

2008《操作系统》考试试卷(A)

一、选择题(若有多个符合,选最好的一个,每空1分,共20分)

1. 在操作系统中,并发性是指若干事件 _____ 发生。
A. 在同一时刻 B. 一定在不同时刻
C. 依次在不同时间间隔内 D. 在某一时间间隔内
2. 分时系统中为了使多个用户能同时与系统交互,关键的问题是 _____。
A. 计算机具有足够快的CPU B. 及时接收和处理多个用户的输入
C. 内外存间的信息交换足够快 D. 短时间内所有用户程序都能运行
3. 下列进程状态变化中, _____ 变化是不可能发生的。
A. 运行→就绪 B. 运行→等待
C. 等待→运行 D. 等待→就绪
4. 一个进程包含多个线程,下面那一项不是线程独立拥有的资源 _____。
A. 地址空间 B. 线程控制块 C. 运行栈 D. 执行状态
5. P、V操作是 _____。
A. 两条低级进程通信原语 B. 两组不同的机器指令
C. 两条系统调用命令 D. 两条高级进程通信原语
6. 下述哪个选项不是管程的组成部分 _____。
A. 局部于管程的共享数据结构
B. 对管程内数据结构进行操作的一组过程
C. 管程外过程调用管程内数据结构的说明
D. 对局部于管程的数据结构设置初值的语句
7. 设 m 为同类资源数, n 为系统中并发进程数。当 n 个进程共享 m 个互斥资源时,每个进程的最大需求是 w ,则下列情况会出现死锁的是 _____。
A. $m=2, n=1, w=2$ B. $m=2, n=2, w=1$
C. $m=4, n=3, w=2$ D. $m=4, n=2, w=3$
8. 下列调度算法中,不是作业调度的算法有 _____。
A. 先来先服务 B. 时间片轮转
C. 优先权 D. 响应比高者优先
9. 动态重定位是在作业的 _____ 中进行的。
A. 编译过程 B. 装入过程 C. 连接过程 D. 执行过程
10. 外部碎片出现在 _____。
A. 固定分区分配 B. 分页存储管理
C. 动态分区分配 D. 段页式存储管理
11. 实现虚拟存储器的目的是 _____。
A. 实现存储保护 B. 实现地址映射
C. 扩充辅存容量 D. 扩充主存容量
12. 在请求分页系统中, _____ 页面置换算法会产生异常现象。
A. 先进先出 B. 最近最久未使用 C. 最佳 D. 页面缓冲
13. 按 _____ 可以将设备分为字符设备和块设备。
A. 从属关系 B. 操作特性 C. 共享属性 D. 信息交换单位
14. 下述数据结构中, _____ 不是设备管理的数据结构。
A. FAT B. DCT C. SDT D. CHCT
15. 下述 _____ 磁盘调度算法可能导致某些进程发生饥饿现象。

- A. 先来先服务 B. 最短寻道时间优先 C. 扫描 D. 循环扫描
16. 若利用 10 行 20 列的位示图来标志盘块的分配现状，盘块号 143 对应的位示图的行列号为 ____。（注意行、列号均从 0 开始，盘块号也从 0 开始）
- A. 7 行 2 列 B. 7 行 3 列 C. 7 行 4 列 D. 7 行 5 列
17. 存放在磁盘上的文件 ____。
- A. 既可随机访问，又可顺序访问 B. 只能随机访问
C. 只能顺序访问 D. 不能随机访问
18. 操作系统提供给程序员的接口是 ____。
- A. 键盘命令 B. 系统调用 C. 汇编指令 D. 标准函数
19. 银行家算法在解决死锁问题中是用于 ____。
- A. 预防死锁 B. 避免死锁 C. 检测死锁 D. 解除死锁
20. 动态分区分配中，回收空闲区时，造成空闲区数目减 1 的情况是 ____。
- A. 上邻接空闲区 B. 下邻接空闲区
C. 上下邻接空闲区 D. 不邻接空闲区

二、填空题（每空 1 分，共 20 分）

1. 进程的特征是：动态性、并发性、独立性、____、____。
2. 程序顺序执行的特征有：顺序性、____、____。
3. 进程的高级通信方式有：____、____、____。
4. 按层次可以将处理机调度分为：____、____、____。
5. 动态分区分配算法有：首次适应算法、____、____、____。
6. 段页式系统中，要想访问信息需要____次访问主存，其中第二访问____。
7. 根据系统设置的缓冲个数，可以将缓冲技术分为：单缓冲、____、____、____。
8. 文件的逻辑结构分为：____、____。

三、判断正误，若错误请改正（每小题 2 分，共 10 分）

1. 用户程序执行时机器处于核心态。
2. 互斥使用及循环等待是死锁产生的原因。
3. 虚存系统中频繁的页面置换现象称为抖动。
4. 通道分为与 CPU 的接口、与设备的接口以及 I/O 处理逻辑三种类型。
5. FAT 格式的文件系统中，外存分配采用的是索引分配方式。

四、问答题（每小题 5 分，共 10 分）

1. 什么是操作系统？操作系统包含哪几类资源管理功能？
2. 什么是设备独立性？常见的输入/输出控制方式有哪几种？

五、设有四道作业，它们的到达时间和计算时间如下表所示：

| 作业 | 到达时间 | 计算时间 |
|----|------|-------|
| 1 | 8:00 | 70 分钟 |
| 2 | 8:20 | 40 分钟 |
| 3 | 9:00 | 10 分钟 |
| 4 | 9:10 | 30 分钟 |

若这四道作业在一台处理机上按单道方式运行，(1) 计算采用先来先服务调度算法时，作业的平均周转时间和平均带权周转时间。(2) 计算采用短作业优先调度算法时，作业的平均周转时间和平均带权周转时间。(10 分)

六、假定系统中有五个进程 P0、P1、P2、P3、P4 和三种类型的资源 R1、R2、R3，在 T0 时刻的资源分配情况如下表所示：

| 资源情况 进程 | Max | | | Allocation | | | Available | | |
|------------|-----|----|----|------------|----|----|-----------|----|----|
| | R1 | R2 | R3 | R1 | R2 | R3 | R1 | R2 | R3 |
| P0 | 7 | 7 | 3 | 0 | 2 | 0 | 3 | 3 | 1 |
| P1 | 3 | 3 | 2 | 2 | 1 | 0 | | | |
| P2 | 9 | 1 | 2 | 3 | 0 | 2 | | | |
| P3 | 2 | 3 | 3 | 2 | 1 | 2 | | | |
| P4 | 4 | 3 | 4 | 0 | 1 | 2 | | | |

试问：T0 时刻是否安全？若 P2 发出请求向量 $Request_2(4, 1, 0)$ ，系统能否将资源分配给它？（要求写出安全性检测过程，10 分）

七、有一请求分页存储管理系统，页面大小为每页 100 字节。有一个 50×50 的整型数组按列连续存放，每个整数占两个字节，将数组初始化为 0 的程序描述如下：

```
int a[50][50];
int i, j;
for (i=0; i<=49; i++)
    for (j=0; j<=49; j++)
        a[i][j] =0;
```

若在程序执行时内存中只有一个存储块用来存放数组信息，试问该程序执行时产生多少次缺页中断？（10 分）

八、多个进程共享一个文件，其中只读文件的称为读者，只写文件的称为写者。读者可以同时读，但写者只能独立写。试 (1) 用 P、V 操作写出其同步算法。(2) 修改上述的同步算法，使得它对写者优先，即一旦有写者到达，后续的读者必须等待。(10 分)

操作系统 2008A 参考答案

一、选择题（每小题 1 分，共 20 分）

1.D 2.B 3.C 4.A 5.A 6.C 7.D 8.B 9.D 10.C
11..D 12.A 13.D 14.A 15.B 16.B 17.A 18.B 19.B 20.C

二、填空题（每空 1 分，共 20 分）

- 1.制约性（异步性）、结构性
2. 封闭性、可再现性
- 3.共享存储器、消息系统、管道（共享文件）
- 4.作业调度（高级调度、长程调度、宏观调度）、中级调度（交换调度、中程调度）、进程调度（低级调度、短程调度、微观调度）
- 5.循环首次适应算法、最佳适应算法、最坏适应算法
6. 3、页表
- 7.双缓冲、循环缓冲、缓冲池
- 8.记录式文件、流式文件

三、判断改错（每小题 2 分，若错误改正 1 分）

- 1.错，用户程序执行时机器处于用户态。
- 2.错，互斥使用及循环等待是死锁产生的必要条件
- 3.对。
- 4.错，设备控制器由与 CPU 的接口、与设备的接口以及 I/O 处理逻辑三部分组成。
- 5.错，FAT 格式的文件系统中，外存分配采用的是链接分配方式。

四、问答题

1. 操作系统是一组控制和管理计算机软硬件资源、合理地组织计算机工作流程、以及方便用户的程序的集合 1 分

操作系统有处理机管理、存储器管理、设备管理及文件管理四大资源管理功能。 4 分

2. 设备独立性是指程序中所使用的设备与具体物理设备无关。 1 分

I/O 控制方式有：程序直接控制方式、中断控制方式、DMA 控制方式、通道控制方式。 4 分

五、先来先服务

| 作业 | 提交时间 | 运行时间 | 开始时间 | 完成时间 | 周转时间 | 带权周转时间 |
|----|------|------|-------|-------|------|--------|
| 1 | 8:00 | 70 | 8:00 | 9:10 | 70 | 1 |
| 2 | 8:20 | 40 | 9:10 | 9:50 | 90 | 2.25 |
| 3 | 9:00 | 10 | 9:50 | 10:00 | 60 | 6 |
| 4 | 9:10 | 30 | 10:00 | 10:30 | 80 | 2.67 |

平均周转时间 $T=75$ 2.5 分

平均带权周转时间 $W=2.98$ 2.5 分

短作业优先

| 作业 | 提交时间 | 运行时间 | 开始时间 | 完成时间 | 周转时间 | 带权周转时间 |
|----|------|------|------|-------|------|--------|
| 1 | 8:00 | 70 | 8:00 | 9:10 | 70 | 1 |
| 2 | 8:20 | 40 | 9:50 | 10:30 | 130 | 3.25 |
| 3 | 9:00 | 10 | 9:10 | 9:20 | 20 | 2 |
| 4 | 9:10 | 30 | 9:20 | 9:50 | 40 | 1.33 |

平均周转时间 $T=65$ 2.5 分

平均带权周转时间 $W=1.895$ 2.5 分

六、状态安全结论（3分）、安全序列（2分）、检测过程（2分）、无法分配（2分）原因（1分）

根据题设条件可得 need 矩阵如下：

| | | | |
|----|---|---|---|
| P0 | 7 | 5 | 3 |
| P1 | 1 | 2 | 2 |
| P2 | 6 | 1 | 0 |
| P3 | 0 | 2 | 1 |
| P4 | 4 | 2 | 2 |

利用安全性算法对此时刻的资源分配情况进行分析，可得到如下表所示的安全性检测情况。从中可以看出，存在安全序列 P3、P1、P0、P2、P4，故该系统状态安全。

| 进程 | Work | Need | Allocation | Work+Allocation | Finish |
|----|--------|-------|------------|-----------------|--------|
| P3 | 3 3 1 | 0 2 1 | 2 1 2 | 5 4 3 | true |
| P1 | 5 4 3 | 1 2 2 | 2 1 0 | 7 5 3 | true |
| P0 | 7 5 3 | 7 5 3 | 0 2 0 | 7 7 3 | true |
| P2 | 7 7 3 | 6 1 0 | 3 0 2 | 10 7 5 | true |
| P4 | 10 7 5 | 4 2 2 | 0 1 2 | 10 8 7 | true |

若 P2 发出请求向量 $Request_2(3, 1, 1)$ ：

因 $Request_2(4, 1, 0)$ 小于 $Need_2(6,1,0)$ ，

$Request_2(4, 1, 0)$ 不小于 $Available(3,3,1)$ ，系统没有足够的资源满足 P2 的申请要求，因此系统暂时无法将资源分配给 P2。

七、每个整数占2字节，每页大小100字节，1列有50个整数，所以1列刚好放在1页中

由于数组初始化程序是按行进行的，而数组是按列存放，因此每次缺页中断调进一页后，位于该页内的1个数组元素赋予0值，然后再调入下一页，所以涉及的页面走向为 $m, m+1, \dots, m+49$ ，故内层循环缺页次数为50次

外层循环50次，故缺页共2500次

八、（1）semaphore mutex=1; 4分

```
semaphore write=1;
int count=0;
main()
{
    cobegin
        reader();
        writer();
    coend }
reader()
{
    while(true)
    {
        p(mutex);
        if(count==0) p(write);
        count ++;
        v(mutex);
        读文件;
    }
}
```

```

        p(mutex);
        count -- ;
        if (count==0) v(write);
        v(mutex);    }    }
writer()
{    while(true)
    {    p(write);
        写文件;
        v(write);    }    }

```

(2) 6 分:

```

semaphore mutex=1;
semaphore write=1;
semaphore s=1;
int count=0;
main()
{    cobegin
        reader();
        writer();
    coend }
reader()
{    while(true)
    {    p(s);
        p(mutex);
        if(count==0) p(write);
        count ++;
        v(mutex);
        v(s);
        读文件;
        p(mutex);
        count -- ;
        if(count==0) v(write);
        v(mutex);    }    }
writer()
{    while(true)
    {    p(s);
        p(write);
        写文件;
        v(write);
        v(s);    }    }

```