计算机科学与技术学院 2014-2015 学年第一学期操作系统设计与实现考卷 A 卷-参考答案

一、选择题

ABDBD　BCCDA　BA

二、简答题

1. 参考答案：脱机批处理技术是指主 CPU 不参与 I/O 过程，而用卫星 CPU 来处理 I/O，从而避免将主 CPU 的资源浪费在等待 I/O 上。联机批处理则相反，需要主 CPU 参与数据的输入或输出。多道程序设计技术强调在内存中同时存放多道相互独立的程序，它们相互穿插地运行。然而，这种穿插，是以主 CPU 能够与外设并行作为基础的。所以，只有脱机批处理系统适用多道程序设计技术。

（参考）打分标准：联机批处理与脱机批处理的概念（也就是它们两者的区别）各 1 分，脱机批处理系统适用多道程序设计技术，3 分。

2. 参考答案：应用程序对该虚地址的访问（由于其对应内容并未调入内存），必然会导致**异常的产生**。在异常产生后，系统（CPU）将**从用户态转到核心态**继续运行，并尝试处理此异常。在操作系统处理异常的过程中，首先会对应用程序所访问的虚地址进行判断，有两种情况：一种是该虚地址合法，即被访问的地址属于应用程序所对应的虚存空间（如栈的扩展）；另一种情况是该虚地址非法，即被访问的地址不属于应用程序所对应的虚存空间。对于第一种情况，操作系统会通过分配内存（如扩展栈空间），或将所需的代码/数据调入内存，**并建立被访问的虚地址与调入内存的代码/数据的页式地址映射**，从而满足应用程序的访问需求，并在完成后**返回应用程序发生异常的那条访存指令继续运行**。对于第二种情况，出于安全和隔离的考虑，系统会报错，并强行终止该应用程序的运行。 另外，如果操作系统在分配内存时，发现**内存不足**，则会启动淘汰算法选择一个在内存的页面淘汰，腾出空间。

（参考）打分标准：描述异常的产生，以及系统（CPU）状态转换，1 分；将数据或代码调入内存，并建立地址映射，1 分；对异常处理完成后，返回到应用程序的过程，2 分；对内存不足情况的考虑，1 分。可能的问题：学生可能将异常回答为中断，这个不扣分。如果学生只回答了第一种情况，而未考虑访问非法的情况，也不扣分。

3. 参考答案：该文件的绝对路径是/home/apple/document/notes.txt

为了找到 notes.txt 文件，系统会首先打开并查找她的当前工作目录（也就是“.”）所对应的目录文件，找到其父目录“..”所对应的 i 节点号，根据 i 节点号所指向的文件索引表地址，找到父目录所对应的目录文件，并将此目录文件读入内存。接下来在父目录所对应的目录文件中查找 document 所对应的 i 节点号，以此类推，直到找到 notes.txt 文件为止。

（参考）打分标准：关键是回答目录文件->按名查找对应的 i 节点号->新的目录文件这一个过程。可能出现的情况是，学生会先详细讲当前工作目录的打开和查找过程，然而实际上，当前工作目录所对应的目录文件往往早已被载入内存。这种情况，因为题目没有特别指出，所以我感觉就不要扣分了，回答清目录文件->按名查找对应的 i 节点号->新的目录文件这一个过程即可给全分。

4. 参考答案：
定义整型共享变量 x，其初值为 5；
定义信号量 S，其初值为 1，用于控制进程对共享变量 x 的访问；
程序描述：

```
main(){
    cobegin
        P1(); P2(); ……
    coend
}
Pi(){ // (i> 0)
    int action = 0; // 0 表示离开，1 表示坐座位。
    P(S);
    if( x>0 ){
        x = x-1; //或者 x -= 1;
        action = 1;
    }
    V(S);
    if (action)  坐座位；
    else  离开；
}
```

（参考）打分标准：本题的关键，是不应直接对长凳的空位设置信号量，而应该用共享变量来控制游客对长凳的使用。如果直接用信号量来描述空位资源，且游客用 P 操作来申请空位资源的话，会导致没有位置情况下的长时间等待，无法实现离开的动作。建议给直接用信号量来描述空位资源的学生 2 分。

三、综合题 1

参考答案：

（1）FIFO 算法

访问：	4+	3+	2+	1+	4	3	5+	4+	3+	2+	1+	5+
	1	2	3	4	5	6	7	8	9	10	11	12
0	4	4	4	4	4	*4*	5	5	5	*5*	1	1
1		3	3	3	3	3	*3*	4	4	4	*4*	5
2			2	2	2	2	2	*2*	3	3	3	3
3				1	1	1	1	1	*1*	2	2	2

发生缺页中断的次数： 10

依次被淘汰的页面序号：4、3、2、1、5、4

（2）LRU 算法

访问：	4+	3+	2+	1+	4	3	5+	4	3	2+	1+	5+
	1	2	3	4	5	6	7	8	9	10	11	12
0	4	4	4	4	4	4	4	4	4	4	*4*	5
1		3	3	3	3	3	3	3	3	3	3	3
2			2	2	2	*2*	5	5	5	*5*	1	1
3				1	1	1	1	1	*1*	2	2	2

发生缺页中断的次数： 8

依次被淘汰的页面序号：2、1、5、4

（参考）打分标准：这题应该是基本题，难度不算大。学生可能会到草稿纸上去写草稿，而把最终的答案写到答卷上，所以，只要学生给出最终的答案（给出缺页中断的次数，以及被淘汰的页面序号）就算对。

四、综合题 2
参考答案：
（1）程序描述
定义信号量 empty1，初值为 1，表示 BUF1 是否为空；
定义信号量 full1，初值为 0，表示 BUF1 是否有数据；
定义信号量 empty2，初值为 1，表示 BUF2 是否为空；
定义信号量 full2，初值为 0，表示 BUF2 是否有数据；

```
main(){
   cobegin
      P();   D();
   coend
}
P(){
   while(输出未结束){
      P(empty1);
      将数据放入 BUF1；
      V(full1)；
      P(empty2);
      将数据放入 BUF2；
      V(full2)；
   }
}
D(){
   while(1){
      P(full1);
      输出 BUF1 中的内容；
      V(empty1)；
      P(full2);
      输出 BUF2 中的内容；
      V(empty2)；
   }
}
```

（2）速度分析：在双缓冲系统中，系统的总速度取决于 P 和 D 中最慢的那个。在不失一般性的前提下，假设 x>y，即输出设备的速度相对慢一些，则双缓冲系统的速度为 y。
如果采用单缓冲技术，则输出一个数据的时间为 1/x+1/y，系统输出结果的速度为 1/(1/x + 1/y) = xy/(x+y) = x/(x+y) *y。
现仍然假设 x>y，则必然有 y > x/(x+y) *y（因为 x/(x+y) < 1），也就是说，采用双缓冲的系统输出计算结果的速度更快。

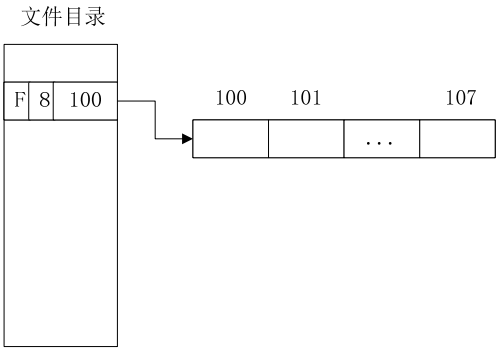
（参考）打分标准：可能出现的问题：1）学生在第一问中可能未写 while 循环，而对于双缓冲系统来说，循环是必须的。所以，如果未写循环，应酌情扣除一定的分数（建议扣 2 分）。2）第二问的速度分析，因为未对 x 和 y 的取值进行限定，所以，如果有学生假设 x=y，这样双缓冲系统的速度为 x，单缓冲的速度为 x/2，这样也是对的。另外，因为第二问有 4

分，建议给出双缓冲系统的速度，给 1 分，给出单缓冲系统的速度，给 1 分，给出两者的对比结果，再给 2 分。

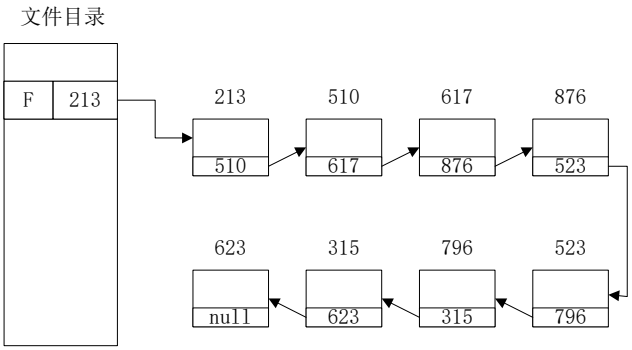
五、综合题 3

参考答案：

（1）连续文件物理结构



（2）串联文件物理结构



（3）若文件采用连续文件结构，需要访问磁盘 5 次；若文件采用串联文件结构，需要访问磁盘 6+8+2+4+3=23 次。

（参考）打分标准：如果答案跟参考答案有出入，可以酌情扣分。建议：第一二问的画图题，错 1 个地方扣 1 分。第三问算错一个扣 2 分。

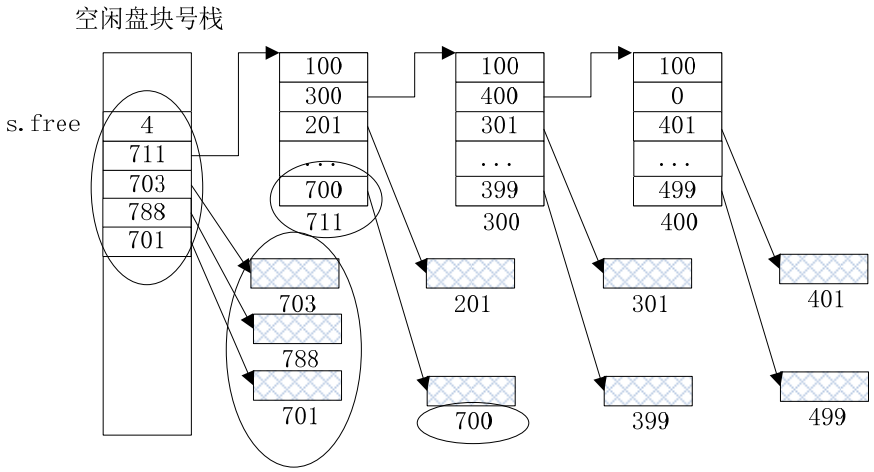
六、综合题 4

参考答案：

（1）该磁盘中目前还有 301 个空闲盘块。

（2）将被分配出去的盘块号为：100、200、299。

（3）回收后的盘块链接情况为：



（参考）打分标准：第三问，注意参考答案中，用椭圆标注的部分，都是计分点。可按学生的答案，酌情扣分。

七、综合题 5

参考答案：

（1）需设置设置 3 个进程用来描述题干中的同步关系，两个食品公司各一个进程 factory1 和 factory2，商店一个进程 shop。

（2）程序描述：

定义信号量 empty1，初值为 m，表示 factory1 能生产食物 A 的个数；

定义信号量 empty2，初值为 m，表示 factory2 能生产食物 B 的个数；

定义信号量 full1，初值为 0，表示商店有 A 食品的个数；

定义信号量 full2，初值为 0，表示商店有 B 食品的个数；

定义信号量 permit1，初值为 k，表示 factory1 能否生产产品；

定义信号量 permit2，初值为 k，表示 factory2 能否生产产品；

定义信号量 mutex，初值为 1，用于对商店的仓库进行访问时的互斥；

```
main(){
   cobegin
      fact1(); fact2(); shop();
   coend
}
shop(){
   P(full1);
   P(full 2);
   P(mutex);
   卖出 1 份 A+B 食品；
   V(mutex);
   V(empty1);
   V(empty2);
}
factory1(){
   P(empty1);
   P(permit1);
   P(mutex);
   生产一个 A 食物，并交给商店；
   V(mutex);
   V(full1);
   V(permit2);
}
factory2(){
   P(empty2);
   P(permit2);
   P(mutex);
   生产一个 B 食物，并交给商店；
   V(mutex);
   V(full2);
   V(permit1);
}
```

（参考）打分标准：若未定义 mutex 来互斥对于商店仓库的访问，可考虑给全分，因本题的重点不在于对商店仓库的访问。其他情况可酌情给分。