

## E02: 并发进程 (UNIX 进程与中断)

### 一、单项选择题

1. 响应外部中断的过程中, 下列不是中断隐指令完成的工作的一项\_\_\_\_\_。  
A. 关中断  
B. 保存通用寄存器的内容  
C. 形成中断入口程序地址并送程序计数器  
D. 保护断点外
2. 下列选项中, 会导致进程从用户态切换到核心态的操作是\_\_\_\_\_。  
I. 整数除以零                      II. `sin()`函数调用                      III. IO 操作结束  
A. I, II  
B. I, III  
C. II, III  
D. I, II 和 III
3. 下列选项中, 在用户态执行的是\_\_\_\_\_。  
A. 命令解释程序  
B. 缺页中断处理程序  
C. 进程调度程序  
D. 时钟中断处理程序
4. UNIX V6++中, 决定代码段是否与进程图象的可交换部分一起被换出到盘交换区的因素是\_\_\_\_\_。  
A. 内存空间的大小  
B. 内存中共享该代码段的进程数量  
C. 盘交换区的大小  
D. 进程图象可交换部分的大小
5. 若共享某一代码段的进程共有 5 个, 其中有三个进程图象的可交换部分在内存, 其余 2 个进程图象的可交换部分在盘交换区, 则该代码段的 Text 结构中, `x_count` 和 `x_ccount` 的值分别为\_\_\_\_\_。  
A. 5, 3  
B. 5, 2  
C. 5, 0  
D. 5, 5

### 二、简答题:

6. UNIX V6++中, 如果已知一个进程的 ID 号, 如何获得该进程图象可交换部分和代码段的起始地址? 如果当前进程图象的可交换部分在内存, 且需要换到盘交换区, 如何确定是否需要将代码段同时换出?
7. 多道计算机系统中为什么要有中断?
8. UNIX 中断返回时, 如果先前态为核心态则不进行例行调度, 这反映了什么思想?

### 三、应用分析题

9. 如果某一程序执行如下代码:

```
=====
.....
int main()
{
    int i, j;
    for (i=0; i<M; i++)
```

```
        for (j=0;j<M;j++)
            produce(i , j);
        ...;
    }
    void produce ( int row, int column )
    {
        int i;
        for (i=0;i<M;i++)
            matrixDes[row][column] += matrixOriginal[row][i] * matrixOriginal[i][column];
    }
}
=====
```

假设汇编后形成的机器指令为 1K，DATA 段长度为 0，BSS 段长度为 64 字节，

- (1) 请绘制上述代码汇编后形成的可执行文件的结构；
- (2) 分别创建进程 **pa**，**pb**，**pc** 执行上述可执行文件，其中，**pa**，**pb** 在内存，**pc** 在盘交换区上，请绘制所有进程的图象。
- (3) 当 **pa** 执行到 **produce** 函数的 **for** 语句时，请绘制该进程的用户栈和核心栈的构成（图中需标出 **main** 函数中的 **i** 和 **j**，**produce** 函数中的 **row** 和 **column**，**i** 和 **j** 的位置）。
- (4) 如果此时中断控制器发来一个中断请求，且 CPU 开中断，请绘制进程 **pa** 响应中断后，核心栈的变化情况。