

南开大学信息技术科学学院本科生 2011-2012 年度第一学期操作系统原理课程期末试卷 (A 卷)

专业_____年级_____姓名_____学号_____成绩_____

得 分

一、简答题（本题共 30 分，每题 6 分，必做）

草稿区

1. 某些进程状态集中存在“新建（New）”、“挂起（Suspend）”和“退出（Exit）”三种状态，请简要解释这三种状态的作用。

共三个答题点，每个答题点 2 分，不回答任何一个答题点扣 2 分。
1.新建：进程的数据结构（PCB）已经被创建，但是进程尚未被插入到就绪队列中，此时进程的指令和数据等映像（Image）并未被加载进入内存。（注意，如果回答中提到指令和数据被加载进入内存，扣 1 分）
2.挂起：挂起并不是阻塞态，如 Windows 的进程挂起态，挂起是指将就绪进程或阻塞进程的内存数据交换到磁盘缓冲区，藉此腾出更多内存供正在运行的进程使用。（注意，凡是把挂起解释为阻塞的，扣 2 分）
3.退出：进程已经完成自身的运行，但是还有其他进程/线程引用或共享着该进程的指令或数据，则该进程无法立即释放，必须等待引用计数为零才能释放。

2. 何谓优先级翻转问题？能否说明一下用什么方法可以有效解决优先级翻转问题？

共两个答题点，每个答题点 3 分，不回答任何一个答题点扣 3 分。
1.优先级翻转问题：在采用优先级的进程调度时，针对进程之间同步互斥的 IPC 问题，如果低优先级的进程已经进入了临界区，则会阻塞临界区外的高优先级进程，但是高优先级进程一直占用 CPU，导致低优先级的进程无法运行并退出临界区，两个进程形成“僵死”状态，这就叫优先级翻转。（注意，优先级调度、进程同步互斥是该答题点的核心内容，如果不强调优先级调度，则扣 1 分）
2.解决方法：需要使用更为合理的进程同步互斥机制，包括信号量机制、消息机制和管程机制等。（注意，这三种方案回答出一种就给 3 分）

3. 动态多道分区内存管理与分段式虚拟内存管理的主要区别是什么？

共三个答题点，每个答题点 2 分，不回答任何一个答题点扣 2 分。
1.虚拟存储：动态多道分区必须把进程的指令和数据一次性的加载进入内存，而分段式虚拟存储管理可通过内存替换的方式加载指令和数据，所以分段式存储能实现小内存大程序的运行。
2.指令和数据空间的分离：动态多道分区不能实现进程的指令与数据空间分离，分段式存储管理可以实现。
3.进程间的内存共享：动态多道分区不能实现进程之间的指令和数据共享，分段式虚拟存储却可以实现方便的共享。

简答题评分细则：
1. 按答题点给分，只要和参考答案的意义类似就给予该答题点分数。
2. 如果答案中出现与扣分说明类似的内容，则按照扣分规则进行扣分。
3. 不以答案的文字多少评判答案的严谨和正确性，符合答题点的参考答案要求即可。

4. 造成进程死锁的原因是什么？

共四个答题点，每个答题点 1.5 分，不回答任何一个答题点扣 1.5 分。每个答题点只要含义相似即给分。

- 1.资源的独占性条件：I/O 资源为独占式资源，被任一进程占用后不能被其他进程共享。
- 2.资源的不可剥夺条件：当一个进程占用了某个 I/O 资源后，操作系统无法剥夺该进程对该 I/O 资源的控制权，也就无法把该 I/O 资源分配给其他等待该 I/O 资源的进程。
- 3.进程的保持并等待条件：当进程占用了某种 I/O 资源后，还需要其他 I/O 资源才能运行完成，当无法获得其他的 I/O 资源时，该进程依然保持对既有 I/O 资源的控制权，并等待其他 I/O 资源，由此造成该进程的阻塞。
- 4.进程间的循环等待条件：多个进程之间构成了循环等待链条，任何一个进程所需要的 I/O 资源都被其他进程保持，由此形成了一个封闭的进程集合，这个进程集合被称为死锁进程集合。

5. 在文件系统中，用什么方法可以检查磁盘扇区是否失效（亦即磁盘扇区的一致性检查）？

共两个答题点，每个答题点 3 分，不回答任何一个答题点扣 3 分。每个答题点只要含义相似即给分。

- 1.磁盘块的两种状态：每个磁盘块都处于两种可能状态中，一种是使用态，一种是空闲态，正常情况下，一个磁盘块要么为使用态，要么为空闲态，用一位二进制数来标识，则“10”和“01”都表示磁盘块的状态正常。
 - 2.一致性检查：操作系统建立磁盘块的使用态列表和空闲态列表，当一个磁盘块在这两个列表中的状态不正常时，意味着不一致，例如“00”意味着块丢失，“02”意味着空闲块重复，“20”意味着使用块重复等。
- 教材中的图 5-18 描述了这种检查机制。

得 分

二、编程计算题（本题共 5 小题，共计 45 分，选做 4 题，多做不得分）

草稿区

- ✧ 请在下面的表格中指定答题顺序，在对应的分值下列明题号。每格只许列出一个题号，否则做无效处理。
- ✧ 下表中必须写明所有题目的题号，如果填写不完全，视为不指定答题顺序。
- ✧ 如填写内容无效或者不填写表格，则按照默认的题面分值评分

第一题（15 分）	第二题（12 分）	第三题（10 分）	第四题（8 分）

6. 某操作系统使用分页式虚拟存储管理机制进行内存管理，该系统的 TLB 保存了最近使用的 3 个页表项，某用户进程被分配了 4 个页帧，运行过程中的页面访问序列为 1,3,2,1,3,4,5,1,2,6,8,10,7,1,9,2,3,5,4,8,9,1，系统使用 NFU 页面替换算法，请问在该进程运行过程中，发生缺页中断的次数是多少？有哪些页面被替换过？TLB 的命中率是多少？

（本题默认分值：12 分，简要说明计算过程，列出最终计算结果即可）

标准分为 10 分，参考答案以标准分的比例给分或扣分。

1. 本题未说明系统是否采用 Pre-paging 还是 paging on demand 技术，默认采用 paging on demand 技术，这就意味着初始的 4 个页帧是空的，由此增加了 4 次缺页中断。当 4 个页帧填满后，进入到 NFU 的页面替换算法流程，共计发生缺页中断 14 次。所以回答缺页中断次数为 14 次或 18 次都是正确的，该问题回答正确，给题目分数的 30%。
2. 一共发生 14 次页面替换，被替换的页面依次为（前面表示被替换页号，后面表示替换完的页号）：
2-5；3-2；4-6；5-8；1-10；2-7；6-1；8-9；10-2；7-3；1-5；9-4；2-8；3-9；4-1；该问题回答正确，给题目分数的 42%，亦即每个替换占题目分值的 3%。
3. 如果按照 paging on demand 技术，TLB 的初始值为空，因此前 3 次的页面访问都是未命中的，如果按照 pre-paging 技术，则 TBL 的前三次页面访问均为命中，因此在 22 次页面访问中，TLB 的命中次数分别为 2 或者 5，这两个结果都是正确的，对应的命中率为 $2/22=9.09\%$ 和 $5/22=22.7\%$ 。该问题占题目分值的 28%。

1	3	2	1	3	4	5	1	2	6	8	10	7	1	9	2	3	5	4	8	9	1	1	1	3	2	1	3	4	5	1	2	6	8	10	7	1	9	2	3	5	4	8	
1	1	1	1	1	1	1	1	1	1	1	10	10	10	10	2	2	2	2	8	8	8		3	3	2	1	3	4	5	1	2	6	8	10	7	1	9	2	3	5	4	8	9
	3	3	3	3	3	3	3	2	2	2	2	7	7	7	7	3	3	3	3	9	9			2	1	3	4	5	1	2	6	8	10	7	1	9	2	3	5	4	8	9	1
		2	2	2	2	5	5	5	5	8	8	8	8	9	9	9	9	4	4	4	1	N	N	N	Y	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	
					4	4	4	4	6	6	6	6	1	1	1	1	5	5	5	5																							
N	N	N	Y	Y	N	N	Y	N	N	N	N	N	N	N	Y	N	N	N	N	N																							

页面替换表

TLB 命中表（均按 Pre-paging 技术计算）

7. 考虑具有如下特征的共享资源：1) 资源总数为 7，序号为 R_1-R_7 ；2) 进程总数为 7，序号为 P_1-P_7 ；对进程 P_i 而言，必须同时获得资源 R_i 和 $R_{(i-1) \% 7}$ 才能正常运行；3) 解决方案必须保证最多 3 个进程能同时运行。请用信号量的方法来描述并解决这个问题。
- 1) 请说明你所定义的信号量及其他数据变量的逻辑含义，注意，应记录资源被占用数、等待资源进程数；
- 2) 使用伪代码描述资源申请及资源使用的同步互斥处理过程。（本题默认分值：15 分）

标准分为 10 分，参考答案以标准分的比例给分或扣分。
本题为标准的哲学家就餐问题，按照讲义中的哲学家就餐问题的信号量解决机制就可以实现，标准答案如下：
1. 信号量及数据变量定义：回答正确给题目分值的 20%，如果漏掉了状态定义项 (Thinking, Hungry, Eating)，扣除 5%

```
typedef int semph
#define N      5
#define LEFT(i)  (i + N - 1) % N
#define RIGHT(i) (i + 1) % N
#define THINKING 0
#define HUNGRY 1
#define EATING 2
int state[N];
semph mutex = 1;
semph s[N];
```

2. 伪代码实现：共有四个函数，答案中如果涉及到了全部功能，则同样给分，如果逻辑错误，酌情扣分，该问占题目分值的 80%。

```
Philosopher
void philosopher(int i)
{
    While(TRUE){
        think ();
        take_ChS(i);
        eat();
        put_ChS(i);}
}
```

```
take_chs
void take_chs (int i)
{
    P(mutex);
    state[i] = HUNGRY;
    test(i);
    V(mutex);
    P(s[i]);
}
```

```
Put_chs
void put_forks (int i)
{
    P(mutex);
    state[i] = THINKING;
    test(LEFT(i));
    test(RIGHT(i));
    V(mutex);
}
```

```
Test
void test (int i)
{
    if((state[i] == HUNGRY)
        && (state[LEFT(i)] != EATING)
        && (state[RIGHT(i)] != EATING))
    {
        state[i] = EATING;
        V(s[i]);
    }
}
```

8. 对采用固定分区的内存管理机制而言，其分区个数被定义为系统的并发度。假设所有进程都能被全部加载进入内存
对以下几类进程，欲使得 CPU 利用率达到 100%，请简要计算其最佳的并发度是多少？

- 1) 进程类型 1：其 80% 的运行时间为 CPU 计算时间，其余为 I/O 处理时间；
- 2) 进程类型 2：CPU 计算时间和 I/O 处理时间各为 50%；
- 3) 进程类型 3：其 80% 的运行时间为 I/O 处理时间，其余为 CPU 计算时间。

（本题默认分值：8 分，必须给出简要计算过程和计算结果）

标准分为 10 分，参考答案以标准分的比例给分或扣分。

1. 本题为三问，每个问题的解题逻辑相同，说明解题逻辑占题目分值的 10%，每一问的答案回答准确，占题目分值的 30%。
2. 解题逻辑：设进程间切换的时间忽略不计，任意两个进程之间无依赖关系，则每个进程的 CPU 计算时间可理解为 CPU 计算时间的概率，分区个数等于进程个数，因此对于 N 个分区而言，其 CPU 利用率的计算公式（本质上为数学期望）为 $\sum_{i=1}^N P_i$ ，其中 P_i 是第 i 个进程的 CPU 使用概率（亦即 CPU 运行时间比例），令该公式的结果为 100%，在已知 P_i 的情况下，就可以计算出 N 为多少比较合适。
3. 第一问：如果所有进程的 CPU 计算时间比例为 80%，亦即 P_i 为 0.8，则 N 为 1.25 就能满足 CPU 利用率 100%，但实际情况中 N 必须为整数，所以设 N 为 2 最佳；
4. 第二问：如果所有进程的 CPU 计算时间比例为 50%，亦即 P_i 为 0.5，则 N 为 2 最佳；
5. 第三问：如果所有进程的 CPU 计算时间比例为 20%，亦即 P_i 为 0.2，则 N 为 5 最佳；

解释：在固定分区的内存管理机制中，一般都需要系统管理员首先对进程的类型进行区分，通过动态设定分区个数来满足不同类型进程的需求，这在进程调度中属于典型的“外部调度”，也叫“宏观调度”，通过设定合理的分区数，才能执行高效率的批处理业务。

9. 死锁问题解决：一个系统中有 4 个进程和 3 种可分配资源，当前分配资源和最大需求如下表所示：

	已分配资源	最大需求量	资源总数
	(R_1, R_2, R_3)	(R_1, R_2, R_3)	(R_1, R_2, R_3)
进程 A	1 0 0	3 2 2	9 3 6
进程 B	4 1 1	6 1 3	
进程 C	2 1 1	3 1 4	
进程 D	0 0 2	4 2 2	

请回答以下问题：

1) 进程 A 申请资源 (1,0,1) 是否允许？2) 进程 B 申请资源 (1,0,1) 是否运行？3) 如果进程 A 和 B 的资源请求同时得到满足，系统是否处于安全状态？注意：问题 1) 和问题 2) 相互独立的，请提供处理思路，并计算安全路径。

(本题默认分值：10 分)

标准分为 10 分，参考答案以标准分的比例给分或扣分。

1. 本题为三问，其中第一问和第二问各占题目分值的 40%，第三问占题目分值的 20%。
2. 首先计算出系统目前可用资源总数为 (2,1,2)。
3. 根据各个进程的已分配资源和最大需求资源，可计算出进程 A 尚需要资源 (2,2,2)，进程 B 尚需要资源 (2,0,2)，进程 C 尚需要资源 (1,0,3)，进程 D 尚需要资源 (4,2,0)；
4. 第一问：进程 A 申请 (1,0,1)，如果分配完成，则剩余可用资源为 (1,1,1)，无法满足任何进程运行完成的需求，不可以分配。
5. 第二问：进程 B 申请 (1,0,1)，如果分配完成，则剩余可用资源为 (1,1,1)，可以满足进程 B 的剩余资源需求，进而获得安全路径，可以。
6. 第三问：如果 A 和 B 的资源请求都分配，则剩余可用资源为 (0,1, 0)，无法满足任何进程的剩余资源需求，不可以分配。

10. I/O 计算问题：在某个操作系统环境中，存在一个字符型 I/O 设备，某用户进程需要向该设备发送一个长度为 8 的字符串，假设用户进程占用 CPU 发送字符的准备时间为 5ns/次，CPU 和内存与 I/O 设备进行一次通信（包括收发双工通信）的时间为 20ns，I/O 设备处理一个字符的时间为 50ns，CPU 处理 I/O 中断的时间为 10ns/次，其他时间不计。

以该用户进程的执行过程为参照，请回答以下问题：

- 1) 若 CPU 与 I/O 设备之间采用忙等待方式进行通信，CPU 利用率是多少？
- 2) 若 CPU 与 I/O 设备之间采用中断方式进行通信，CPU 利用率是多少？
- 3) 若 CPU 与 I/O 设备之间采用 DMA 方式进行通信，CPU 利用率是多少？

（本题默认分值：15 分，请给出计算分析的思路、过程和结果）

标准分为 10 分，参考答案以标准分的比例给分或扣分。

1. 本题为三问，第一问占题目总分值的 40%，第二和第三问各占题目分值的 30%。

2. 第一问：忙等待方式的工作流程：

（1）准备字符等待发送（2）CPU 不停询问 IO 设备是否准备就绪（3）发送和接受一个字符（4）IO 设备进行字符处理处理，处理完毕，跳转到（3）

依此思路，得到：

准备时间：5ns，后面的 7 次在 IO 设备处理阶段完成，故不计入

CPU 和 IO 通信为 8 次，每次通信时间为：20ns，一次发送，CPU 通过忙等待测试，就可以判断可以进行后续发送了。I/O 设备的处理时间为：50ns*8 次。那么，总共的时间为：5+8*20+50*8 = 565 ns 。CPU 浪费的时间为 I/O 处理字符过程中的忙等待时间，所以 CPU 的利用率为：1- (50*8)/565 = 29.2%

3. 第二问：中断处理的流程

中断控制方式进行工作的流程：

（1）CPU 准备发送字符（2）与设备通信，向设备申请 IO 资源（3）设备准备就绪，发送中断（4）CPU 发送字符，设备接收（5）CPU 处理其它工作，设备处理字符（6）设备发送中断，提示处理完毕，转（4），直到发送完毕。

CPU 准备字符的时间：5ns，同样只需要准备最开始的那次；CPU 和设备通信，8 次为通信传送字符；IO 中断次数为：8 次，每次为要求 CPU 进行后续字符传递工作；IO 设备处理字符时间：8 次，每次 50ns

那么总共的时间：(5+8*20+8*10+8*50) ns = 645

其中浪费的时间是中断处理的时间，其它时间，CPU 都在处理运算：

$$1 - 80/645 = 87.6\%$$

4. 第三问：DMA 方式

CPU 向 DMA 发送命令，然后 CPU 将数据拷贝到对应的缓冲中去，响应基本的 DMA 请求，让出总线，DMA 自己和外设进行数据的收发，结束后 DMA。CPU 准备时间为 5ns，中断处理时间为 10ns，所以 CPU 的总计消耗时间为 15ns，但 CPU 的利用率却接近 100%。

得 分

三、系统分析题（本题共 3 小题，共计 25 分，选做 2 题，多做题目不得分）

✧

✧ 请在下面的表格中指定答题顺序，在对应的分值下列明题号。每格只许列出一个题号，否则做无效处理。

✧ 下表中必须写明所有题目的题号，如果填写不完全，视为不指定答题顺序。

✧ 如填写内容无效或者不填写表格，则按照默认的题面分值评分

第一题（15 分）	第二题（10 分）

11. 进程通信与 I/O 系统设计：物联网已经成为蓬勃发展的产业，其核心技术为无线传感器，设在某应用环境下，存在如下需求：一个传感装置需要将数据通过总线写入到缓冲区中，缓冲区写满后才能被处理进程读取，处理进程读完缓冲区的全部内容后，才能将新的数据写入到缓冲区中；传感装置写入数据的时间间隔为 100ns，处理进程读完缓冲区的时间为 150ns；同时存在两个处理进程需要读取缓冲区，这两个处理进程之间的时延为 20ns。

由于多处理进程之间的时延以及缓冲区读写速度的差异，系统面临着丢失传感数据的风险。

请回答以下问题：

- 1) 请尝试设计一种合理的 I/O 缓冲体系，能够确保传感数据不丢失，保证处理进程能够完整读取数据；
- 2) 请尝试设计一种合理的进程通信机制，使得传感数据采集进程与两个处理进程能和谐稳定的运行；

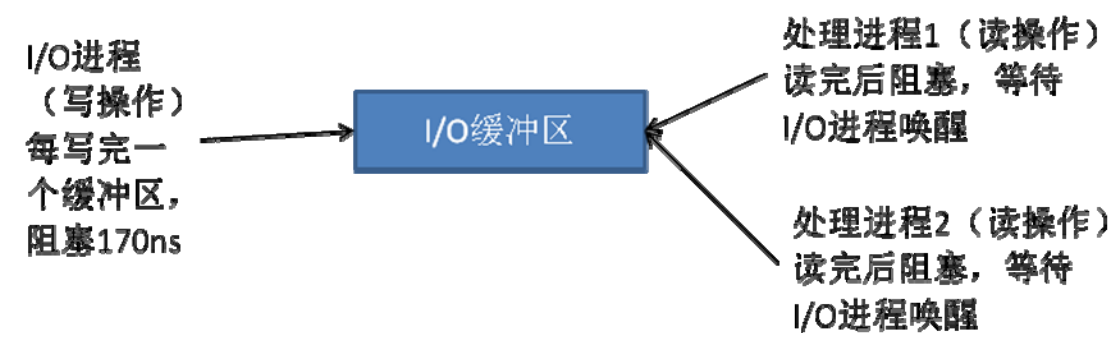
（本题默认分值：15 分）

标准分为 10 分，参考答案以标准分的比例给分或扣分。

本题为两问，第一问占题目总分值的 50%，第二占题目的 50%。

第一问参考答案：

由题目可知，I/O 进程写满一个缓冲区后，至少需要 170ns（150+20）该缓冲区才能腾空，所以 I/O 进程要阻塞 170ns 后才能继续写数据，当 I/O 进程写数据时，处理进程必须阻塞，所以一个简单的单缓冲结构如下图所示：



I/O 进程用 100ns 的时间把数据写入缓冲区，然后阻塞，处理进程 1 和处理进程 2 分别读完数据后，两个处理进程阻塞，由处理进程 2 负责唤醒 I/O 进程。
注意：学生只要回答出 I/O 进程和处理进程之间的同步关系，本题就可以得满分。题目中的时延数据，其实都是障眼法，呵呵。

第二问参考答案：

该题目是一个典型的读者-写者问题，有 1 个写者（I/O 进程）和两个读者（处理进程），所以 IPC 方案可设定如下(信号量方案)：

信号量定义：

Semaphore Writer = 1;
Semaphore Mutex = 1;
Semaphore Reader = 0;
int iReaderCount = 0;

```
I/O 进程代码:

P(Writer);
写操作.....;
V(Reader);
V(Reader);
结束
```

```
处理进程代码:

P(Reader);
P(Mutex);//该互斥信号量保护 iReaderCount 计数
iReaderCount++;
P(Mutex);
读操作.....;
P(Mutex);
iReaderCount--;
if(iReaderCount == 0)
{V(Writer);} //第二个读完缓冲区的进程唤醒 I/O 进程
P(Mutex);
```

12. 内存管理体系设计：动态链接库（DLL）是一种非常重要的软件设计技术，一个 DLL 可同时为多个不同的进程提供共享指令，并支持不同进程的独立运行，请尝试完成如下设计工作：

- 1) 请简要描述动态链接的基本实现过程；
- 2) 假设采用纯段式的内存管理机制，请设计一种适用于动态链接库的进程数据结构，能够记录所有正在共享该 DLL 的进程的有关信息。
- 3) 请描述你所设计的数据结构在实际运行过程中是如何工作的。

（本题默认分值：15 分）

标准分为 10 分，参考答案以标准分的比例给分或扣分。

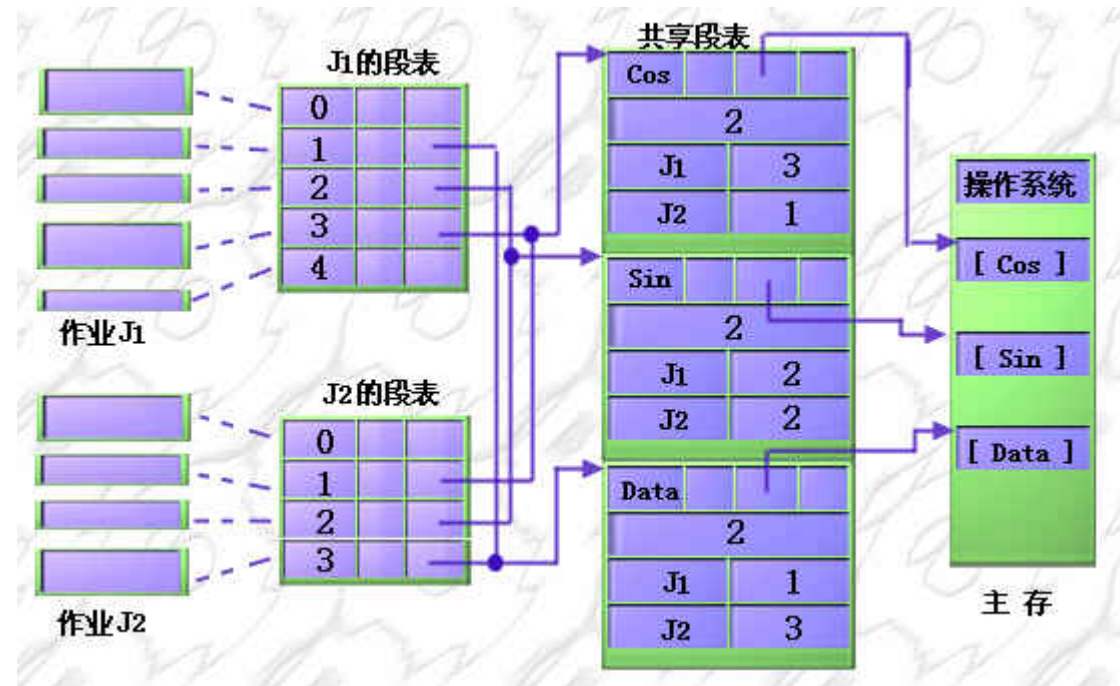
第一问占题目分值的 30%，第二问占 40%，第三问占 30%。本题是课程中详细介绍的动态链接的实现过程，参考答案如下：

第一问，动态链接的实现过程：

动态链接是通过间接字（Indirect Word）来实现的，间接字的结构包括“Link Indicator”和“Seg ID|Offset”两部分，当程序编译时，会将调用 DLL 的指令写成间接字的形式，在运行过程中，OS 将翻译间接字，寻找 DLL 中对应的段及其指令（靠 Offset 来计算）并执行，由此完成动态链接的基本过程。回答出间接字的基本含义即可。

第二问，数据结构设计

该数据结构其实是一个简单的共享段表，对 DLL 而言，每个段都有一个共享段表，主要包括“进程 ID、进程段号、共享计数值、共享段内偏移”共四项内容，下图描述了一个基本的共享段结构。学生只要回答出这四项内容中的 2 项，本问就给满分。



第三问：学生只要简单描述基于分段的共享处理过程既给满分。

13. 文件系统设计：基于手机和平板电脑的移动互联网已经风靡全世界，用户通过移动终端共享图片、视频和数据文件已经成为非常重要的 IT 应用模式，假设你现在就是一名非常优秀的移动互联系统管理员，你面临着不同应用环境下的文件管理系统管理问题，请尝试回答如下问题：

- 1) 请设计一种文件目录体系，能够保存大量用户上传的各种小文件。
- 2) 为提升用户访问各种共享文件的效率，必须为这种文件目录体系建立合理的索引管理机制，请简要设计一种有效的索引机制，能够快速告知用户最新共享文件、最热门共享文件等信息。

注意：当用户量以百万计的时候，你必须考虑 CPU 和内存的负载。

（本题默认分值：10 分）

标准分为 10 分，参考答案以标准分的比例给分或扣分。

本题为开放式的设计问题，只考察学生对文件系统的基本思维，含义点到即可给分。每一问占题目分值的 50%

1. 目录体系设计：因为用户上传的都是小文件，所以这意味着目录项要尽可能的短小，否则大量小文件将消耗大量的目录项存储空间。第二点要考虑的是目录层次，因为许多文件会经常被共享，所以目录体系不宜过深，如果目录体系太深的话，那么用户共享文件时就会消耗大量的寻址时间。基于以上两点考虑，目录体系最好设为 2-3 级，顶级目录是根节点，第二级目录是分类的文件目录，第三级目录可以考虑按文件上传时间（以周/月为单位）进行分类，这样的目录体系比较合理。
2. 文件索引机制：按照分类、分时的目录体系设计，可在每一级目录上设定该目录内的文件索引，这样就能实现多级索引表，由于第二级目录是分类文件目录，所有第二级目录中的索引信息直接合并就能形成完整的文件索引。