

座位号：

杭州电子科技大学学生考试卷（A）卷							6. 程序并发运行时失去了封闭性，其运行结果具有不可再现的特征，而进程的并发运行则具有封闭性和可再现性的特征。	
考试课程	操作系统		考试日期	2012 年 1 月 日		成绩		7. 时间片轮转调度算法只适用于进程调度，而不适用于作业调度。
课程号		教师号		任课教师姓名		赵伟华, 周旭, 梁红兵		8. 把所有作业移动到内存的一端，从而在内存的另一端形成一个大的空闲分区的技术叫动态重定位。
考生姓名		学号 (8 位)		年级		专业		9. CSCAN 磁盘调度算法可以避免磁臂粘着现象的发生。
备注：判断题、选择题、填空题的答案请全部写在后面的答题纸上；综合题的答案请写在答题纸或白纸上。试卷和答题纸全部上交。							10. 分页系统页面的大小可根据需要由系统管理员设定为某个值，但该值必须是 2 的幂。	
							二、选择题（每选 1 分，共 24 分）	
一、判断题(共 10 分,每小题 1 分)							1. 现有以下计算机的应用场合，请为其选择适当的操作系统：① 航空航天，核变研究 <u>(1)</u> ；②机房学生上机学习编程 <u>(2)</u> ；③民航机票订购系统 <u>(3)</u> 。	
							A. 配置实时操作系统； B. 配置批处理操作系统； C. 配置分时操作系统； D. 配置网络操作系统。	
1. 分配内存、进程调度以及将高级语言程序翻译成机器语言这些工作都是由操作系统完成的。							2. 进程的三个基本状态是 <u>(4)</u> 、 <u>(5)</u> 、 <u>(6)</u> 。由（4）到（5）是由进程调度所引起的；由（5）到（6）是正在执行的进程发生了某事件，使之无法继续执行而引起的。	
							A. 挂起； B. 阻塞； C. 就绪； D. 执行； E. 完成。	
2. 在文件系统中引入磁盘索引结点的主要目的是为了加快在目录中检索文件的速度。							3. 下述存储管理方式中，会产生外部碎片的是 <u>(7)</u> ，会产生内部碎片的是 <u>(8)</u> 。	
							A. 页式和段式； B. 页式和段页式；	
3. 采用多道程序设计技术后，在同一时刻也可在一台处理机上运行多个程序。								
4. 在消息缓冲队列通信中，消息队列属于临界资源，因此多个进程必须对它进行互斥地访问。								
5. 临界区是指进程中用来实现互斥的那段代码。								

座位号：

<p>C. 动态分区方式和段页式；</p> <p>4. 下述页面淘汰算法中，<u>(9)</u> 置换算法会产生“分配给进程的内存块数增加后，反而可能导致缺页次数增加”的现象。</p> <p>A. 最近最少使用；</p> <p>C. 最近最久未用；</p> <p>5. 下述进程调度算法中，<u>(10)</u> 可使所有交互型作业均能获得较好的响应时间；<u>(11)</u> 能满足短作业优先且不会产生饥饿现象。</p> <p>A. 先来先服务；</p> <p>C. 高响应比优先；</p> <p>6. 在支持多线程的系统中，隶属于同一个进程的多个线程不能共享的是 <u>(12)</u>。</p> <p>A. 进程的代码段；</p> <p>C. 进程所打开的文件；</p> <p>7. 下列信息中，<u>(13)</u> 不属于 CPU 现场信息。</p> <p>A. 指令计数器；</p> <p>C. 堆栈的栈顶指针；</p> <p>8. 下列进程间通信方式中，<u>(14)</u> 的通信效率最高；<u>(15)</u> 的通信效率最低；<u>(16)</u> 是利用共享的文件来完成进程间通信。</p> <p>A. 共享数据结构；</p> <p>C. 管道；</p>	<p>D. 动态分区方式和段式</p> <p>B. 先进先出；</p> <p>D. 最佳。</p> <p>B. 短作业优先；</p> <p>D. 时间片轮转。</p> <p>B. 进程的全程变量；</p> <p>D. 保存函数参数、返回地址等信息的堆栈。</p> <p>B. 进程的就绪、阻塞、执行等基本状态；</p> <p>D. 段表控制寄存器。</p> <p>B. 共享存储区；</p> <p>D. 消息缓冲队列。</p>	<p>9. 引入缓冲技术的主要目的是 <u>(17)</u>。</p> <p>A. 改善用户编程环境；</p> <p>C. 提高 CPU 与设备之间的并行程度；</p> <p>B. 提高 CPU 的处理速度；</p> <p>D. 降低计算机的硬件成本。</p> <p>10. 采用动态分区分配方式回收内存时，若释放区前的分区 F1 也是空闲分区，则应 <u>(18)</u>；若释放区前后的分区 F1、F2 均已被分配出去，则应 <u>(19)</u>；若释放区前后的分区 F1、F2 均为空闲分区，则应 <u>(20)</u>。</p> <p>A. 为回收区建立一分区表项，填上分区的大小和始址；</p> <p>B. 以 F1 分区的表项作为新表项，修改新表项的大小，删除 F2 的表项；</p> <p>C. 以 F1 分区的表项为新表项，但修改新表项的大小；</p> <p>D. 以 F2 分区的表项作为新表项，同时修改新表项的大小和始址。</p> <p>11. 打开文件操作时的主要工作是 <u>(21)</u>：</p> <p>A. 把指定文件的 FCB 复制到内存指定的区域；</p> <p>B. 把指定文件复制到内存指定的区域；</p> <p>C. 在指定文件所在的内存介质上找到指定文件的目录；</p> <p>D. 在内存寻找指定的文件。</p> <p>12. 为了实现设备独立性，应用程序必须通过 <u>(22)</u> 来请求使用某类设备。</p> <p>A. 物理设备名；</p> <p>C. 虚拟设备名；</p> <p>B. 逻辑设备名；</p> <p>D. 设备牌号。</p> <p>13. 在文件系统中可设置一张 <u>(23)</u> 表，其中的每一个二进制位可用来表示磁盘中的一个块的使用情况；若在文件系统中配置一张 <u>(24)</u> 表，则既可以用它来管理空闲盘块，又可以用它来指出分配给文件的下一个盘块的物理地址。</p> <p>A. 文件描述符表；</p> <p>C. 空闲区表；</p> <p>B. 文件分配表(FAT)；</p> <p>D. 位示图。</p>
--	---	--

座位号:

三、填空题（每空 1 分，共 25 分）

- 1. 采用多道程序设计技术的现代操作系统最基本的两个特征是(1) 和 (2)。
- 2. 进程实体由 (3)、(4) 和 (5) 三部分组成。
- 3. 进程之间的直接制约主要源于 (6)，而进程之间的间接制约则源于 (7)。
- 4. 进程调度可采用抢占和非抢占两种方式，其中抢占的主要原则有 (8)、(9) 和 (10)。
- 5. 在采用信号量实现临界资源的互斥共享时，当 $S.Value > 0$ 时，信号量的值表示 (11)；而当 $S.Value < 0$ 时，其绝对值表示 (12)。
- 6. 一次性为进程分配所有资源，可破坏产生死锁的(13)必要条件；将所有资源编号，并要求进程严格按资源号递增的次序申请资源则可破坏 (14) 条件；而采用 SPOOLING 技术则可破坏 (15) 条件。
- 7. 在请求分页存储管理方式中，地址变换过程中可能会因为 (16)、(17) 和 (18) 等原因而产生中断。
- 8. 用户在编程时通常采用符号名来表示变量和指令的地址，编译程序会将这些地址转换成 (19) 地址，而 CPU 执行指令时则将通过 (20) 地址来访问数据和指令。
- 9. 现代操作系统通常采用 (21) 的文件逻辑结构。
- 10. 在引入索引结点的文件系统中，目录项中只需包含 (22) 和 (23) 两个数据项。
- 11. 按信息交换的单位来分，磁盘属于 (24) 设备，磁盘的 I/O 通常采用 (25) 控制方式。

四、综合题（共 41 分）

- 1. (8 分)有些孩子想要画画，其中有的孩子手上握了一支笔，有的孩子手上拿着一张纸，他们必须同时拿到纸和笔之后才可以开始画画。张老师负责源源不断地提供纸或笔，每次只提供一件并将它放在桌上，如果他放的是笔，则缺笔的孩子可以拿走笔并开始画画；如果他放的是纸，则缺纸的孩子可以拿走纸开始画画。假设桌子上同时存放的纸和笔的数量不受限制。
 - (1) 请问这个问题中有哪些进程？
 - (2) 请用信号量来解决上述进程的同步问题。
- 2. (5 分) 试说明设备驱动程序应完成哪些功能。
- 3. (8 分) 与基本分页系统的页表相比较，请求调页系统的页表中应该增加哪些数据项？每项的作用是什么？
- 4. (8 分) 在一个采用段页式存储管理的系统中，页的大小为 1KB。某个正在执行的作业情况如下：
 - (1) 请给出将逻辑地址 (2, 3500) 转换成物理地址的过程；
 - (2) 计算出上述逻辑地址对应的物理地址的值。

座位号：

段表控制寄存器

	3
--	---

段表

0	4	
1	6	
2	5	

0	22
1	23
2	29
3	30
4	36

页表 2

0	15
1	17
2	18
3	19
4	20
5	21

页表 1

0	7
1	8
2	10
3	11

页表 0

5. (6 分) 在银行家算法中，若出现下述资源分配情况：

Process	Allocation	Need	Available
P0	0 0 3 2	0 0 1 2	1 6 2 2
P1	1 0 0 0	1 7 5 0	
P2	1 3 5 4	2 3 5 6	
P3	0 3 3 2	0 6 5 2	
P4	0 0 1 4	0 6 5 6	

试问：(1) 该状态是否安全？（必须写出安全性检查的过程）

(2) 若进程 P2 提出请求 Request (1, 2, 2, 2) 后，系统能否将资源分配给它？为什么？

6. (6 分) 在某个采用混合索引分配的文件系统中，FCB 中有 i_addr[0]~i_addr[8] 共 9 个物理地址项，其中 i_addr[0]~i_addr[6] 是 7 个直接地址项，i_addr[7] 是 1 个一次间址项，i_addr[8] 是 1 个二次间址项。如果一个盘块的大小是 4KB，每个盘块号占 4 个字节。请写出将下列文件的字节偏移量转换成物理地址的过程：

(1) 10000；(2) 500000。

座位号：

[答题纸]

一、判断题(共 10 分,每小题 1 分)

1. X 2. √ 3. X 4. √ 5. X
6. X 7. √ 8. X 9. X 10. X

二、选择题（每选 1 分，共 24 分）

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

- (21) A (22) B (23) D (24) B

三、填空题（每空 1 分，共 25 分）

- (1)_____ 并发 ； (2)_____ 共
享 _____ ；
(3)_____ PCB _____； (4)_____ 程序
段 _____ ；
(5)_____ 数据段 _____； (6)_____ 进程合
作 _____ ；
(7)_____ 互斥共享资源 _____； (8)_____ 时间片原
则 _____ ；
(9)_____ 短作业优先原则 _____； (10)_____ 最高优先权优先
原则 _____ ；
(11)_____ 可用的该类临界资源数 _____； (12)_____ 等待该资源而阻塞

座位号：

的进程数_____；	【注】（1）和（2）可交换次序；
(13)_____请求与保持_____；(14)_____环路等待_____；	（3）、（4）、（5）次序可互换；
(15)_____互斥_____；(16)_____地址越界_____；	（8）、（9）、（10）次序可互换；
(17)_____缺页_____；(18)_____访问权限非法_____；	（16）、（17）、（18）次序可互换；
(19)_____逻辑_____；(20)_____物理_____；	（22）和（23）次序可互换。
(21)_____字符流式_____；(22)_____文件名_____；	 四、综合题（共 41 分）
(23)_____磁盘索引结点编号_____；(24)_____块_____；	1、（1）每个孩子是一个进程，张老师也是一个进程。
(25)_____DMA_____。	（2）本题中需要设置两个信号量 paper 和 pencil，它们的初值均赋值为 0。 张老师对应的算法可描述为：

```
Teacher(){
    while(1){
        取一件物品放桌上；
        if (放的是笔) signal(pencil);
        else signal(paper);
    }
}
```

已经拥有笔的孩子可共享同一个程序段，其对应的算法可描述为：

```
Child1(){
    wait(paper);
    从桌上拿起纸；
    开始画画；
}
```

已经拥有纸的孩子可共享另一个程序段，其对应的算法可描述为：

座位号：

<pre>Child2(){ wait(pencil); 从桌上拿起笔； 开始画画； }</pre>	<p>找到段表中对应于第 2 段的段表项，从而得到第 2 段的页表长度和页表起始地址；③根据段内偏移 3500 得到段内页号为 3，页内偏移为 428；④把段内页号 3 和段表项中的页表长度 5 进行比较，得知页号没越界；⑤页表起始地址加上页号得到对应的页表项，从而得到第 2 段的第 3 页的物理块号 30；⑥物理块号 30 和块内偏移 428 拼接成物理地址 31148。</p> <p>(2) 逻辑地址 (2, 3500) 对应的物理地址为十进制数 31148，十六进制 79AC。</p>																																																																																																																																																																																																												
<p>2. 设备驱动程序应完成以下工作：</p> <p>(1) 将抽象要求转换成具体要求，如用户要求访问磁盘的某个盘块时，将用户的抽象要求中的盘块号转换成磁盘的具体的盘面、磁道号、扇区号。</p> <p>(2) 检查 I/O 请求的合法性。如对输出设备，用户要求用它进行输入，则请求非法。</p> <p>(3) 读出和检查设备的状态。如，在 I/O 前必须先必须从设备控制器的状态寄存器中读出状态，只有设备处于空闲态时，才能启动它进行下一次 I/O。</p> <p>(4) 传送必要的参数。如读块设备时，要向控制器传送本次读的字节数、内存始址等参数。</p> <p>(5) 方式的设置。对有多种工作方式的设备，应设置其工作方式。如对 RS-232 接口进行通信前，应设置其波特率、奇偶校验方式等。</p> <p>(6) 启动 I/O 设备。最后，驱动程序应向设备控制器的命令寄存器发相应的控制命令，启动设备进行真正的 I/O。</p>	<p>5. 答：(1) 对该状态进行安全性检查：</p> <table><tr><th></th><th colspan="4">work</th><th colspan="4">need</th><th colspan="4">allocation</th><th colspan="4">work+allocation</th></tr><tr><th>finish</th><th>A</th><th>B</th><th>C</th><th>D</th><th>A</th><th>B</th><th>C</th><th>D</th><th>A</th><th>B</th><th>C</th><th>D</th><th>A</th><th>B</th><th>C</th><th>D</th></tr><tr><td>P0</td><td>1</td><td>6</td><td>2</td><td>2</td><td>0</td><td>0</td><td>1</td><td>2</td><td>0</td><td>0</td><td>3</td><td>2</td><td>1</td><td>6</td><td>5</td><td>4</td></tr><tr><td>true</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>P3</td><td>1</td><td>6</td><td>5</td><td>4</td><td>0</td><td>6</td><td>5</td><td>2</td><td>0</td><td>3</td><td>3</td><td>2</td><td>1</td><td>9</td><td>8</td><td>6</td></tr><tr><td>true</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>P1</td><td>1</td><td>9</td><td>8</td><td>6</td><td>1</td><td>7</td><td>5</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>2</td><td>9</td><td>8</td><td>6</td></tr><tr><td>true</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>P2</td><td>2</td><td>9</td><td>8</td><td>6</td><td>2</td><td>3</td><td>5</td><td>6</td><td>1</td><td>3</td><td>5</td><td>4</td><td>3</td><td>12</td><td>14</td><td>18</td></tr><tr><td>true</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>P4</td><td>3</td><td>12</td><td>13</td><td>14</td><td>0</td><td>6</td><td>5</td><td>6</td><td>0</td><td>0</td><td>1</td><td>4</td><td>3</td><td>12</td><td>14</td><td>18</td></tr><tr><td>true</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table> <p>由安全性检查得知，可以找到一个安全序列 {P0、P3、P1、P2、P4}，因此系统是安全的。</p> <p>(2) 若 P2 提出 Request (1, 2, 2, 2), 系统按银行家算法进行检查，因</p> <p>Request (1, 2, 2, 2) <= Need (2, 3, 5, 6)</p> <p>Request (1, 2, 2, 2) <= Available (1, 6, 2, 2)</p> <p>系统暂时先为进程 P2 试行分配资源，并修改有关的确数据，如下图所示：</p>		work				need				allocation				work+allocation				finish	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D	P0	1	6	2	2	0	0	1	2	0	0	3	2	1	6	5	4	true																	P3	1	6	5	4	0	6	5	2	0	3	3	2	1	9	8	6	true																	P1	1	9	8	6	1	7	5	0	1	0	0	0	2	9	8	6	true																	P2	2	9	8	6	2	3	5	6	1	3	5	4	3	12	14	18	true																	P4	3	12	13	14	0	6	5	6	0	0	1	4	3	12	14	18	true																
	work				need				allocation				work+allocation																																																																																																																																																																																																
finish	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D																																																																																																																																																																																													
P0	1	6	2	2	0	0	1	2	0	0	3	2	1	6	5	4																																																																																																																																																																																													
true																																																																																																																																																																																																													
P3	1	6	5	4	0	6	5	2	0	3	3	2	1	9	8	6																																																																																																																																																																																													
true																																																																																																																																																																																																													
P1	1	9	8	6	1	7	5	0	1	0	0	0	2	9	8	6																																																																																																																																																																																													
true																																																																																																																																																																																																													
P2	2	9	8	6	2	3	5	6	1	3	5	4	3	12	14	18																																																																																																																																																																																													
true																																																																																																																																																																																																													
P4	3	12	13	14	0	6	5	6	0	0	1	4	3	12	14	18																																																																																																																																																																																													
true																																																																																																																																																																																																													

座位号:

P r o c e s s	A l l o c a t i o n	N e e d	A v a i l a b l e
P0	0 0 3 2	0 0 1 2	0 4 0 0
P1	1 0 0 0	1 7 5 0	
P2	2 5 7 6	1 1 3 4	
P3	0 3 3 2	0 6 5 2	
P4	0 0 1 4	0 6 5 6	

现在，所有进程的 Need 都不满足 $Need \leq Availabe$ ，系统将进入不安全状态，所以，不能进行上述的分配，而必须将状态恢复到分配前的状态。

6. (1) $10000/4096=2\cdots1808$
因此字节偏移量 10000 位于逻辑块 2 偏移为 1808 字节的位置。
 $2<7$ ，应采用直接地址来进行地址转换，逻辑块 2 对应的物理块号登记在 $i_addr[2]$ 中。
所以，10000 对应的物理地址是 $i_addr[2]$ 块的 1808 字节。

(2) $500000/4096=122\cdots288$
因此字节偏移量 500000 位于逻辑块 122 偏移为 288 字节的位置。
 $7\leq122<7+1K$ ，应采用一次间址的方式来进行地址转换。
所以，应从 $i_addr[7]$ 中获得一次间址块的物理块号 a，并将第 a 块读入内存。然后便可在一次间址块 a 的第 115 项中获得逻辑块 122 的物理块号 b。块内偏移仍然为 288 字节。