

# 简单题

## 零散

文件系统的数据一致性检查 = 块 + 文件的 ...

OS中进程调度不需要硬件支持

动态重定向是在运行(执行)期间进行的

中断向量指 中断处理程序入口地址

OS两个接口 - 命令行接口 + 程序接口

批处理首先考虑周转时间和系统吞吐量

响应比高者优先

短作业 等待时间(最长) 考虑要素

## 文件管理

如果文件采用直接存取方式, 且文件大小不固定, 则采用(索引)结构较合适

目录管理的要求, 首先是要能够按名存取, 其次是提高对目录的查找速度

已知请求分页系统中, 处理器访问一次内存的时间为 $t$ , 访问一次快表的时间为 $v$ , 缺页中断处理时间为 $w$ , 若页面不在内存, 则处理器访问一次逻辑地址所需时间为  $2(t+v)+w$

顺序结构不利于文件长度动态增长

## 进程同步

若进程P一旦被唤醒（完成I/O操作）就能够投入运行，系统可能为（D）。

- A. 在分时系统中，进程P的优先级最高
- B. 抢占调度方式，就绪队列上的所有进程的优先级均比P的低
- C. 就绪队列为空队列
- D. 抢占调度方式，P的优先级高于当前运行的进程

两种可能：

- 1. 就绪队列为空且CPU空闲；
- 2. 抢占式调度，P的优先级高于当前运行进程和就绪队列上的所有进程

某系统中有11台打印机，n个进程共享打印机资源，每个进程要求获得3台打印机，当n的取值不超过（5）时，系统不会发生死锁。

给每个进程分配2台，所以当 $n=5$ ， $2 \times 5=10$ ，还剩一台打印机可以轮流使用，系统安全

某计算机系统中有8台打印机，有K个进程竞争使用，每个进程最多需要3台打印机。该系统可能会发生死锁的K的最小值是（C）

- A. 2 B. 3 C. 4 D. 5

$2 \times 4=8$ ，如果每个进程分配两台打印机产生死锁

## 简答题

## 计算题

## 进程同步专题

### 思路

[整理自CSDN](#)

关系分析。找出题目中描述的各个进程，分析它们之间的同步、互斥关系。

整理思路。根据各进程的操作流程确定P、V操作的大致顺序。

设置信号量。并根据题目条件确定信号量初值。（互斥信号量初值一般为1，同步信号量的初始值要看对应资源的初始值是多少）

```
semaphore mutex = 1; // 互斥信号量，实现对缓冲区的互斥访问
semaphore empty = n; // 同步信号量，表示空闲缓冲区的数量
semaphore full = 0; // 同步信号量，表示产品的数量，也即非空缓冲区的数量
```

这里作了代码层次优化

注意：

- 互斥信号量加不加，加上肯定没错
- 实现互斥的P操作一定要在实现同步的P操作之后，否则可能导致死锁
- V操作不会导致进程阻塞，因此两个V操作顺序可以交换。
- 分析问题应该以**事件角度考虑**，而不是以进程角度考虑

## 打井水

某寺庙有大、小和尚若干，有一水缸。由小和尚挑水入缸供老、大和尚饮用。水缸可容10桶水，水取自同一井。水井很窄，每次只能容一个水桶取水。水桶总数为3个。每次入、取缸水仅为1桶，且不可同时进行。试给出取水、入水的同步算法

```
semaphore mutex1,mutex2,count,full,empty;
```

```
mutex1.value=1,mutex2.value=1,  
count.value=3,  
full.value=0,empty.value=10;
```

```
process 小和尚Pi(i=1,2,3...)
```

```
{  
    while(true)  
    {  
        wait(empty);  
        wait(count);  
  
        wait(mutex1);  
        从井中取水  
        signal(mutex1);  
  
        wait(mutex2);  
        把水放入缸中  
        signal(mutex2);  
  
        signal(count);  
        signal(full);  
    }  
}
```

```
process 老和尚Pj(j=1,2,3...)
```

```
{  
    while(true)  
    {  
        wait(full);  
        wait(count);
```

```

        wait(mutex2);
        从缸中取水
        signal(mutex2);

        signal(count);
        signal(empty);
    }
}

main()

{
    cobegin

        小和尚P1();
        老和尚P1();
        小和尚P2();
        老和尚P2();
        .....

    coend;
}

```

## 过桥

一条河上架设了由若干个桥墩组成的一座桥。若一个桥墩只能站一个人，过河的人只能沿着桥向前走而不能向后退。过河时，只要对岸无人过，就可以过。但不允许河对岸的两个人同时过，以防止出现死锁。请给出两个方向的人顺利过河的同步算法。

由于每一个桥墩可以站一个人，同一方向可以走多个人，但不能两个方向的人同时过，所以对于每一个方向，首先判断这个方向上的人数是不是0，如果是就去申请mutex，不是的话，说明桥目前已经是这个方向的“主场”了，直接过桥就可以了，在判断人数之前首先要设置一个互斥量比如SA，因为如果不设置，就可能同时countA+=2，桥墩就站了两个人，过完桥后，只有这个方向上的人全走过去了才释放mutex。

```

int countA=0,countB=0;
semaphore SA,SB,mutex;
SA.value=1,SB.value=1,mutex.value=1;

process Ai()
{
    wait(SA);

    if(countA==0) wait(mutex);
    countA++;

    signal(SA);
    过桥
    wait(SA);

    if(countA==0) signal(mutex);
    countA--;

    signal(SA);
}

process Bi()
{
    wait(SB);

    if(countB==0)wait(mutex);
    countB++;

    signal(SB);
    过桥
    wait(SB);

    if(countB==0)signal(mutex);
    countB--;

    signal(SB);
}

main()
{
    Cobegin
    A1(),A2(),.....
    B1(),B2(),.....
    CoEnd
}

```

## 进程调度

使用自创多级流水线示意图法清晰快速把握题干

这样就一目了然了

```
P1 [|||| ]
P2  [|||| ]
P3  [|||||]
```

## FCB存储目录大小计算

## 页式块式大小判断

例：考虑一个由8个页面，每页有1024个字节组成的逻辑空间，把它装入到有32个物理块的存储器中，问：

- (1) 逻辑地址需要多少二进制位表示？
- (2) 物理地址需要多少二进制位表示？

在分页存储管理中，逻辑地址结构 = ...

它由两个部分组成：前一部分表示该地址所在页面的页号 $p$ ；后一部分表示页内地址（页内位移） $d$ 。页号的地址位数决定了页的多少，假设页号有20位，则地址空间中最多可容纳的页面数为220，即1MB个页面。页内地址位数确定了每页的大小，若页内地址为12位，则每页大小为212，即2KB。

- 同理，物理地址中块号的地址位数决定了块的数量。由于页式存储管理内存空间块的大小与页面大小相同，所以物理地址中块内地址与逻辑地址中的页内地址位数相同。

- 解因为页面数为8=23，故需要3位二进制数表示。每页有1024个字节，1024=210，于是页内地址需要10位二进制数表示。32个物理块，需要5位二进制数表示（32=25）。

（1）页的逻辑地址由页号和页内地址组成，所以需要3+10=13位二进制数表示。

（2）页的物理地址由块号和页内地址的拼接，所以需要5+10=15位二进制数