

武汉大学计算机学院

课程实验(设计)报告

专业(班): 计算机科学与技术 三班

学 号: 2013301500100

姓 名: 秦贤康

课程名称: 操作系统设计

任课教师: 宋伟

2015 年 11 月 26 日

说 明

1、本课程设计报告包括：实习题目、实习内容及设计思想（设计思路、主要数据结构、主要代码结构及代码段分析）、上机实习所用平台及相关软件、调试过程（测试数据设计、测试结果分析）、总结（实习中遇到的问题及解决方法、实习中产生的错误及原因分析、实习体会及收获）。

2、课程设计报告用 A4 纸打印。

标题 4 号宋体加粗，正文 5 号宋体，图表文字标注、程序等用 6 号。

页边距：左边 3.17cm，右边 3.17 cm，上 2.54 cm，下 2.54 cm。

行间距：1.5 倍行距，一页 30 行，每行 39 字。

页脚：1.5 cm，页脚(页码，居中)。

1	实习内容.....	4
2	实习题目及设计思想.....	4
	2.1 实习题目.....	4
	2.2 设计思想.....	4
	2.3 主要数据结构.....	4
	2.4 主要代码结构及代码段分析.....	5
3	上机实习所用平台及相关软件.....	6
4	调试过程.....	6
	4.1 测试数据设计.....	6
	4.2 测试结果分析.....	7
5	总结.....	8

处理器调度

1 实习内容

选择一个调度算法，实现处理器调度。

2 实习题目及设计思想

2.1 实习题目

第一题：设计一个按优先数调度算法实现处理器调度的程序。

[提示]：

(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——按优先数的大小把 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单位时间数。
- 优先数——赋予进程的优先数，调度时总是选取优先数大的进程先执行。
- 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) 每次运行之前，为每个进程任意确定它的“优先数”和“要求运行时间”。

(3) 处理器总是选择队首进程运行。采用动态改变优先数的办法，进程每运行 1 次，优先数减 1，要求运行时间减 1。

(4) 进程运行一次后，若要求运行时间不等于 0，则将它加入就绪队列，否则，将状态改为“结束”，退出队列。

(5) 若就绪队列为空，结束，否则转到(3)重复。

要求能接受键盘输入的进程优先数及要求运行时间，能显示每次进程调度的情况，如哪个进程在运行，哪些进程就绪，就绪进程的排列情况。

2.2 设计思想

- 把类的对象指针队列保存到类里面
- 每次某个进程运行之后，按队列中进程的优先级进行排序
- 每次运行队列中的第一个进程，直到运行结束。
- 进程结束后仍然保存在队列中用于显示，把优先级调成最低

2.3 主要数据结构

```
class PCB{
private:
    int name;                //进程 id
    int time_needed;         //需要运行时间
    char state;              //进程状态
    int priority_number;      //优先级
```

```

public:
    PCB(int tn, int pri);           //构造函数
    static int size;               //队列中进程数
    static list <PCB*> pcb_list;   //进程队列
    void disp();                   //显示当前进程信息
    static void run();             //运行进程队列
    bool prun();                   //运行当前进程
    static bool compare_pcb(const PCB* f,
                             const PCB* s); //优先级比较函数
    static void disp_list();       //显示所有队列信息
};

```

2.4 主要代码结构及代码段分析

主源代码:

main.cpp

```

int main(void)
{
    int num, tn, pri;
    //cout<<"请输入任务数:";
    cin>>num;
    PCB* pcb[num];
    while(num-->0)
    {
        //cout<<"\n 任务"<<num+1<<endl;
        //cout<<"请输入所需运行时间:"<<endl;
        cin>>tn;
        //cout<<"请输入优先级:"<<endl;
        cin>>pri;
        pcb[num] = new PCB(tn, pri);
    }
    PCB::run();
    return 0;
}

```

新建任务之后，只要直接通过类名调用 run 方法即可。

Pcb.cpp

```

bool PCB::prun()
{
    if(this->time_needed <=0 || this->state == 'E'){
        size--;
        this->state='E';
        this->priority_number = MIN;
        return false;
    }
    this->priority_number--;
    this->time_needed--;
    if(this->time_needed<=0) //当一个进程运行完毕

```

```

    {    this->state = 'E';                                //更改状态为 E
        //size--;                                         //队列中进程数减一
    }
    return true;
}

```

运行某个进程进行的操作

3 上机实习所用平台及相关软件

OS: mint Mint 17.2 raphaela
 Kernel: x86_64 Linux 3.16.0-38-generic
 CPU: AMD A8-4500M APU with Radeon HD Graphics @ 1.9GHz
 GPU: Gallium 0.4 on AMD ARUBA
 RAM: 1522MiB / 3337MiB
 Vim:用 doxygen 自动生成注释
 g++ 4.8.4

4 调试过程

```

Test.sh
#!/bin/sh
# 测试三条数据
cat test1|./pcb >test1.out
cat test2|./pcb >test2.out
cat test3|./pcb >test3.out

```

运行 **test.sh** 自动进行输入输出，只要事先把测试数据写到 **test** 文件中就可以了

4.1 测试数据设计

Test1 文件中的内容

```

3
2 3
4 5
1 2

```

正常输入

Test2 文件中的内容

```

2
1 0
0 5

```

含有运行时和优先级为 0 结束任务的错误输入

Test3 文件中的内容

```

6
0 1
1 1
2 2

```

3 5

4 3

2 4

以上两种情况

4.2 测试结果分析

内容有点多，只以 test1.out 为例说明

name		time_needed	pri_num	state

Q1	2	3	R	
Q2	4	5	R	
Q3	1	2	R	

task_num:3 run:Q2 step:1

name		time_needed	pri_num	state

Q2	3	4	R	
Q1	2	3	R	
Q3	1	2	R	

task_num:3 run:Q2 step:2

name		time_needed	pri_num	state

Q2	2	3	R	
Q1	2	3	R	
Q3	1	2	R	

task_num:3 run:Q2 step:3

name		time_needed	pri_num	state

Q2	1	2	R	
Q1	2	3	R	
Q3	1	2	R	

task_num:3 run:Q1 step:4

name		time_needed	pri_num	state

Q1	1	2	R	
Q2	1	2	R	

Q3	1	2	R
----	---	---	---

task_num:3 run:Q1 step:5

name	time_needed	pri_num	state	
Q1	0	0	E	
Q2	1	2	R	
Q3	1	2	R	

task_num:3 run:Q2 step:6

name	time_needed	pri_num	state	
Q2	0	0	E	
Q3	1	2	R	
Q1	0	0	E	

task_num:3 run:Q3 step:7

name	time_needed	pri_num	state	
Q3	0	0	E	
Q2	0	0	E	
Q1	0	0	E	

从结果上来看，一共运行了7次，与运行时 $2+4+1=7$ 符合，同时，运行顺序为 2 2 2 1 1 2 3，符合按照优先级调度的结果，运行之后三个进程的所需要运行时间都为0,状态为E。

5 总结

这次实验，让我回去好好地把操作系统的书又找出来看了一下，发现自己好多东西学了就忘了。通过实验能加深自己对操作系统中进程调度的理解。

遇到的问题及解决方法：

- 1、类中 static 变量必须初始化
- 2、代码没问题，编译时不通过

解决方法：把 Makefile 中编译器改成 g++

- 3、mint 中默认为 dash，与 bash 的 shell 语法有点不一样
- 4、进程运行结束后，仍然有状态为 R 的进程

解决方法：代码写错，没有考虑到用户会输入运行时非正数的情况，加上判断