第三章　实验三

模拟时间片轮转调度算法

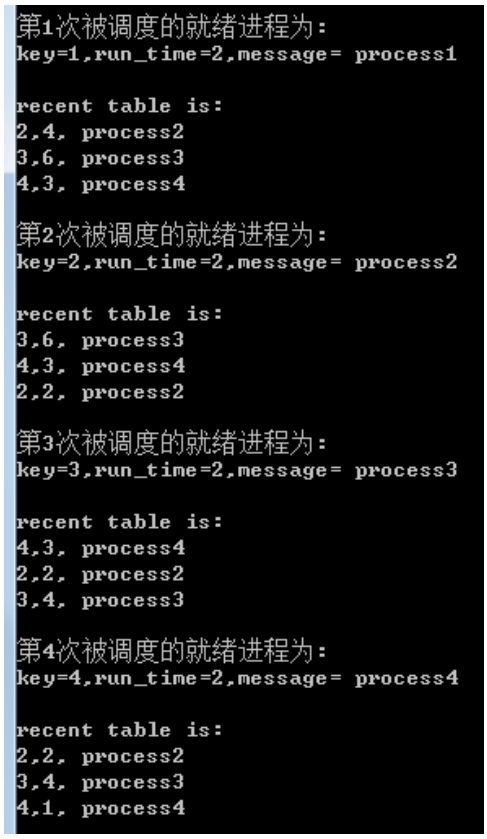
按照时间片轮转调度进程

- 动态地输入进程(key,run_time,message)，按照输入次序建立就绪队列
- 输入 CPU 运行的单位时间片(cpu_base_time)
- 按照时间片轮转方式模拟进程逐个被调度并执行单位时间片(运行结束进程结束，否则修改进程运行时间 run_time,将该进程放置在就绪队列尾巴)。

例如：

输入多个进程

输入时间片



输出结果

- 使用 Micrsoft Visual Studio C++ 6.0 或 CodeBlocks 编程：程序 3_4_timeslice.cpp。完善如下程序代码：

```
#include <malloc.h>
#include <stdio.h>
#include <string.h>
#define NULL 0
typedef struct table
 { int key;                    /*进程 ID 号*/
    int run_time;              /*进程运行时间*/
    char message[10];          /*进程说明信息*/
    struct table *next;
 }node;
node *creat(void)          /*定义函数，输入 ID 号和顺序号，按照输入次序建立进程链表*/
  { 填补程序 }
void print (node *head)    /*输出链表*/
  { 填补程序 }
node *insert(node *head,node *news)     /*将进程 news 插入到队列尾部*/
  { 填补程序 }
node *timeslice(node *head,int cpu_base_time)
                    /*模拟时间片轮转调度过程:队列首进程使用一个时间片的 CPU*/
  { 填补程序 }
void main()
  { int count=0,cpu_base_time;
    node *p;
    printf("新建的进程控制表为:\nkey run_time message\n");
    p=creat();          /*输入进程控制表*/
    print(p);           /*输出原始进程控制表*/
    printf("CPU 运行的单位时间片 cpu_base_time 为:\n");
    scanf("%d",&cpu_base_time);
    while(p)            /*模拟按单位时间片进程逐个被调度并进入 CPU 运行的过程*/
     { 填补程序 }
  }
```