



《中山大学授予学士学位工作细则》第六条

考试作弊不授予学士学位

计算机科学系 2012 第二学期

《操作系统原理》期末考试试题(A)

任课教师：凌应标 & 李才伟 考试形式：闭卷 考试时间：2 小时

年级：11 班别：1~3 专业：计科 姓名：\_\_\_\_\_ 学号：\_\_\_\_\_ 成绩\_\_\_\_\_

一、单项选择题(本大题共 10 小题，每小题 1 分，共 10 分)

在每小题列出的四个备选项中只有一个是符合题目要求的，请将其代码填写在题后的括号内。错选、多选或未选均无分。

- 1、在双处理机的机器中，若有4个用户进程，则处于就绪状态的用户进程的个数最多为( )。  
A. 1 B. 2 C. 3 D. 4
- 2、在动态分区的放置算法中，性能最差的算法通常是( )。  
A. 首次适配 B. 邻近适配 C. 最佳适配 D. 其他适配
- 3、竞态指一组并发执行的进程产生了不可预测的结果，它发生的最关键的原因是一组进程( )。  
A. 共享内存的代码 B. 共享内存的变量 C. 共享CPU D. 共享系统总线
- 4、在进程短程调度的下列算法中，可以实现的为( )。  
A. 最短进程优先SPN B. 最短剩余时间SRT C. 最高响应比优先HRRN D. 时间片轮转RR
- 5、一组进程共享一批资源，资源按序分配法可以防止在这批资源上发生死锁，因为这种方法破坏了( )。  
A. 死锁的所有必要条件 B. 互斥条件 C. 占有且等待条件 D. 环路等待条件
- 6、在请求页式存储管理中，如果采用( )算法进行页替换，则可能产生系统抖动(或颠簸)现象。  
A. 最近最少使用 LRU B. 先进先出 FIFO C. 二次机会 CLOCK D. 最优替换 OPT
- 7、在处理器调度中，如果采用( )去选作业，则计算时间长的进程可能产生长时间等待。  
A. 先来先服务算法 B. 计算时间短优先算法 C. 最高响应比优先算法 D. 时间片轮转算法
- 8、在磁盘调度算法中，如果采用( )算法，完成一批磁盘读写操作的总移臂量最小。  
A. 先来先服务 B. 最小移臂优先 C. 电梯调度 D. 随机调度
- 9、在I/O数据传输控制方式的发展过程中，如果采用( )，则CPU利用率最差。  
A. 程序直接控制式I/O B. 中断方式I/O C. DMA式I/O D. 通道式I/O
- 10、如果采用( )组织文件存储，则随机存取效率最差。  
A. 连续文件 B. 链接文件 C. 索引文件 D. 哈希文件

二、多项选择题(本大题共 5 小题，每小题 2 分，共 10 分)

在每小题列出的五个备选项中至少有两个是符合题目要求的，请将其代码填写在题后的括号内。错选、多选、少选或未选均无分。

1、CDE 2、CD 3、BCDE 4、BCDE 5、BCE

1、下面关于系统进程和用户进程的说法中正确的是 ( )。

- (A) 系统进程和用户进程是进程的两种不同结构；
- (B) 调用操作系统功能的进程称为系统进程；
- (C) 完成用户功能的进程称为用户进程；
- (D) 系统进程实现操作系统功能；
- (E) 系统进程可以运行在用户态。

2、关于操作系统结构，下列说法正确的有 ( )。

- (A) 单体内核易于扩展新功能
- (B) 分层内核下层模块可以调用上层模块
- (C) 微内核的系统进程之间不能互相调用
- (D) 微内核具有更好的安全性
- (E) 微内核具有更高的运行效率

3、关于有挂起状态的进程模型，下列说法正确的有 ( )。

- (A) 进程必定要经历就绪和阻塞状态
- (B) 运行的进程可以转入阻塞状态
- (C) 阻塞的进程可以转入挂起状态
- (D) 挂起的进程不能转入执行状态
- (E) 就绪的进程可以转入挂起状态

4、关于 I/O 的缓冲，下列说法正确的有 ( )。

- (A) 缓冲技术可提高设备利用率
- (B) 缓冲技术不能减少 I/O 传输数据总量
- (C) 缓冲技术允许进程处理数据与设备传输数据的并行
- (D) 增加缓冲区有利于提高命中率
- (E) 操作系统采用软件方法实现缓冲技术

5、关于虚拟存储器，下列说法正确的有 ( )。

- (A) 虚拟存储器大小取决于辅存的容量，与物理内存的大小无关
- (B) 虚拟存储器必须采用动态地址转换
- (C) 虚拟存储器必须采用部分加载方式
- (D) 虚拟存储器必须采用不连续分配方式

(E) 虚拟存储器结构与物理内存结构无关

### 三、填空题(本大题共 5 小题，每空 3 分，共 15 分)

- 1、 如果一个在设备 A 上 I/O 均匀分布于进程的 CPU 占用率为 30%，另一个在设备 B 上 I/O 均匀分布于进程的 CPU 占用率为 20%，这两个进程在单个（单核）CPU 上并发执行，忽略 OS 执行占用的 CPU 时间，且设备 A 与 B 的控制器完全独立，则 CPU 利用为\_\_\_\_\_%。
- 2、 如果进程 A 有 2 条顺序执行的指令，进程 B 有 3 条顺序执行的指令，那么在 SMP 结构两个 CPU 的机器内并发执行这两个进程，不同的相对时序有\_\_\_\_\_种。
- 3、 在基本的进程状态模型中，一个执行中的进程可能进入的下一状态有\_\_\_\_\_种。
- 4、 在分页管理中，如果页尺寸为 1K，逻辑地址是 4320 对应的页号为\_\_\_\_\_。
- 5、 如果在 CPU 和内存之间设立了缓存，缓存命中率为 0.9，读缓存时间为 1，而读内存的时间为 100，则 CPU 访问内存的平均时间为\_\_\_\_\_。

### 四、简答题（本大题共 5 小题，每小题 4 分，共 20 分）

- 1、 中断技术在操作系统中的作用是什么？
- 2、 进程切换的主要工作有哪些？
- 3、 为什么多线程比多进程更有效率？
- 4、 用 EXCH 指令如何实现互斥？
- 5、 进程为什么不直接使用物理地址空间？

### 五、应用分析题（共 4 题，共 45 分）

1、（10 分）某操作系统中，支持信号量机制，且用系统调用 `ccprint` 实现向屏幕输出一个汉字，`ccprint` 是原语。例如进程或线程中调用 `ccprint('汉')` 可在屏幕当前位置显示一个‘汉’字。如果想利用 4 个进程，分别输出‘爱’、‘我’、‘中’、‘大’这四个汉字，协作完成在屏幕上输出“我爱中大”和“爱我中大”这样的信息，不要输出其它结果，如“我中大爱”或“中爱大我”，诸如此类。请描述如何利用信号量实现这 4 个进程的同步，要求说明用到几个信号量，每个信号量的初值是什么，每个进程对应的程序中如何调用 P 操作（即 `semWait` 操作）或 V 操作（即 `semSignal` 操作）。可参考下面的程序框架来回答问题。

```
Program printAWZD;
Var
    semaphore /*在这里信号量声明并初始化 */
Procedure printA();
Begin
    /*可以在这里对信号量操作进行调用 */
    ccprint('爱');
    /*可以在这里对信号量操作进行调用 */
End
Procedure printW();
Begin
    /*可以在这里对信号量操作进行调用 */
    ccprint('我');
    /*可以在这里对信号量操作进行调用 */
End
Procedure printZ();
Begin
```

```
/*可以在这里对信号量操作进行调用 */
    ccprint('中');
    /*可以在这里对信号量操作进行调用 */
End
Procedure printD();
Begin
    /*可以在这里对信号量操作进行调用 */
    ccprint('大');
    /*可以在这里对信号量操作进行调用 */
End

Begin
    /*主程序 创建 4 个进程*/
    Parbegin
        printD();/*创建输出‘大’字的进程*/
        printZ();/*创建输出‘中’字的进程*/
        printW();/*创建输出‘我’字的进程*/
```

```

    printA());/*创建输出‘爱’字的进程*/
Parent
End

```

2、（15 分）中大东校区拟开发校园游，方案规定：

1) 为了不影响教学，每天最多 100 个游客可以在校园内观光。

2) 统一在校门处设立触屏联网自助入口卡机和出口卡机各一个，每个游客在进入时自行操作，入口和出口卡机在检验游客的有效证件后放行，一次一人，自觉排队。

3) 有一辆 12 座新款电动观光车，规定游客必须坐车观光，司机在满座后才开车，车未停定，不准下车。

在方案实施前，学校管理层要求编程模拟该方案。请你用进程模拟某一天宋江带领 108 条好汉来中大东校区观光及司机的活动，用信号量和相关操作实现它们的同步协作。可参考下面的程序框架回答问题。必要时，可以在程序的任意行增加对信号量的操作。

Program Solution;

Var

```

    semaphore    /*在这里信号量声明并初始化 */

```

Procedure Visitor(i);

Begin

```

    ReadVisitorGuid();/*了解大屏幕上的观光说明*/

```

```

    DIY_CheckIn();/*检入*/

```

```

    GetOnCar();/*上车等待*/

```

```

    Visiting();/*随车观光：自由拍照、吃小食*/

```

```

    GetOffCar();/*观光结束，下车*/

```

```

    DIY_CheckOut();/*检出*/

```

End

Procedure Driver();

Begin

```

    While(1) {

```

```

        /*等待座满*/

```

```

        Start_Car();

```

```

        VisitRouteDriving();

```

```

        StopCar();

```

```

    }

```

End

Begin

```

    /*主程序 创建 108 个游客进程和 1 个司机进程*/

```

```

    Parbegin

```

```

        For(i=1;i<=108;i++)/*创建 108 个游客进程*/

```

```

            Visitor(i);

```

```

        Driver();

```

```

    Parend

```

End

3、（10 分）在**请求分页**储存管理系统中，设一个作业访问页面的序列为 1、2、3、4、3（5）、2、1、4、3、5、4、3、2、1、5。设分配给该进程的存储页框有 4 块，且最初未装入任何页。试给出 LRU 算法和 Clock 算法的工作过程描述，并注明缺页情况和计算缺页率。

4、（10 分）某 FAT12 文件系统中每个文件控制块(目录项)信息有 32 字节，根目录最多 224 项，磁盘块大小 512 字节。回答下列问题：

1) 在根目录中检索一个条目时，所读磁盘块的数量范围和平均值各是多少？

2) 如果采用 3 层的层次目录，且各目录中的文件目录项数量都是 224 个。在检索第三层目录中的一个条目时，所读磁盘块的数量范围和平均值各是多少？

## 参考答案

### 一、单项选择题(本大题共 10 小题，每小题 1 分，共 10 分)

1、C(D) 2、C 3、B 4、D 5、D 6、B 7、B 8、B 9、A 10、A(B)

### 二、多项选择题(本大题共 5 小题，每小题 2 分，共 10 分)

1、CDE 2、CD 3、BCDE 4、BCDE 5、BCE

### 三、填空题(本大题共 5 小题，每空 3 分，共 15 分)

1、44 ( $=30+20*0.7=20+30*0.8$ ) 2、10 3、3(2) 4、4 5、11(10.9) ( $1*0.9+(100*0.1+1*0.1)=11$ )

### 四、简答题(本大题共 5 小题，每小题 4 分，共 20 分)

- 1、提高 CPU 的利用率、OS 获取 CPU 的控制权(进程调度)、系统调用
- 2、保护处理器上下文环境、更新当前处于运行态进程 PCB 的控制信息并修改状态、将该进程 PCB 挂入相应队列、选择一个就绪进程、更新所选进程 PCB(包括状态)、更新存储管理数据结构(涉及地址转换)、恢复被选中进程的处理器上下文环境
- 3、线程不拥有资源，且与进程的其他有线程共享进程的代码和地址空间，进程切换的开销小
- 4、门闩 bolt=0：临界区无进程、bolt=1：只有 key=0 的进程在临界区。只有发现 bolt=0 的进程才能进入临界区。算法的本质： $\text{bolt} + \sum \text{key}_i = n$ 。参考代码：

```
const int n= /* number of processes */
```

```
int bolt;
```

```
void p(int i) {
```

```
    int keyi=1;
```

```
    while (true) {
```

```
        do exchange(keyi, bolt);
```

```
        while (keyi!=0);
```

```
        /* critical section */
```

```
        bolt=0;
```

```
        /* remainder */
```

```
    }
```

```
}
```

```
void main() {
```

```
    bolt=0;
```

```
    parbegin(p(1), p(2), ... , p(n));
```

```
}
```

5、程序重定位、虚拟内存管理

### 五、应用分析题(共 4 题，共 45 分)

1、

```
Program printAWZD;
```

```
Var
```

```
    semaphore  a=0, w=0, z=0;
```

```
Procedure printA();
```

```
Begin
```

```
    ccprint( '爱' );
```

```
    V(a);
```

```
End
```

```
Procedure printW();
```

```
Begin
```

```
    ccprint( '我' );
```

```
    V(w);
```

```
End
```

```
Procedure printZ();
```

```
Begin
```

```
    P(a); P(w);
```

```
    ccprint( '中' );
```

```

    V(z);
End

Procedure printD();
Begin
    P(z);
    ccprint( '大' );
End

```

2、

Program SolutionA;

Var

```

    semaphore  sVisitorCount=100;
    semaphore  sSeatOnCar=12;
    semaphore  sVisitorOnCar=0;
    semaphore  sEndStop=0;

```

Procedure Visitor(i);

Begin

```

    ReadVisitorGuid();/*了解大屏幕上的观光说明*/
    P(sVisitorCount);
    DIY_CheckIn();/*检入*/
    P(sSeatOnCar);
    GetOnCar();/*上车等待*/
    V(sVisitorOnCar);
    Visiting();/*随车观光：自由拍照、吃小食、说唱*/
    P(sEndStop);

```

Begin

/\*主程序 创建 108 个游客进程和 1 个司机进程\*/

Parbegin

for(int i=1; i<=108; i++)/\*创建 108 个游客进程\*/

Visitor(i);

Driver();

Parend

End

3、

LRU:

1	2	3	4	3	2	1	4	3	5	4	3	2	1	5
1	1	1	1	1	1	1	1	1	1	1	1	2	2	2
	2	2	2	2	2	2	2	2	5	5	5	5	1	1
		3	3	3	3	3	3	3	3	3	3	3	3	3
			4	4	4	4	4	4	4	4	4	4	4	5
X	X	X	X						X			X	X	X

缺页率：8/15 (4/15)

Begin

/\*主程序 创建 4 个进程\*/

Parbegin

```

    printD();/*创建输出 ‘大’ 字的进程*/
    printZ();/*创建输出 ‘中’ 字的进程*/
    printW();/*创建输出 ‘我’ 字的进程*/
    printA();/*创建输出 ‘爱’ 字的进程*/

```

Parend

End

GetOffCar();/\*观光结束，下车\*/

V(sSeatOnCar);

V(sVisitorCount);

DIY\_CheckOut();/\*检出\*/

End

Procedure Driver();

Begin

while(1) {

/\*等待座满\*/

for(int i=0; i<12; i++) P(sVisitorOnCar);

Start\_Car();

VisitRouteDriving();

StopCar();

for(int i=0; i<12; i++) V(sEndStop);

}

End

CLOCK:

1	2	3	4	3	2	1	4	3	5	4	3	2	1	5
1*	1*	1*	→1*	→1*	→1*	→1*	→1*	→1*	5*	5*	5*	5*	5	5*
→	2*	2*	2*	2*	2*	2*	2*	2*	→2	→2	→2	→2*	1*	1*
	→	3*	3*	3*	3*	3*	3*	3*	3	3	3*	3*	→3	→3
		→	4*	4*	4*	4*	4*	4*	4	4*	4*	4*	4	4
X	X	X	X						X				X	

缺页率: 6/15 (2/15)

4、

$224 \times 32 / 512 = 14$

1) 范围 1~14 块, 平均  $(1+14)/2 = 7.5$  块

2) 范围  $3 \times (1 \sim 14) = 3 \sim 42$  块, 平均  $(3+42)/2 = 3 \times 7.5 = 22.5$  块