南京林业大学



操作系统课程实习

任务书

# 实验一 进程调度

**一．实验目的**

加深对进程调度的理解，熟悉进程调度的不同算法，比较其优劣性。

**二．实验内容**

假如一个系统中有5个进程，它们的到达时间内如表1所示，忽略I/O以及其他开销时间。若分别按**抢占的短作业优先（SJF）**、**时间片轮转（RR，时间片=1）**进行CPU调度，请按照上述2个算法，编程计算出各进程的完成时间内、周转时间、带权周转周期、平均周转周期和平均带权周转时间。

表1 进程到达和需服务时间

|  |  |  |
| --- | --- | --- |
| 进程 | 到达时间 | 服务时间 |
| A | 0 | 3 |
| B | 2 | 6 |
| C | 4 | 4 |
| D | 6 | 5 |
| E | 8 | 2 |

**实验二 进程的同步与互斥**

**一．设计目的**

分析进程争用资源的现象，学习解决进程互斥的方法。

**二．设计内容**

用程序实现生产者—消费者问题。具体问题描述：

   一个仓库可以存放K件物品。生产者每生产一件产品，将产品放入仓库，仓库满了就停止生产。消费者每次从仓库中去一件物品，然后进行消费，仓库空时就停止消费。

数据结构：

* Producer - 生产者进程，Consumer - 消费者进程
* buffer: array [0..k-1] of integer;
* in, out: 0..k-1; in记录第一个空缓冲区，out记录第一个不空的缓冲区
* s1,s2,mutex: semaphore; s1控制缓冲区不满,s2控制缓冲区不空,mutex保护临界区；
* 初始化s1=k,s2=0,mutex=1

原语描述：

producer（生产者进程）：

   item\_Type item;

  {

     while (true)

     {

       produce(&item);

       p(s1);

       p(mutex);

       buffer[in]:=item;

       in:=(in+1) mod k;

       v(mutex);

       v(s2);

     }

  }

  consumer（消费者进程）：

   item\_Type item;

  {

     while (true)

     {

       p(s2);

       p(mutex);

        item:=buffer[out];

        out:=(out+1) mod k;

        v(mutex);

        v(s1);

     }

   }

# 实验三 存储管理

**一．设计目的**

通过请求页面式存储管理中页面置换算法设计，了解存储技术的特点，掌握请求页式存储管理的页面置换算法。

**二．设计内容**

用程序实现生产者——消费者问题，将指令序列转换为用户虚存中的请求调用页面流。

**具体要求：**

* 页面大小为1K
* 用户内存容量为4页到40页
* 用户外存的容量为40k

在用户外存中，按每K存放10条指令，400条指令在外存中的存放方式为：

* 0-9条指令为第0页
* 10-19条指令为第1页

......

* 390-399条指令为第39页

按以上方式，用户指令可组成40页，通过随机数产生一个指令序列，共400个指令（0-399）。模拟**请求页式存储管理**中页面置换算法，执行一条指令，首先在外存中查找所对应的页面和页面号，然后将此页面调入内存中，模拟并计算下列三种算法在不同内存容量下的命中率(页面有效次数/页面流的个数):

1. 最久未使用算法(LRU)
2. 改进的Clock置换算法

**提示**

* 随机指令的产生 ：rand() 或srand()
* 用户内存中页面控制结构采用链表

struct p\_str{

int pagenum; /\* 页号 \*/

int count; /\* 访问页面的次数 \*/

struct p\_str next; /\* 下一指针 \*/

}p\_str;