

# 山东大学 2016——2017 操作系统考题

## 一：概念解释

1. 抢先式调度(剥夺式调度 抢占式调度)
2. RAID
3. SPOOLING
4. 虚拟机
5. 虚拟文件系统

## 二：简答题

1. 什么是颠簸（抖动 thrashing）？说明采用工作集模型预防系统抖动的思想与过程。
2. 请图示说明页式存储管理系统中对页面共享的方法，并说明共享代码和共享数据有何限制条件。
3. 设有四个进程，到达就绪队列时间及执行时间如下，若采用剥夺式（抢先式）最短作业优先调度，geichu8 各进程的调度次序及平均等待时间，给出计算过程。

进程到达及执行时间：

进程	到达就绪队列时间	执行时间
P1	0	6
P2	1	8
P3	2	3
P4	3	12

4. 在文件系统的管理中，当用户访问某文件时，需要给出要访问的文件名，系统根据该文件对应的文件控制块 FCB 确定该文件在磁盘上的物理位置，然后对文件内容进行存取。
  - （1）请说明文件系统如何依据用户给出的文件名找到该文件对应的 FCB；
  - （2）在连续、链式、索引三种文件的磁盘块组织方式中，分别说明 FCB 如何给出文件在磁盘上的物理位置。

## 三：回答下列问题

1. 有两个并发进程 P1 和 P2，他们都要使用临界资源 A 和 B，为了实现对着两种资源的互斥访问，定义了两个信号量 mutexA 和 mutexB 分别对应资源 A 和资源 B，信号量的定义及进程 P1 和 P2 的程序如下。

Semaphore mutexA=1,mutexB=1;

<b>Process P1</b> { wait(mutexA); wait(mutexB); 使用资源 A 和 B; signal(mutexA); signal(mutexB); }	<b>Process P2</b> { wait(mutexA); wait(mutexB); 使用资源 A 和 B; signal(mutexB); signal(mutexA); }
--------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------

请回答以下问题：

- (1) 上述两个并发进程 P1 和 P2 会不会存在死锁的可能？说明理由。
  - (2) 如果可能产生死锁，请修改上述程序，使得不会产生死锁，说明你的修改依据。
2. 计算机系统为保证操作系统及应用程序的可靠真确执行，采取了很多保护措施，阐述你所了解的相关技术。（提示：从硬件的执行模式、进程对临界区的并发访问、存储管理、文件管理、设备管理等内容涉及的保护技术进行阐述）
3. 有一个活动场地最多可容纳 22 名同学参加活动，其中一部分同学参与打篮球活动，不妨设为活动 A，另一部分同学参与大羽毛球活动，不妨设为活动 B。规定如下：
  - (1) 若活动场地中同学人数已经超过 22 人，则申请进入活动场地的同学等待。
  - (2) 参与 A、B 类活动的同学人数之差不能超过 5 人，即若参与活动 A 的人数比参与活动 B 的人数多 5 人，则申请参与活动 A 的同学等待；若参与过的 B 的人数比参与 A 的人数多 5 人，则申请参与活动 B 的同学等待。
  - (3) 参与 A、B 活动的同学可以随时离开。

请用“参与 A”、“参与 B”、“离开 A”、“离开 B”及信号量机制描述同学进入活动场地的过程。

- 4.某进程创建了一个子进程，子进程中又创建了两个线程，程序的大致结构如下：

```
void *Runner1(void *param);
void *Runner2(void *param);
Int main(int argc;char *argv[])
{
    pthread_t tid1,tid2;
    pthread_attr_t attr1,attr2;
    int value=5;
    int pid=fork();
    If(pid=0)
```

```

{
    pthread_attr_init(&attr1);
    pthread_create(&tid1,&attr1,Runner1,NULL);
    pthread_join(tid1,NULL);
    printf("\nCHILD:Runner1:value=%d\n",value);    //输出 1

    pthread_attr_init(&attr1);
    pthread_create(&tid2,&attr2,Runner2,NULL);
    pthread_join(tid2,NULL);
    printf("\nCHILD:Runner2:value=%d\n",value);    //输出 2
}
else
if(pid>0)
{
    wait(NULL);
    printf("parent:value=%d\n",value);    //输出 3
}
void *Runner1(void *param)
{
    value=value+3;
    pthread_exit(0);
}
void *Runner2(void *param)
{
    value=3;
    pthread_exit(0);
}
}

```

请问上述代码执行时，三条输出语句（输出 1、输出 2、输出 3）分别输出的结果是什么，说明你的理由。

Written By 龙堇