态
    int super;//优先级
    int ntime;//需要时间
    int rtime;//已经执行时间
    struct pcb* link;
}*ready = NULL, *p;
```

建立对进程进行最高优先级调度算法，同级条件下进行先来先服务算法函数：

```
void sort() {
    PCB* temp;
    p->super -= 1;
    temp = ready;
    if (temp != NULL) {
        if (p->super > temp->super) { //插入队首
            p->link = ready;
            ready = p;
        }
    }
    else {
```

```

        while (temp != NULL) {
            if (temp->link == NULL) { //插入队尾
                temp->link = p;
                break;
            }
            if (p->super > temp->link->super) {
                p->link = temp->link;
                temp->link = p;
                break;
            }
            temp = temp->link;
        }
    }
}
else {
    ready = p;
}
}
}

```

建立进程控制块输入函数:

```

void input() {
    int i, num;
    printf("请输入即将运行的进程总数目: ");
    scanf("%d", &num);
    for (i = 0; i < num; i++) {
        p = (PCB*)malloc(sizeof(PCB)); //申请空间, 指针
        if (p == NULL) {
            printf("内存分配不成功! \n");
        }
        else {
            printf("\n 请输入第%d 个进程名称:", i+1 );
            scanf("%s", p->name, 10);
            printf("请输入第%d 个进程的优先级: ", i+1);
            scanf("%d", &p->super);
            printf("请输入第%d 个进程运行时间:", i+1 );
            scanf("%d", &p->ntime);
            p->rtime = 0;
            p->state = 'W';
            p->link = NULL;
            linkpcb();
        }
    }
}
}

```

建立函数查看进程:

```
void check() {
    PCB* temp;
    printf("\n **** 当前正在运行的进程是:%s\n", p->name); //显示当前运行进程
    disp(p);
    temp = ready;
    printf("\n ****当前就绪队列状态为:\n"); //显示就绪队列状态
    if (temp == NULL)printf("\n 就绪队列为空.\n ");
    while (temp != NULL) {
        disp(temp);
        temp = temp->link;
    }
    free(temp);
}
```

建立进程就绪函数(进程运行时间到, 置运行状态):

```
void running() {
    if (p->rtime == p->ntime) { //进程已完成
        printf("\n 进程 [%s] 已完成.\n", p->name);
    }
    else {
        sort(); //调用 sort 函数
    }
}
```

主函数:

```
int main() {
    int len, h = 0;
    input(); //输入进程
    system("cls"); //清屏
    len = PCB_Len();
    while ((len != 0) && (ready != NULL)) {
        printf("\n\nThe execute number:%d\n", ++h);
        p = ready;
        ready = p->link;
        p->link = NULL;
        p->state = 'R';
        check(); //查看进程
        p->rtime += 1; //运行时间加 1;
        p->state = 'W'; //置为就绪状态
        running();
    }
    printf("\n\n 进程已经完成.\n");
}
```

```
    system("Pause");  
    return 0;  
}
```

**测试用例：**

进程	P2	P1	P3	P4
优先级	3	2	4	1
运行时间	4	2	3	2

**运行结果：**

请输入即将运行的进程总数目: 4

请输入第1个进程名称:P2  
请输入第1个进程的优先级: 3  
请输入第1个进程运行时间:4

请输入第2个进程名称:P1  
请输入第2个进程的优先级: 2  
请输入第2个进程运行时间:2

请输入第3个进程名称:P3  
请输入第3个进程的优先级: 4  
请输入第3个进程运行时间:3

请输入第4个进程名称:P4  
请输入第4个进程的优先级: 1  
请输入第4个进程运行时间:2

The execute number:1

\*\*\*\* 当前正在运行的进程是:P2

Name	State	Super	NTime	RunTime
P2	R	3	4	0

\*\*\*\*当前就绪队列状态为:

Name	State	Super	NTime	RunTime
P1	W	2	2	0

Name	State	Super	NTime	RunTime
P3	W	4	3	0

Name	State	Super	NTime	RunTime
P4	W	1	2	0

The execute number:2

\*\*\*\* 当前正在运行的进程是:P1

Name	State	Super	NTime	RunTime
P1	R	2	2	0

\*\*\*\*当前就绪队列状态为:

Name	State	Super	NTime	RunTime
P3	W	4	3	0

Name	State	Super	NTime	RunTime
P2	W	2	4	1

Name	State	Super	NTime	RunTime
P4	W	1	2	0

The execute number:3

\*\*\*\* 当前正在运行的进程是:P3

Name	State	Super	NTime	RunTime
P3	R	4	3	0

\*\*\*\*当前就绪队列状态为:

Name	State	Super	NTime	RunTime
P2	W	2	4	1

Name	State	Super	NTime	RunTime
P4	W	1	2	0

Name	State	Super	NTime	RunTime
P1	W	1	2	1

The execute number:4

\*\*\*\* 当前正在运行的进程是:P3

Name	State	Super	NTime	RunTime
P3	R	3	3	1

\*\*\*\*当前就绪队列状态为:

Name	State	Super	NTime	RunTime
P2	W	2	4	1

Name	State	Super	NTime	RunTime
P4	W	1	2	0

Name	State	Super	NTime	RunTime
P1	W	1	2	1

The execute number:5

\*\*\*\* 当前正在运行的进程是:P2

Name	State	Super	NTime	RunTime
P2	R	2	4	1

\*\*\*\*当前就绪队列状态为:

Name	State	Super	NTime	RunTime
P3	W	2	3	2

Name	State	Super	NTime	RunTime
P4	W	1	2	0

Name	State	Super	NTime	RunTime
P1	W	1	2	1

The execute number:6

\*\*\*\* 当前正在运行的进程是:P3

Name	State	Super	NTime	RunTime
P3	R	2	3	2

\*\*\*\*当前就绪队列状态为:

Name	State	Super	NTime	RunTime
P4	W	1	2	0

Name	State	Super	NTime	RunTime
P1	W	1	2	1

Name	State	Super	NTime	RunTime
P2	W	1	4	2

进程 [P3] 已完成.

The execute number:7

\*\*\*\* 当前正在运行的进程是:P4

Name	State	Super	NTime	RunTime
P4	R	1	2	0

\*\*\*\*当前就绪队列状态为:

Name	State	Super	NTime	RunTime
P1	W	1	2	1

Name	State	Super	NTime	RunTime
P2	W	1	4	2



The execute number:8

\*\*\*\* 当前正在运行的进程是:P1

Name	State	Super	NTime	RunTime
P1	R	1	2	1

\*\*\*\*当前就绪队列状态为:

Name	State	Super	NTime	RunTime
P2	W	1	4	2
Name	State	Super	NTime	RunTime
P4	W	0	2	1

进程 [P1] 已完成.

The execute number:9

\*\*\*\* 当前正在运行的进程是:P2

Name	State	Super	NTime	RunTime
P2	R	1	4	2

\*\*\*\*当前就绪队列状态为:

Name	State	Super	NTime	RunTime
P4	W	0	2	1

The execute number:10

\*\*\*\* 当前正在运行的进程是:P4

Name	State	Super	NTime	RunTime
P4	R	0	2	1

\*\*\*\*当前就绪队列状态为:

Name	State	Super	NTime	RunTime
P2	W	0	4	3

进程 [P4] 已完成.

The execute number:11

\*\*\*\* 当前正在运行的进程是:P2

Name	State	Super	NTime	RunTime
P2	R	0	4	3

\*\*\*\*当前就绪队列状态为:

就绪队列为空。

进程 [P2] 已完成.

进程已经完成.  
请按任意键继续. . .