

第三章 实验一

模拟先来先服务（FCFS）调度算法

按照 FCFS 调度进程

- 动态地输入进程(key,sequence,message)，按照输入次序建立队列
- 按照 FCFS 的次序模拟进程逐个被调度并进入 CPU 运行的过程

例如：

```

The table is:
22,1, process22
30,2, process30
13,3, process13
90,4, process90

第1次被调度的就绪进程为:
key=22,sequence=1,message= process22

第2次被调度的就绪进程为:
key=30,sequence=2,message= process30

第3次被调度的就绪进程为:
key=13,sequence=3,message= process13

第4次被调度的就绪进程为:
key=90,sequence=4,message= process90
Press any key to continue

新建的进程控制表为:
key sequence message
22 1 process22
30 2 process30
13 3 process13
90 4 process90
0
```

输入

输出结果

- 使用 Microsoft Visual Studio C++ 6.0 或 CodeBlocks 编程：程序 3_1_fcfs.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 sequence; /*进程进入队列顺序号*/
    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 *fcfs(node *head) /*模拟就绪进程队列中最先进入进程(队首进程)出队*/
{
    填补程序
}
void main()
void main()
{
    int count=0;
    node *p,*q;
```

```
printf("新建的进程控制表为:\nkey sequence message\n");
p=creat(); /*输入进程控制表*/
print(p); /*输出原始进程控制表*/
while(p) /*模拟进程逐个被调度并进入 CPU 运行的过程*/
{ 填补程序 }
}
```