中 山 大 学 软 件 学 院 **2010** 级 软 件 工 程 专 业 (2011 春季学期)

《SE-223 操作系统》期末试题试卷（A）

(考 试 形 式： 闭　卷　　考 试 时 间： 2 小 时)

警 示

《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

方向：_____　　姓名：_____　　学号：_____

出卷：常会友、刘宁、凌应标　　审核：_____

一、**Explain following terms(15 pts)**

1. Semaphore
2. Threads
3. Address mapping
4. PCB
5. File system

二、**Short Answer(25 pts)**

1. Please describe the necessary conditions of deadlock.
2. Please describe the interrupt procedure.
3. What is the difference between paging and demand paging? How OS with demand paging deals with page fault, i.e when refers a page that isn't in the memory?
4. Please describe the difference between cache and buffer.
5. What is the difference between sequential access and random access?

三、(**15 pts**)We consider a system with three resource types(A, B, C) and five process(P1, P2, P3, P4, P5). The number of resource A is 17, resource B is 5, resource C is 20. Consider the following snapshot of the system at time T_0.

process	MAX A B C	Allocation A B C	Available A B C
P1	5 5 9	2 1 2	2 3 3
P2	5 3 6	4 0 2	
P3	4 0 11	4 0 5	
P4	4 2 5	2 0 4	
P5	4 2 4	3 1 4	

Answer the following question using the banker's algorithm:

(1)At time T_0, is the system in a safe state? If it is in a safe state, write down the safe sequence.**(7 pts)**

(2)At time T_0, if a request from process P2 arrives for (0, 3, 4), can the request be granted? Why?**(3 pts)**
(3)At time T_0, if a request from process P4 arrives for (2, 0, 1), can the request be granted? Why?**(5 pts)**

四、(**15 pts**) Consider the logical address of a job is 24 bits, the high-order 8 bits for segment-number, the low-order 16 bits for the offset. Try to answer the following question:
(1)How many segments can a job hold at most?**(3 pts)**
(2)What is the max length of a segment?**(3 pts)**
(3)A segment table as follow, try to calculate the physical address for the following logical address:[0, 430]、[1, 50]、[2, 30]、[3, 70](The first number in square brackets is segment-number, the second is offset). And declare what kind of interrupt will generate, when the logical address can not change to physical address.**(9 pts)**

Segment	Length	Base	Is in the main memory
0	600	2100	Y
1	40	2800	Y
2	100		N
3	80	4000	Y

五、(**15 pts**)A system has a page request management schemes. The access page sequence of a process is 4, 3, 2, 1, 4, 3, 5, 4, 3, 2, 1, 5. The system will use **LRU** page replacement algorithm. Answer the following question: (you need to write down the **computational procedure**)
(1)If the process has 4 frames, write down the page fault sequence and calculate the page fault rate.**(6 pts)**
(2)If the process has 5 frames, write down the page fault sequence and calculate the page fault rate.**(6 pts)**
(3)What can we know from the result of (1) and (2)?**(3 pts)**

六、**Please select and answer ONE of the following two questions**
1.(**15 pts**)Consider a multi-level index file system, there are 13 pointers of disk block in the file's i_node(10 direct index, 1 single indirect index, 1 double indirect index, 1 triple indirect index). A date block is 4K, the disk address is 4 bytes. Try to answer the following question:
(1)What is the max file length in this file system?**(7 pts)**
(2)How many space does a 2GB file actually occupy in this file system? (the i_node does not occupy the space)**(8 pts)**

2.(**15 pts**) Consider a system with three smoker processes and one agent process. Each smoker continuously rolls a cigarette and then smokes it. But to roll and smoke a cigarette, the smoker needs three ingredients: tobacco, paper, and matches. One of the smoker processes has paper, another has tobacco, and the third has matches. The agent has an infinite supply of all three materials. The agent places two of the ingredients on the table. The smoker who has the remaining ingredient then makes and smokes a cigarette, signaling the agent on completion. The agent then

puts out another two of the three ingredients, and the cycle repeats.

(1)According to the following example programs in a C-like language, write out the whole program of the smokers and the agent processes where are omitted; **(5 pts)**

(2)Synchronize the agent and the smokers using counting semaphores.**(10 pts)**

Examples programs of the smoker process with paper and the agent process

```
TableType TABLE;

SmokerA () {    /* the smoker processes with paper */
   CigaretteType    c;
   PaperType    p;
   TobaccoType    t;
   MatchesType    m;
   While (1)
      p= Take_Paper_From_The_Pocket_of_SmokerA();
      t=Take_Tobacco_From_ The_Table(TABLE);
      m=Take_Matches_From _The_Table (TABLE);
      c=Make_A_ Cigarette(p,t);
      Smoking(c);
   }
}
SmokerB () {
    /* the smoker processes with tobacco */
   ……
}
SmokerC () {
    /* the smoker processes with matches */
   ……
}
ServerAgent() {
      /* agent process */
   PaperType    p;
   TobaccoType    t;
   MatchesType    m;
   While (1)
      ch=RandamNumberFrom_1_2_3();
      /* generate a randam number From 1,2 or 3 */
      if (ch= =1)    {
            t= Prepair_Tobacco();
            m= Prepair_Matches();
            Put__Tobacco _To_The_Table(TABLE, t);
            Put_Matches_To_ The_Table (TABLE, m);
       }
```

```
        if (ch= =2)    {
              p= Prepair_Paper();
              m= Prepair_ Matches ();
              ……
        }
        if (ch= =3)    {
               p= Prepair_Paper();
               t= Prepair_Tobacco();
               ……
        }
  }
```

一

1.信号量是一种与临界区资源相联系的变量

2.线程是指“进程的一个可调度实体”，是 CPU 调度的基本单位

3.地址映射是指将逻辑地址变换为物理地址

4.PCB 是为了描述进程的运动变化过程而引入的一个与进程相联系的数据结构，用于记录系统管理进程所需的信息，描述进程的瞬间特征。

5.文件系统是指操作系统中负责存取和管理文件信息的机构

二

1、互斥控制，不可剥夺控制，请求和控制，环路等待条件

2、现场信息保护，查找中断源，转中断处理程序，处理完成后恢复现场信息

3、静态页面管理将一个作业的全部地址空间同时装入主存，请求页式管理则根据需要将页面按需调入主存。当所需页面不在主存中时，将发生缺页中断。缺页中断处理程序将所需的页面调入主存。当主存无空闲块时，按一定的页面置换算法将某页淘汰而腾出空间。

4、高速缓存是设备的扩展，它与设备联系紧密，不能离开设备而存在。例如，磁盘缓存就是设在磁盘控制器中的缓冲区，它不能离开磁盘而存在。缓冲区则是设备之间进行沟通的中间地带，用于协调不同设备之间运行速度的差异，它通常独立于设备而存在。

5、顺序存取法就是严格按物理记录排列的顺序依次存取，随机存取法允许随意存取文件中的任何一个物理记录，而不管上次存取了哪一个记录。

三、（1）T_0 时刻是安全状态，因为可以找到一个安全序列（P4、P5、P1、P2、P3）

（2）不能分配。因为所剩余的资源数量不够

（3）可以分配，当分配完成以后，系统剩余的资源数量为（0,3,2），这时仍可以找到一个安全序列队（P4、P5、P1、P2、P3）

四、（1）一个作业最多可以有 2^8=254 个段

（2）每段的最大长度为 2^{16}=64KB

（3）逻辑地址[0, 430]的主存地址为：2100+430=2530

逻辑地址[1, 50]的段内地址超过段长，无法进行地址变换，将产生越界中断。

逻辑地址[2, 30]所在的第 2 段没有驻存在内存中，无法进行地址变换，将产生缺段中断。

逻辑地址[3, 70]的主存地址为：4000+70=4070

五、（1）当分配给程序 4 个存储块时，缺页中断情况如下表所示：

时刻	1	2	3	4	5	6	7	8	9	10	11	12
访问页面	4	3	2	1	4	3	5	4	3	2	1	5
内存页面	4	3 4	2 3 4	1 2 3 4	4 1 2 3	3 4 1 2	5 3 4 1	4 5 3 1	3 4 5 1	2 3 4 5	1 2 3 4	5 1 2 3
缺页	+	+	+	+			+			+	+	+

缺页率为：8/12

（2）当分配给程序 4 个存储块时，缺页中断情况如下表所示：

时刻	1	2	3	4	5	6	7	8	9	10	11	12
访问页面	4	3	2	1	4	3	5	4	3	2	1	5
内存页面	4	3 4	2 3 4	1 2 3 4	4 1 2 3	3 4 1 2	5 3 4 1 2	4 5 3 1 2	3 4 5 1 2	2 3 4 5 1	1 2 3 4 5	5 1 2 3 4
缺页	+	+	+	+			+					

缺页率为 5/12

（3）以上结果说明：采用 LRU 页面置换算法的情况下，增加主存容量将降低缺页中断的次数和缺页率

六、（1）直接索引中盘块总容量为 4K*10=40KB

一次间接索引中盘块总容量为：一个地址用 4 个字节，4K 大小可以表示 1K 个地址，就是可以记录 1K 个物理盘块，则总容量为 4K*1K=4MB

二次间接索引中盘块总容量为：4K*1K*1K=4GB

三次间接索引中盘块总容量为：4K*1K*1K*1K=4TB

所以这个文件系统允许的最大文件长度为 4TB+4GB+4MB+4KB

（2）实际占用空间为 2G+2M+4K

七、参考答案

(1)红色部分是补充的代码

```
TableType TABLE;

SmokerA () {    /* the smoker processes with paper */
   CigaretteType    c;
   PaperType    p;
   TobaccoType    t;
   MatchesType    m;
   While (1)
      p= Take_Paper_From_The_Pocket_of_SmokerA();
      t=Take_Tobacco_From_ The_Table(TABLE);
      m=Take_Matches_From _The_Table (TABLE);
      c=Make_A_ Cigarette(p,t);
      Smoking(c);
   }
}
SmokerB () {
    /* the smoker processes with tobacco */
   CigaretteType    c;
```

```
    PaperType    p;
    TobaccoType    t;
    MatchesType    m;
    While (1)
      p= Take_Paper_From_ The_Table(TABLE);
      t=Take_Tobacco_From_ The_Pocket_of_SmokerB();
      m=Take_Matches_From _The_Table (TABLE);
      c=Make_A_ Cigarette(p,t);
      Smoking(c);
    }
}
SmokerC () {
     /* the smoker processes with matches */
    CigaretteType    c;
    PaperType    p;
    TobaccoType    t;
    MatchesType    m;
    While (1)
      p= Take_Paper_From_ The_Table(TABLE);
      t=Take_Tobacco_From_ The_Table (TABLE);
      m=Take_Matches_From _ The_Pocket_of_SmokerC();
      c=Make_A_ Cigarette(p,t);
      Smoking(c);
    }
}
ServerAgent() {
      /* agent process */
    PaperType    p;
    TobaccoType    t;
    MatchesType    m;
    While (1)
      ch=RandamNumberFrom_1_2_3();
      /* generate a randam number From 1,2 or 3 */
      if (ch= =1)    {
            t= Prepair_Tobacco();
            m= Prepair_Matches();
            Put__Tobacco _To_The_Table(TABLE, t);
            Put_Matches_To_ The_Table (TABLE, m);
       }
      if (ch= =2)    {
           p= Prepair_Paper();
           m= Prepair_ Matches ();
            Put_Paper _To_The_Table(TABLE, p);
            Put_Matches_To_ The_Table (TABLE, m);
```

```
        }
        if (ch= =3)    {
                p= Prepair_Paper();
                t= Prepair_Tobacco();
                Put_Paper _To_The_Table(TABLE, p);
                Put_ Tobacco _To_ The_Table (TABLE, t);
        }
}
```

(2)设置信号量 s1,s2,s3, s(初值均为 1)

信号量 mutex，初值为 1

在代码中加入 P、V 操作，见下表：

```
TableType TABLE;

SmokerA () {    /* the smoker processes with paper */
   CigaretteType    c;
   PaperType    p;
   TobaccoType    t;
   MatchesType    m;
   While (1)
      p= Take_Paper_From_The_Pocket_of_SmokerA();
      p(s1)
      p(mutex)
      t=Take_Tobacco_From_ The_Table(TABLE);
      m=Take_Matches_From _The_Table (TABLE);
      v(mutex)
      v(s)
      c=Make_A_ Cigarette(p,t);
      Smoking(c);
   }
}
SmokerB () {
    /* the smoker processes with tobacco */
   CigaretteType    c;
   PaperType    p;
   TobaccoType    t;
   MatchesType    m;
   While (1)
      t=Take_Tobacco_From_ The_Pocket_of_SmokerB();
```

```
        p(s2)
        p(mutex)
        p= Take_Paper_From_ The_Table(TABLE);
        m=Take_Matches_From _The_Table (TABLE);
        c=Make_A_ Cigarette(p,t);
        v(mutex)
        v(s)
      Smoking(c);
    }
}
SmokerC () {
     /* the smoker processes with matches */
    CigaretteType    c;
    PaperType    p;
    TobaccoType    t;
    MatchesType    m;
    While (1)
        m=Take_Matches_From _ The_Pocket_of_SmokerC();
        p(s3)
        p(mutex)
        p= Take_Paper_From_ The_Table(TABLE);
        t=Take_Tobacco_From_ The_Table (TABLE);
        c=Make_A_ Cigarette(p,t);
        v(mutex)
        v(s)
        Smoking(c);
    }
}
ServerAgent() {
        /* agent process */
    PaperType    p;
    TobaccoType    t;
    MatchesType    m;
    While (1)
        ch=RandamNumberFrom_1_2_3();
        /* generate a randam number From 1,2 or 3 */
        p(s)
        if (ch= =1)    {
                t= Prepair_Tobacco();
                m= Prepair_Matches();
                p(mutex)
                Put__Tobacco _To_The_Table(TABLE, t);
                Put_Matches_To_ The_Table (TABLE, m);
                v(mutex)
```

```
            v(s1)
        }
        if (ch= =2)    {
              p= Prepair_Paper();
              m= Prepair_ Matches ();
               p(mutex)
               Put_Paper _To_The_Table(TABLE, p);
               Put_Matches_To_ The_Table (TABLE, m);
               v(mutex)
                v(s2)
        }
        if (ch= =3)    {
               p= Prepair_Paper();
               t= Prepair_Tobacco();
               p(mutex)
               Put_Paper _To_The_Table(TABLE, p);
               Put_ Tobacco _To_ The_Table (TABLE, t);
               v(mutex)
                v(s3)
        }
    }
```