同济大学课程考核试卷（A 卷）
2024—2025 学年第一学期

命题教师签名：　　审核教师签名：

课号：101020　　课名：操作系统　　考试考查：考试

此卷选为：期中考试(　　)、期终考试(∨)、重考(　)试卷

年级_____专业_____学号_____姓名_____得分_____

一、（10 分）2 个生产者进程和 2 个消费者进程共享有界缓冲区 buffer。生产者进程执行 producer( )函数，将产生的新数据，整数 item 存入缓冲区，in 指针指向的单元；消费者进程执行 consumer( )函数，取用、处理 out 指针指向的数据。以下是正确的代码。

```
int in = out =0;
int buffer[N];
Semaphore full, empty, mutex;
full.value  = (        );
empty.value = (        );
mutex.value = (        );

parbegin { producer1, producer2,
           consumer1, comsumer2 }
```

```
void  producer()
{
  int item;
  while(true)
  {
      item = produceItem();
      p(empty);
      p(mutex);
      buffer[in] = item;
      in=(in+1) mod N;
      v(mutex);
      v(full);
  }
}
```

```
void  consumer()
{
  int item;
  while(true)
  {
      p(full);
      p(mutex);
      item=buffer[out];
      out=(out+1) mod N;
      v(mutex);
      v(empty);
      consumeItem(item);
  }
}
```

1、 请在代码中填空，为所有信号量赋正确的初值。

full 的初值是 0，empty 的初值是 N，mutex 的初值是 1

2、删除 producer( )和 consumer( )函数中的 p(mutex)和 v(mutex) 之后，2 个生产者进程并发，有可能产生（ A ）现象；2 个消费者进程并发，有可能产生（ B ）现象。产生数据丢失现象时，所丢失的是（ C ）。

A、数据丢失　　B、数据重复　　C、先写入的数据　　D、后写入的数据

3、题干所示代码中，生产者互换 P(empty)和 P(mutex)，消费者互换 P(full)和 P(mutex)。系统有可能会出现死锁。试分析系统可能出现死锁的所有情况。

（1）缓存满时，生产者进程执行。（2）缓存空时，消费者进程执行。

4、名词解释：死锁。

在一个进程集合中,每个进程都因等待永远不会发生的事件而阻塞，称这种系统状态为死锁。

5、列出死锁产生的 4 个必要条件。

互斥条件、不可剥夺条件，请求保持条件，循环等待条件。

6、无论是连续内存管理方式还是请求调页式系统，内存不足系统会抖动。抖动发生时，无法执行有效运算。请问，抖动是死锁吗？用第 4、5 小问考察的知识解释你的判断。

抖动不是死锁。因为，进程竞争的 CPU 和内存是可剥夺资源。死锁必要条件 2 不满足。

二、（6 分）快餐店有 30 个座位。顾客光临时，若座位全满，转身离开。请问（1）下面的 2 个解正确吗？（2）如果有错，分别错在哪里。（3）请在代码中修正解 1 的错误。

解 1

```
semaphore mutex;
mutex.value = 1;
int seats = 30;

customer( )
{
    p(mutex);
    if(seats == 0)
        return;
    else
        seats--;
    v(mutex);
    eating( );
    p(mutex);
        seats++;
    v(mutex);
}
```

{ v(mutex); return; }

解 2

```
semaphore seats;
seats.value = 30;
customer( )
{
    p(seats);
    eating( );
    v(seats);
}
```

解 1 是错误的。顾客离开快餐店时忘记解锁互斥信号量 mutex。状态永远不会更新，系统就死了。
解 2 也是错误的。快餐店满的时候，顾客排队等待而不是转身离开。
解 1 修正参见代码中的注释。

三、（9 分）系统中有五个进程 {P0,P1,P2,P3,P4}和三类资源{A,B,C}，资源总数为{10,5,7}。T 时刻空闲资源数量是{3,3,2}，资源分配状态如下表所示。请问（1）T 时刻，系统状态安全嘛？如果安全，给出一个安全序列（2）P0 发出资源请求 Request0＝{3, 3, 0}，系统是否可以满足此次资源申请。要求写出判断依据和计算过程。（3）回到 T 时刻的资源分配状态，请随意设计一个能够被系统接纳的资源申请。给出系统接纳该请求的理由。

（1）

进程	Max A B C	Allocation A B C	Need A B C
P0	7 5 3	0 1 0	7 4 3
P1	3 2 2	2 0 0	1 2 2
P2	9 0 2	3 0 2	6 0 0
P3	2 2 2	2 1 1	0 1 1
P4	4 3 3	0 0 2	4 3 1

安全。
安全序列：P1、P3、P0、P2、P4

（2）

进程	Max A B C	Allocation A B C	Need A B C

P0	7 5 3	0 1 0（340）	7 4 3（413）
P1	3 2 2	2 0 0	1 2 2
P2	9 0 2	3 0 2	6 0 0
P3	2 2 2	2 1 1	0 1 1
P4	4 3 3	0 0 2	4 3 1

系统会拒绝此次资源分配申请。理由是，若满足 P0 的这次资源申请，资源分配状态如上表所示。空闲资源数量下降：{0, 0, 2}不再能够满足任何进程的资源需求，会导致系统进入不安全状态。

（3）

进程	Max A B C	Allocation A B C	Need A B C
P0	7 5 3	0 1 0	7 4 3
P1	3 2 2	2 0 0	1 2 2
P2	9 0 2	3 0 2	6 0 0
P3	2 2 2	2 1 1（221）	0 1 1（001）
P4	4 3 3	0 0 2	4 3 1

P3 发出资源请求 Request0 = {0, 1, 0}。系统可以接纳这个资源访问请求。理由是，满足此次资源申请后，空闲资源数量变成{3,2,2}。存在安全序列 P1、P3、P0、P2、P4，系统不会因为此次资源分配进入不安全状态。

四、（8 分）某一级页表内存管理系统，页的大小是 4K 字节。现运行进程页表内容如下图所示。

逻辑页号	物理页框号
0	7
1	5
……	……

请问（1）逻辑地址 0x1500 对应的物理地址是（ 0x5500 ）。（2）已知访问主存单元平均耗时 200ns，若 CPU 不设 cache 单元，逻辑地址访存耗时是（400）ns。（3）若 CPU 设置 cache 单元。TLB 命中率是 99%，从 TLB 中取 PTE 的平均耗时是 10ns；指令、数据缓存命中率是 96%，从缓存中取指令/数据的平均耗时是 10ns。则逻辑地址访存平均耗时是（30）ns。 T 时刻，CPU 依次、先后访问逻辑地址 0x1500 和 0x1504，请问访问内存单元 0x1504 的耗时是（20）ns。（0x1500 和 0x1504 一定在一个 cache 行。所以 TLB 命中，指令数据也命中，均无需访问内存）。

计算：EAT(Effective Access Time，内存有效访问时间)
逻辑地址访存耗时

- EAT = 地址映射时间（访问页表时间） + 访问指令/数据时间
  = TLB访问时间 + TLB不命中的概率 * 主存访问页表耗时
  + 缓存访问时间 + 缓存不命中的概率 * 主存访问耗时

二级页表。PTE没有命中TLB的话，MMU就。。。先去主存找PDE，然后去主存找PTE并缓存，最后去主存找数据

- 练习3：一级页表。

某一级页表内存管理系统，TLB 命中率是99%。从TLB中取PTE的平均耗时是10ns。指令、数据缓存命中率是95%。从缓存中取指令/数据的平均耗时是10ns。访问主存单元平均耗时200ns (取PTE、访问指令/数据相同)。求该系统内存有效访问时间。

EAT = （10+200*0.01 ） + （10+200*0.05） = 32ns

不设cache，EAT = 400ns。使用 cache，访存提速10倍以上。

操作系统 电信学院计算机系 邓蓉 51

五、（14 分）某一级页表内存管理系统，位示图表示为 32 位整形数组 bitmap，0#元素、0#bit 表示 0#物理页框的分配情况，1 表示已分配，0 表示未分配。堆栈位于虚空间最高端，需要扩展堆栈时，系统为现运行进程追加 1 个物理页框。T 时刻，现运行进程 PA 触发堆栈扩展操作，页表如下图所示，空白处页表项为空。已知，bitmap[0]=bitmap[1]=……= bitmap[15]=0xffffffff，bitmap[16]=0x0fff。请问

（1）系统为该进程追加的物理页框号是（ 0x20C ）写 16 进制数。

解题步骤：找第一个 0bit。

bitmap[0]=0xffffffff，物理页框 0~31 分配出去了。

bitmap[1]=0xffffffff，物理页框 32~63 分配出去了。

……

所以，第一个 0bit，对应物理页框 16*32+3*4 = 0x20C。

页式存储管理系统，一个一个物理页框分配。分配操作伪代码：

```
int bitmap[];
int i;
while( bitmap[i++] = = 0xffffffff )
    ;
```

找到 bitmap[i] 中的第一个 0

（2）简述堆栈扩展过程；修改页表，展示堆栈扩展后的页表状态。

分配 1 个空闲的物理页框，0x20C

PCB 找到内存描述符，找到 stack 长度（1 个页面）。

填页表，倒数第 2 个 PTE（已有堆栈上方的那个 PTE）

base = 0x20C，prot = U RW

base	prot
0x203	U RO
0x205	U RW
0x206	U RW

0x20C U RW

（3）以此题为背景，简述页式存储管理方式的优点。

堆栈扩展方便，已有进程图像不需要移动。

进程图像按页离散存放，物理内存利用率高。

（4）一级页表内存管理系统有 2 个主要的缺点。

- 页表内存消耗大，且必需占据连续内存单元 ，克服这个缺点，可以使用 二级页表（多级页表） 技术。
- 地址映射需要读内存中的页表，访存速度减半 ，克服这个缺点，可以使用 在 CPU 内部增设 TLB 单元，用以缓存最近用过的 PTE 技术。

（5）栈为什么要放在虚空间的最高端？

栈在虚空间中不可移动（因为栈帧里有调用者基地址）。

放在最高端，可以与堆最大限度共用空闲虚空间。

六、（9 分）某请求调页式虚拟存储器。逻辑地址空间为 64KB，页面长 4KB。系统规定每个进程只能使用 3 个物理（内存）页框；若缺页，用被淘汰的逻辑页面所占据的页框装所缺的逻辑页面。系统采用 LRU 置换算法。时刻 256，现运行进程 P 的页表如下所示：请给出下列逻辑地址对应的物理地址：（a）0x422A（写）（b）0x1234（写）。要求写出地址映射过程和更新后的页表。

逻辑页号	物理页框号	存在位 P	装入时刻	上次访问时刻	修改位 M	RW 1, 可读可写 0，只读	外存起始扇区号
0	0x100	1	130	250	1	0	0x2000
1	-	0	-	-	-	0	0x2008
2	0x102	1	230	240	1	1	0x2010
3	0x106	1	160	170	1	1	0x2018
4	-	0	-	-	-	1	0x2020

答：

（a） 写操作。地址映射过程：

分割逻辑地址： 4#逻辑页，页内偏移量：0x22A

内存保护操作： 4#PTE 可写，通过

地址映射操作： 存在位 P==0，抛出缺页异常。

异常处理程序：

（1） 物理页框用完了，淘汰 LRU 3#页面。修改位是 1，写回磁盘，IO 参数：
内存首地址 0x106000，外存起始扇区号 0x2018，写操作，数据量 4096 字节。

（2） 从磁盘调入 4#逻辑页。IO 参数：
内存首地址 0x106000，外存起始扇区号 0x2020，读操作，数据量 4096 字节。

（3） 修改 PTE，如下表所示。异常处理程序返回。

逻辑页号	物理页框号	存在位 P	装入时刻	上次访问时刻	修改位 M	RW 1，可写	外存起始扇区号
0	0x100	1	130	250	1	0	0x2000
1	-	0	-	-	-	0	0x2008
2	0x102	1	230	240	1	1	0x2010
3	0x106	0	-	-	0	1	0x2018
4	0x106	1	256	-	-	1	0x2020

重新执行地址映射操作，存在位 P==1，生成物理地址 0x10622A。更新 PTE。

逻辑页号	物理页框号	存在位 P	装入时刻	上次访问时刻	修改位 M	RW 1，可写	外存起始扇区号
0	0x100	1	130	250	1	0	0x2000
1	-	0	-	-	-	0	0x2008
2	0x102	1	230	240	1	1	0x2010
3	0x106	0	-	-	0	1	0x2018
4	0x106	1	256	256	1	1	0x2020

（b） 0x1234（写）

分割逻辑地址： 1#逻辑页，页内偏移量：0x234

内存保护操作： 1#PTE 不可写，通不过，抛出缺页异常。现运行进程被杀死。

页表没有更新。

七、（6 分）某机械硬盘，磁头当前位于磁道 143，向小号磁道方向移动。磁道访问请求序列为：

86，147，91，177，94，150，102，175，130

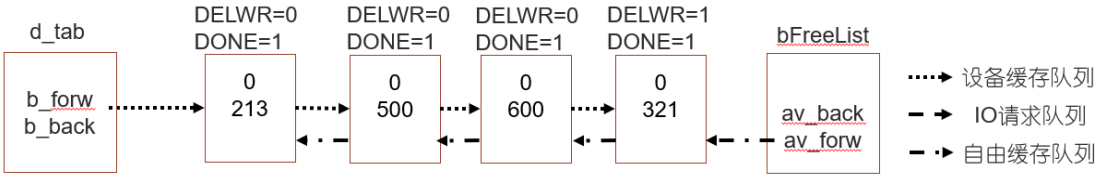
（1）请写出电梯调度算法处理上述请求产生的服务序列。

143 130 102 94 91 86 147 150 175 177

（2）给出磁头移动总磁道数最少的请求服务序列。

143 147 150 175 177 130 102 94 91 86

八、（6 分）T 时刻，磁盘高速缓存状态如下所示。PA、PB、PC 进程同时发起对 660#块，偏移量为 0 的字节的访问请求。PB 先写。随后 PA，PC 先后发出读请求。请回答以下问题。



（1） 画出读写操作全部完成后，磁盘高速缓存的状态。

设备缓存队列 660（DELWR=1，DONE=1） → 213 → 500 → 321（DELWR=0，DONE=1）

自由缓存队列 500 → 213 → 321 → 660

（2） PA、PC 均能读到 PB 写入的新数据。正确否？一定是 PA 先于 PC 读到目标数据吗？

是的。

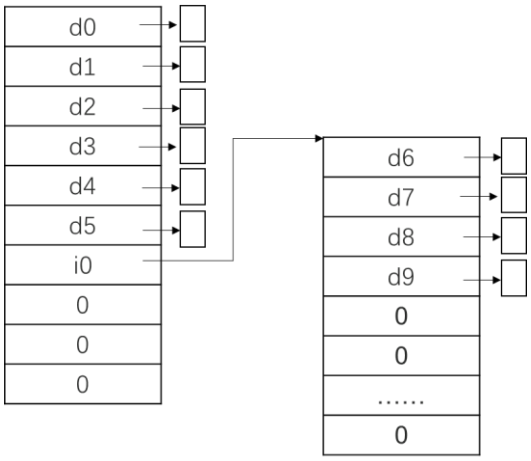
PA 不一定先于 PC 读到目标数据。PC 有可能先调度到。

九、（10 分）Unix V6++系统，磁盘数据块 512 字节，文件的 d_addr 数组管理 6 个直接块，2 个一次索引块和 2 个二次索引块。本题读入指磁盘 IO，假定打开时 fileA 所有数据块缓存不命中。请问：

（1）系统需要为一个长度为 5120 字节的文件 fileA 分配多少磁盘存储资源？（2 分）

答：一个目录项，一个 DiskInode，10 个数据块和 1 个索引块。

（2）填写 fileA 的 d_addr 数组并画出其混合索引结构，分配给该文件的磁盘数据块号（扇区号）可以自定。 （2 分）



（3）现运行进程成功打开该文件后，访问偏移量是 1000 的字节需要读入____1____个磁盘数据块。访问偏移量是 4096 的字节需要读入____2____个磁盘数据块。随后，访问偏移量是 4196 的字节需要读入___0__个磁盘数据块，访问偏移量是 5000 的字节需要读入___1___个磁盘数据块。（4 分）

（4）简述混合索引这种文件物理结构的优缺点。（2 分）

优点（1）索引结构灵活、扩充方便：小文件无需索引块，大型、巨型文件索引块随文件长度增加逐步分配，无需改动已有结构（2）支持随机访问，工作效率高，访问文件首部 6 个数据块无需索引块，速度快。

缺点：对插入、删除操作不友好。文件内部插入、删除数据块，索引结构改动非常大。

十、（12 分）某 Unix V6 文件卷，0#、1#扇区是超级块，2#~30#扇区是 inode 区，每个 DiskInode 64 字节。已知文件 abc 尺寸 512 字节，是文件卷中的 12#文件；下图所示程序运行时，内存中没有需要引用的磁盘数据。

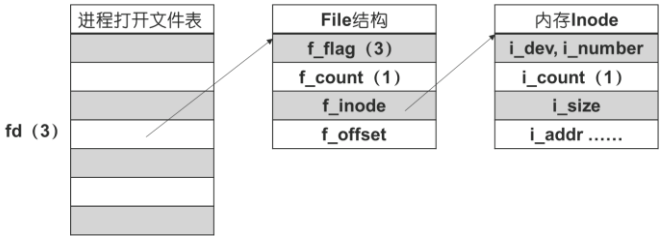
```
main( )
{
    char data[100];
```

```
L1:    int fd = open("abc", O_RDWR);   // 以可读可写方式打开文件
L2:    int count = read (fd, data, 100);
L3:    seek(fd, 512, 0);
       write (fd, data, count);
       close(fd);
    }
```

（1） 这段程序完成什么功能？ 将文件首部的 100 个字节追加写至文件末尾 （2 分）。

（2） 本例，open 系统调用会引发 IO 操作吗？ 会 。如果会，进程会从磁盘读入 父目录文件的数据块 和 abc 文件的 DiskInode （2 分）。

（3） 语句 L1、L2、L3 执行完毕后，fd 的值是 3 ，count 的值是 100 ，绘制 seek 系统调用执行完毕后，这个进程的打开文件结构。



（4） 简述本例，write 系统调用的执行过程。

答：f_offset==512，分 1 次写入。初始化 IO 参数：

m_offset = f_offset = 512，m_base = data，m_count = 100

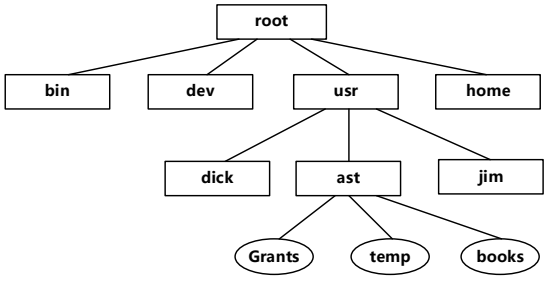
- 当前逻辑块 512/512 = 1，块内偏移量 512%512 = 0。混合索引表中 1#逻辑块的物理块号是 0，系统为其分配新数据块 new，登记：i_addr[1] = new。写入 100 字节，未写满，需先读，启动 IO 将磁盘数据块 new 同步读入分配给它的缓存块。data[0]~ data[99]写入该缓存块。未写至底部，延迟写：B_DELWR=1，B_DONE=1，释放缓存块。（1 分）
- m_offset = 612，m_count = 0，write 系统调用结束，返回实际写入文件的字节数 100。递增 f_offset=612；写操作导致文件长度增加，i_size = 612。

（5） 如果 close 系统调用执行完毕后系统断电，描述文件系统可能出现的故障。以此情景为背景，结合 8.3 小题，论缓存技术在改善文件系统性能方面起到的作用和付出的代价（3 分）。

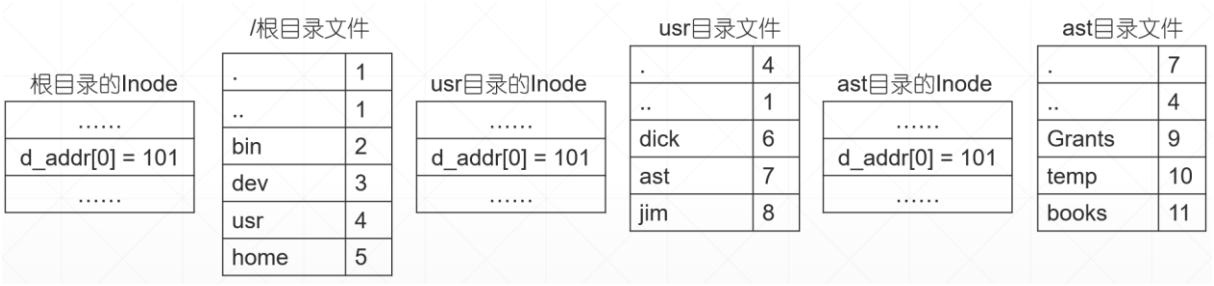
答：1#逻辑块中的新数据未写回磁盘，断电后丢失。

使用缓存技术可以提高文件系统的访问速度，减少磁盘 IO 次数，提高磁盘吞吐率。但可能会丢数据，降低文件系统的一致性。

十一、（6 分）某 UNIX V6++系统中，文件系统的目录结构如下图所示：



（1）（3 分）根据上述目录结构，补充完整下面几个目录文件的内容（INode 节点号请任意指定。后续问题回答中请沿用相同的 INode 号）。

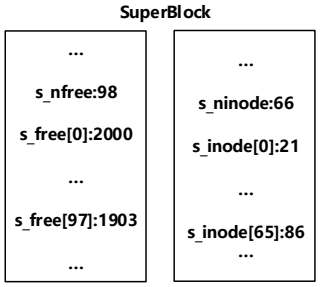


特殊目录项，1 分

目录号和盘块号弄混了，扣 1 分

目录结构，1 分

（2）（3 分）T0 时刻，SuperBlock 中空闲 INode 栈和空闲盘块栈中记录的内容如下图所示。系统删除文件/usr/ast/books。已知，books 文件长 1500 个字节，占用磁盘上的 200#盘块（0#逻辑块）、201#盘块（1#逻辑块）和 205#盘块（2#逻辑块）。请以图示方式说明该操作结束后，SuperBlock 中空闲 INode 栈和空闲盘块栈的变化情况。各目录文件的内容是否有更新？如果有，请图示说明。



（3）（3 分）随后的 T1 时刻，系统创建一个新文件：/usr/ast/new，并向其中写入字符串“Hello World!\0”，关闭该文件。已知，（T0,T1]时间区间，系统不曾对文件系统实施任何操作。请绘制出目录文件 ast 的内容，并给出 new 文件的索引结构。