

南开大学信息技术科学学院本科生 2010-2011 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

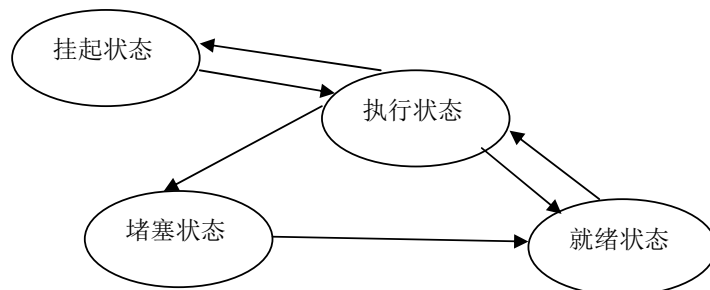
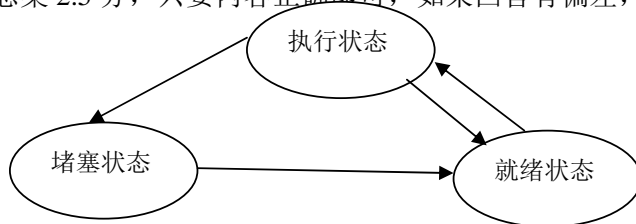
1. 为什么需要设计进程状态呢？请列出 2 组你了解的进程状态集，并简要描述状态转移过程。

为什么需要设计进程状态：通过进程状态才能实现对进程的调度和管理（1 分）

进程状态集 1：就绪状态、执行状态、堵塞状态

进程状态集 2：就绪状态、执行状态、堵塞状态、新建、退出、挂起态

每个状态集 2.5 分，只要内容正确即可，如果回答有偏差，酌情扣分。



2. 在进程通信问题的解法中，请举出两种典型的技术思想，并对比其优点和缺点。

IPC 问题的主要技术思想包括忙等待、信号量、消息机制以及管程方法，只需列出两种即可（每种 1 分）。

忙等待：优点是实现过程非常简单，缺点是无法在本质上解决 IPC 问题。

信号量：优点是能彻底解决 IPC 问题，缺点是技术实现复杂非常容易出错。

消息机制：无明显缺点，是目前最主要的解决方法。

管程：依靠编译器将 IPC 问题封装在一个封闭的处理过程中，优点是简单，缺点是难以适应所有应用环境。

针对优缺点分析只要大致含义正确即可，如有偏差酌情扣分，一种技术思想的优缺点分析占 2.5 分。

3. 何谓并发度？请列出至少三种支持多道程序的内存管理技术方法？

并发度是指系统在同一时间间隔内两个或多个事件同时发生、并行处理的能力。（含义正确即可，3 分）

内存管理方法：固定分区、动态分区、分页机制、分段机制等（列出三种即可，一种 1 分）

4. 请简要说明 CPU 与 I/O 设备进行通信的三种模式，你认为哪种模式最有效？为什么？

1. 轮询

2. 中断驱动

3. 直接存储访问 (DMA)

DMA 是最有效的。因为实际的数据传输过程不需要 CPU 的参与。

5. 请列出三种文件系统磁盘空间的分配形式，并简要分析其优缺点。

	优点	缺点
连续分配	1.实现简单。 2.读取性能高。	容易产生磁盘碎片
链表分配	能充分利用磁盘空间，可自由动态的实现文件内容的扩展。	1. 随机读取性能较差。 2. 数据字节数不是 2 的整数次幂，

		降低系统的运行效率。
索引表分配 (I-Node)	能高效的管理磁盘空间，支持复杂的文件系统	管理成本高，实现技术复杂

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. 内存访问时间问题：考虑有这样一个分页系统，该系统在内存中存放了二级页表，页面大小为 4K，在 TLB 中存储了最近访问的 16 个页表表项。如果内存访问需要 80ns，TLB 检查需要 20ns，页面交换（写/读磁盘）时间需要 5000ns。假设有 20% 的被替换页面的内容被更改过，如果 TLB 命中率是 95%，缺页率是 10%，请问 CPU 访问一个数据需要多长时间？（本题默认分值：12 分，列出计算公式与计算结果即可）

解题思路：

- 1) 二级页表：32 位计算机中的地址字长度为 32 位，页面大小为 4 K，二级页表意味着一级页表地址长度为 10 位，二级页表长度为 10 位，页内地址偏移为 12 位。
- 2) 数据访问流程：首先读 TLB，如果 TLB 命中则访问内存一次即可获得数据，如果 TLB 不命中，则访问内存中的一级页表（第一次访问内存），然后再通过一级页表访问二级页表（第二次访问内存），如果发生缺页中断则进行页面替换（可能会读写磁盘），然后访问内存中的页面内容（第三次访问内存）。

- 3) CPU 访问数据的时间：题目中给出了每一个环节的访问时间和相关概率，则本题求解的其实是数学期望，或者说是各种访问流程的加权平均值，权重就是每一个环节中的命中概率。
- 4) 注意：TLB 所保存的页表项对应的页面是不会被替换出内存的，所以只要 TLB 命中就意味着肯定没有缺页中断出现，另外一个问题是二级页表意味着即便页面在内存中，也需要三次访问内存才能获得数据。如果被替换页面的内容有更改，则需要两次页面交换才能完成缺页调度，以上三点问题是本题的关键所在，其考察内容是对 TLB 和二级页表、缺页中断与页面交换全流程的准确理解。

计算过程：

计算公式，CPU 访问内存中数据的时间 =  $\sum_{i=1}^n T_i P_i$ ；其中 i 是指各种内存访问情况， $T_i$  是指在情况 i 下的访问时间， $P_i$  是指发生该种情况的概率，题目中给出的各种百分比其实就是所谓的概率。

- 1) 第一种情况：TLB 检查 (20ns) 并命中 (0.95 概率)，直接访问内存中的页面 (80ns)，总时间是  $(20 + 80) * 0.95 = 95\text{ns}$
- 2) 第二种情况：TLB 检查 (20ns) 未命中 (0.05 概率)，直接访问内存中的一级页表 (80ns)，通过一级页表访问二级页表 (80ns)，页面保存在内存中 (0.90 概率，缺页率是 10%，则不缺页概率为 90%)，直接访问内存中的页面 (80ns)。该情况的总时间是  $(20 + (80 + 80 + 80) * 0.9) * 0.05 = 11.8\text{ns}$
- 3) 第三种情况，TLB 检查 (20ns) 未命中 (0.05 概率)，直接访问内存中的一级页表 (80ns)，通过一级页表访问二级页表 (80ns)，然后发生缺页中断 (0.10 概率)，被替换的页面无内容更改 (0.80 概率，页面内容被更改的概率是 20%，则页面内容无更改的概率是 80%)，执行一次页面交换 (5000ns)，最后直接访问内存中的页面 (80ns)。该情况的总时间是  $(20 + (80 + 80 + 5000 * 0.8 + 80) * 0.1) * 0.05 = 22.2\text{ns}$
- 4) 第四种情况，TLB 检查 (20ns) 未命中 (0.05 概率)，直接访问内存中的一级页表 (80ns)，通过一级页表访问二级页表 (80ns)，然后发生缺页中断 (0.10 概率)，被替换的页面内容更改 (页面内容被更改的概率是 20%)，执行两次

页面交换（ $5000\text{ns} * 2$ ），最后直接访问内存中的页面（ $80\text{ns}$ ）。

该情况的总时间是  $(20 + (80 + 80 + 5000 * 2 * 0.2 + 80) * 0.1) * 0.05 = 11.4\text{ns}$

评分细则：

以上四种情况的总和为  $140.4\text{ns}$ 。分析出至少四种情况的给总分的 60%，每种情况计算正确给 10%。如果分析错误，则酌情给分。

备注：本题解法中其实有更为复杂的情况，那就是二级页表不在内存中引发缺页中断，但是由于题目中并未给出二级页表本身缺页的概率，所以可以默认二级页表一直在内存中。

注解：从本题的解题过程中可以看出，如果 TLB 作为 Cache 的命中率足够高，那么原本非常浪费时间的多次内存访问事件反而由于发生概率小使得其对数学期望的贡献较小，最终的内存访问数据期望值是  $140.4\text{ns}$ ，非常接近一次 TLB 访问+一次内存访问（共计  $100\text{ns}$ ）的时间，这就是操作系统中进行内存访问效率优化的奥秘所在。

草稿区

7. 考虑具有如下特征的共享资源：1) 资源总数为 3，每个进程只能占用 1 个资源；2) 当使用该资源的进程小于 3 时，新申请资源的进程可以立刻获得资源；3) 当 3 个资源均被占用时，只有当 3 个资源均被释放后，其他申请资源的进程才能获得资源。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）

(1) 信号量定义及数据变量定义

本题要求中，最特殊的一点在于“当 3 个资源被占用时，只有当 3 个资源均被释放后，其他申请资源的进程才能获得资源”，这个要求意味着，一旦发生了 3 个资源均被占用的情况，那么即便某一个资源被释放了，也不能立即分配给其

他进程，必须等其他两个资源都被释放才能继续分配。基于此，设计信号量及数据变量如下：

信号量 R：记录当前可用资源的信号量，初始值为 3；

信号量 T：记录当前进程已申请的资源数，初始值为 1；题目中要求每个进程只能占用一个资源，这意味着在每个进程的实现代码中，都应定义一个信号量 T，用来控制该进程申请资源的次数；

信号量 M：互斥信号量，通用的互斥信号量，用于保护任何临界数据；

信号量 N：记录是否可申请分配资源的信号量，初始值为 1；该信号量用于记录资源分配状态。

数据变量 A：记录当前被占用资源的数量，当期为 3 时触发对信号量 N 的处理。

(2) 一个进程申请资源时的伪代码

Semaphore S = 3; //记录可用资源总数的信号量，为全局变量

Int A = 0; //记录当前已分配资源数量

BOOL bFull = FALSE; //记录是否 3 个资源均被占用

Semaphore M = 1; //互斥信号量，用以保护对变量 A 和 bFull 的访问

VOID Proc () //进程实体

{

    Semaphore R = 1; //记录该进程已申请资源的信号量，初值为 1 意味着每个进程只能申请一个资源

    While()

    {//资源申请临界区伪代码如下

        P(R); //首先执行信号量 R 的 P 操作，这意味着在一个进程中不可能同时占有两个资源

        P(S); //其次执行信号量 S 的 P 操作，这意味着如果可用资源数大于 0 且未出现 3 个资源均被占用的情况，则该进程可获得 1 个资源

        P(M); //用互斥信号量保护对 A 和 bFull 的操作

        A++;

```

If (A == 3) bFull = TRUE; //bFull 为 TRUE 意味着出现了 3 个资源均被占用的情况，以此为状态来控制 V(S)操作
V(M);
.....获取一个资源并使用它
.....释放资源
P(M)
A--;
If((A== 0) && bFull) {bFull = FALSE;V(S);V(S);V(S);} //该段代码意味着如果三个资源均被占用，则在第三个被占用资源释放后才一次性的发送三个 V(S)消息
Else if(!bFull){V(S)}; //该行代码是指并未出现三个资源均被占用的情况时，直接执行 V(S)以允许其他进程使用被释放的资源
V(M);
V(R); //该进程在释放了占用的资源后方可继续申请资源
...其他非临界区代码
}
}

```

评分规则：此题的关键之处在于如何记录 3 个资源均被占用的情况并以此为依据控制资源分配信号量，第一项要求（信号量 **R**）和第二项要求（信号量 **S**）各占总分的 30%，第三项要求（信号量 **M**，变量 **A** 与 **bFull**）的用法占总分的 40%。

如有其他解法可酌情给分，采用同样的分数比例进行评判。



8. 现有 5 个批处理作业，编号为从 A 到 E，它们同时到达。其估计运行时间为 15,9,3,6 和 12（按 A 至 E 顺序），其优先级为 6,3,7,9,4（值越小，表示优先级越高）。对如下 4 种调度算法，请计算每个进程的周转时间和平均周转时间。假设所有作业都是 CPU 密集型的。

- 1) 时间片为 1 的轮转调度算法;
- 2) 优先级调度算法;
- 3) 先来先服务算法（按 A 至 E 顺序）;
- 4) 最短作业优先算法。（本题默认分值：8 分，必须给出简要计算过程和计算结果）

1)严格轮转调度，A:45 B:35 C:13 D:26 E:42 平均周转时间： $(45 + 35 + 13 + 26 + 42) / 5 = 32.2$

2)调度顺序为 BEACD，A:36 B:9 C:39 D:45 E:21 平均周转时间： $(36 + 9 + 39 + 45 + 21) / 5 = 30$

3)调度顺序为 ABCDE，A:15 B:24 C:27 D:33 E:45 平均周转时间： $(15 + 24 + 27 + 33 + 45) / 5 = 28.8$

4)调度顺序为 CDBEA，A:45 B:18 C:3 D:9 E:30 平均周转时间： $(45 + 18 + 3 + 9 + 30) / 5 = 21$

评分规则：

每一项调度算法的计算过程为总分的 25%，每个进程的周转时间结果正确 15%，平均周转时间正确为 10%。

9. 死锁问题解决：一个系统中有 4 个进程和 4 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 ( $R_1, R_2, R_3, R_4$ )	最大需求量 ( $R_1, R_2, R_3, R_4$ )	资源总量 ( $R_1, R_2, R_3, R_4$ )
进程 A	4 0 0 1	4 4 2 1	16 5 2 8
进程 B	1 2 1 0	4 3 1 1	
进程 C	1 1 0 2	13 5 2 7	
进程 D	3 1 1 0	6 1 1 1	

请回答以下问题：

1) 当前状态是否安全？2) 进程 A 申请 1 个单位的  $R_2$  是否允许？3) 进程 C 申请 6 个单位的  $R_1$  是否允许？

注意：以上三个问题是相互独立的，请提供计算思路，并对每一问题给出安全路径或不安全路径。

（本题默认分值：10 分）

1)

初始状态

	还需要的资源数量	系统总共还空闲的资源数量 7 1 0 5
A	0 4 2 0	
B	3 1 0 1	
C	12 4 2 5	
D	3 0 0 1	

分给 B 之后，B 退出

	还需要的资源数量	系统总共还空闲的资源数量 <b>8 3 1 5</b>
<b>A</b>	<b>0 4 2 0</b>	
<b>C</b>	<b>12 4 2 5</b>	
<b>D</b>	<b>3 0 0 1</b>	

分给 **D** 之后，**D** 退出

	还需要的资源数量	系统总共还空闲的资源数量 <b>11 4 2 5</b>
<b>A</b>	<b>0 4 2 0</b>	
<b>C</b>	<b>12 4 2 5</b>	

分给 **A** 之后，**A** 退出

	还需要的资源数量	系统总共还空闲的资源数量 <b>15 4 2 6</b>
<b>C</b>	<b>12 4 2 5</b>	

分给 **C** 之后，**C** 退出

	还需要的资源数量	系统总共还空闲的资源数量 <b>16 5 2 8</b>
--	----------	---------------------------------

所以是安全的，安全路径 **B -> D -> A -> C**

初始状态

	还需要的资源数量	系统总共还空闲的资源数量 <b>7 1 0 5</b>
<b>A</b>	<b>0 4 2 0</b>	
<b>B</b>	<b>3 1 0 1</b>	
<b>C</b>	<b>12 4 2 5</b>	
<b>D</b>	<b>3 0 0 1</b>	

分给 D 之后，D 退出

	还需要的资源数量	系统总共还空闲的资源数量 <b>10 2 1 5</b>
<b>A</b>	<b>0 4 2 0</b>	
<b>B</b>	<b>3 1 0 1</b>	
<b>C</b>	<b>12 4 2 5</b>	

分给 B 之后，B 退出

	还需要的资源数量	系统总共还空闲的资源数量 <b>11 4 2 5</b>
<b>A</b>	<b>0 4 2 0</b>	
<b>C</b>	<b>12 4 2 5</b>	

分给 C 之后，C 退出

	还需要的资源数量	系统总共还空闲的资源数量
A	0 4 2 0	12 5 2 7

分给 A 之后，A 退出

	还需要的资源数量	系统总共还空闲的资源数量
		16 5 2 1

所以是安全的，安全路径为 D->B->C->A

综上，安全路径为 D->B->C->A

B->D->C->A

D->B->A->C

B->D->A->C

2)

初始状态

	还需要的资源数量	系统总共还空闲的资源数量
A	0 4 2 0	7 1 0 5
B	3 1 0 1	

<b>C</b>	<b>12 4 2 5</b>	
<b>D</b>	<b>3 0 0 1</b>	

进程 A 申请一个单位的 R2 之后

	还需要的资源数量	系统总共还空闲的资源数量 <b>7 0 0 5</b>
<b>A</b>	<b>0 3 2 0</b>	
<b>B</b>	<b>3 1 0 1</b>	
<b>C</b>	<b>12 4 2 5</b>	
<b>D</b>	<b>3 0 0 1</b>	

分给 D，D 退出

	还需要的资源数量	系统总共还空闲的资源数量 <b>10 1 1 5</b>
<b>A</b>	<b>0 3 2 0</b>	
<b>B</b>	<b>3 1 0 1</b>	
<b>C</b>	<b>12 4 2 5</b>	

分给 B，B 退出

	还需要的资源数量	系统总共还空闲的资源数量 <b>11 3 2 5</b>
<b>A</b>	<b>0 3 2 0</b>	
<b>C</b>	<b>12 4 2 5</b>	

分给 A, A 退出

	还需要的资源数量	系统总共还空闲的资源数量
C	12 4 2 5	
		15 4 2 6

分给 C, C 退出

	还需要的资源数量	系统总共还空闲的资源数量
		16 5 2 8

所以进程 A 申请 1 个单位的 R2 是容许的, 安全路径为 D->B->A->C,且路径是唯一的。

3) 进程 C 申请 6 个单位的 R1 是否允许

初始状态

	还需要的资源数量	系统总共还空闲的资源数量
A	0 4 2 0	
B	3 1 0 1	
C	12 4 2 5	
D	3 0 0 1	
		7 1 0 5

进程 C 申请 6 个单位的 R1 之后

	还需要的资源数量	系统总共还空闲的资源数量
A	0 4 2 0	
B	3 1 0 1	
		1 1 0 5

<b>C</b>	<b>6 4 2 5</b>	
<b>D</b>	<b>3 0 0 1</b>	

发生死锁，所以是不容许的，不安全路径为 C

评分规则，如果正确的描述了每个进程还需要的资源数量以及系统目前可用资源数量，则给 10% 的分数，每一问回答正确给 30% 的分数，对每一问而言，如果没有计算过程只有答案，即便答案正确该问得分也只给 5%。

草稿区

10. 工作集计算分析：一个进程的页访问序列是指该进程在连续的时钟周期内访问的页号组成的数列，例如某进程的页访问序列为 0123103，是指第一个时钟周期访问 0 号页面，第二个时钟周期访问 1 号页面，以此类推....。假设对某个进程，其页访问序列的长度为 P，其中包含 N 个不同的页号，工作集大小为 M 个页框，初始时工作集为空，页面调度采用按需调页模式，请回答以下问题：

- 1) 若使用 FIFO 页面置换算法，请问发生缺页中断次数的上限和下限是多少？
- 2) 若使用 LRU 页面置换算法，请问发生缺页中断次数的上限和下限是多少？

（本题默认分值：15 分，请给出计算分析的思路、过程和结果）

根据题意， $N \leq P$ ， $M \leq N$ ；由于初始时所有页框为空，所以至少需要 N 次缺页中断才能将 N 个所需要的页面加载进入内存，因此下限是 N；而最坏的情况是页面访问序列中的每一位对应的页面都不在内存中，因此发生缺页中断的上限为 P。

1) 使用 FIFO 页面置换算法的情形下，分两种情况讨论

当  $N > M$  时，发生缺页中断的下限是 N，上限是 P。

2) 使用 LRU 页面置换算法的情形下

当  $N > M$  时，下限是 N，上限是 P

评分规则：说明分析思路，占总分的 40%，两问问题的答案正确，各占总分的 30%。



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. 选择通道和多路通道是 I/O 设备管理中常见的两种管理技术，其中多路通道适用于低速 I/O 设备，而选择通道适用于高速 I/O 设备，I/O 处理机采用类似于进程调度的方法控制通道资源的分配。假如 1 台 32 位计算机共有两个选择通道和一个多路通道，共有两个磁盘设备、两个磁带设备、两个行式打印机、两个卡片阅读机，并且需要外联 10 个信息终端，这些 I/O 设备的数据传输速率如下：

磁盘驱动器：800KB/S；磁带驱动器：200KB/S；行式打印机：6.6KB/S；卡片阅读机：1.2KB/S；终端：1KB/S。

请回答以下问题：

- 1) 简要描述多路通道和选择通道的工作原理（图示最佳）；
- 2) 基于题目内容，请尝试设计一种 I/O 连接体系，描述两种通道与各种 I/O 设备的连接方式，描述 I/O 处理机所采用的调度方法，特别注意，我们将通过最大 I/O 数据传输速率来评价你的设计成果！
- 3) 如果希望优化一个 I/O 连接体系的数据传输速率，你能否提出一些优化设计的建议呢？

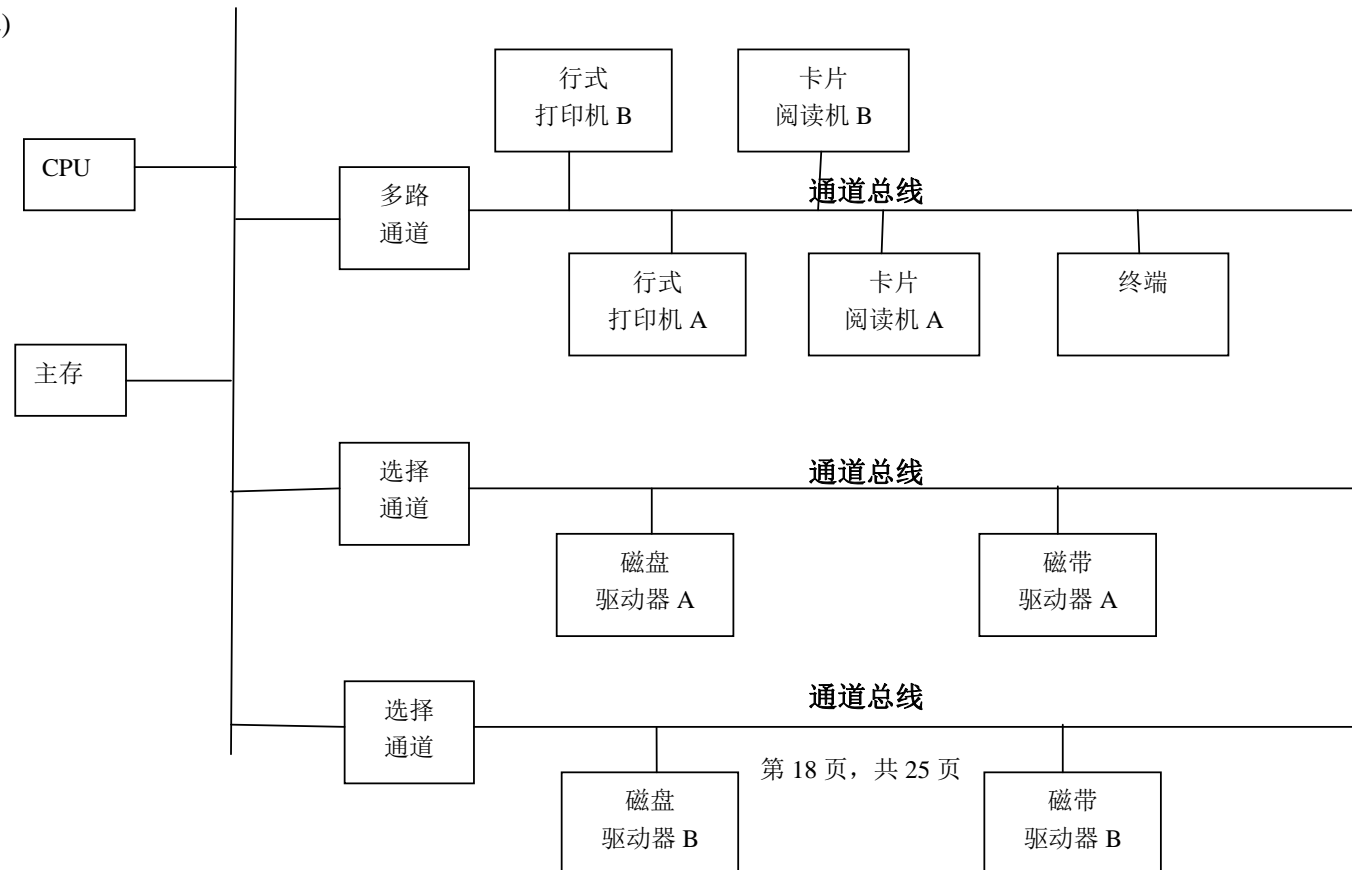
(本题默认分值: 15 分)

草稿区

1) 多路通道工作原理是系统内存在多个子通道, 每个子通道连接一个设备。各个子通道通过分时轮询调度自制占用主通道, 来完成设备数据的传输。

选择通道工作原理是在不同时间内只能选择一个设备, 然后完成该设备的数据传输工作。一旦设备被选中后, 通道进入“忙”状态, 只有当设备的数据传输全部结束为止, 通道才能选择其他设备。可以认为此时系统只有一个子通道, 然后逐次为每个物理连接的高速设备服务。

2)



草稿区

多路通道采用字节交叉传送方式；选择通道采用数组方式进行数据传输。

3) 在优化 I/O 连接体系优化工作中，需要把握几项原则：

1. 尽量减少使用的通道数量，因为通道价格昂贵；

2. 在其他条件确定的情况下，尽可能的增加各个外设的通道使用率；

这里还涉及到设备与设备控制器间连接方式的问题。为了提升整体系统的吞吐量，最有效的方法是尽可能的减少通道数量，并且增加设备到主机间的通路数（不是通道数）。这样还可以提升系统的可靠性，因为个别通道或控制器的故障，不会使设备和存储器之间没有通路。

评分规则：作为设计题，本题 3 个问题各占总分的 30%，50% 和 20%，根据答案的准确与详实程度给分，在第一问中，每一种通道的工作方式及特点回答正确给总分的 15%；

在第二问中，如果存在将高速设备（磁盘驱动器、磁带驱动器）放在选择通道的设计则扣除 20% 的总分，如果存在将两个磁盘驱动器或两个磁带驱动器放在一个选择通道的情况，则扣除 10% 的总分；注解：因为磁带驱动器和磁盘驱动器作为 I/O 设备的使用频率并不相同，一般来说磁盘设备的使用频率远高于磁带设备的使用频率（存储设备的金字塔原则决定），因此应在每一个选择通道上连接一个磁盘驱动器和一个磁带驱动器，这样做的好处是两个磁盘驱动器可分别使用一个选择通道同时提供数据，进而使得系统的总体数据吞吐量提升。

在第三问中，根据回答的情况酌情给分。

12. 内存管理体系设计：动态分区技术是一种简洁易用的多道程序内存管理方法，在一些嵌入式系统中，经常采用动态分区技术。假设在一个嵌入式操作系统环境中，所有用户进程都是被一次性加载进入内存，你的任务是设计完成用户空间内存的动态分区管理技术体系，请尝试完成如下设计工作：

1) 请定义支持动态分区内存管理所需要的数据结构，注意，为了提高效率，数据结构越简单越好；

2) 请用伪代码或文字来描述内存分配与回收的处理过程；

3) 动态分区技术的最大问题是容易产生内存碎片，请设计一种方法，能够在操作系统运行过程中，定期的清理内存碎片，请说明为实现内存碎片清理，需要定义的系统进程及其内存碎片清理过程；

草稿区

(本题默认分值: 15 分)

### 1) 需要定义一个空闲分区表和空闲分区链表

首先需要定义一个全局常量, 将其作为内存分配的基本单元, 一般来说内存分配的基本单元可以为 512 字节或 1K 字节

//空闲分区表

**Struct StFreeTbl**

{

**Int nID;**                    //空闲分区号

**Char \*pStartAddr;**      //空闲分区起始地址

**Int nSize;**              //空闲分区的大小

};

//空闲分区链表

**Struct StFreeNode**

{

**StFreeNode \*pPreNode;**    //指向前一个空闲分区的指针

**StFreeTbl freeTbl;**      //空闲分区表

**StFreeNode \*pNextNode;**   //指向后一个空闲分区的指针

};

### 2) 内存分配过程

假设任务需要分配的内存大小为 **size**, 空闲分区链表为 **freeNodes**

//遍历空闲分区表

For (; freeNodes->pNextNode != NULL; freeNodes = freeNodes->pNextNode)

{

    If (当前空闲分区满足任务要求, 此处可添加各种内存分配策略, 如首次满足、最小适配, 最大适配等)

    {

        返回当前空闲分区首地址;

        freeNodes->freeTbl->pStartAddr += size;         //当前空闲分区起始地址向后移动 size 个字节

        freeNodes->freeTbl->nSize -= size;             //当前空闲分区大小减少 size 个字节

        //如果该空闲分区的大小刚好等于请求分配内存的大小, 则删除这个空闲分区节点;

    }

}

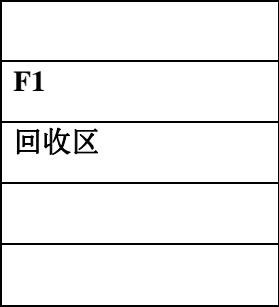
内存回收过程

分为四种情况:

1. 回收区与空闲分区的起始地址相邻, 将回收区与空闲分区合并为一个分区, 并将空闲分区的起始地址设置为回收区起始地址, 大小设置为回收区大小和 F1 大小之和。



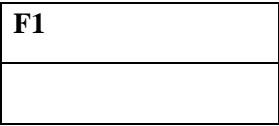
2. 回收区与空闲分区的结束地址相邻，将回收区与空闲分区合并为一个分区，将空闲分区的起始地址设置为 **F1** 的起始地址，大小设置为 **F1** 和回收区大小之和。



3. 回收区在两个空闲分区之间，空闲分区起始地址设置为 **F2** 的起始地址，大小为 **F2** 的大小+回收区的大小+**F1** 的大小



4. 回收区不与任何空闲分区相邻，新建一个空闲分区表，并加入空闲分区链表中。



回收区
F2

3) 清理内存碎片的过程：内存碎片的清理是指如果在空闲链表中存在大量的小分区，则这些分区因为无法满足一些大容量的内存分配请求而无法被有效使用，内存紧缩技术的实现过程是定义一个系统进程，该进程可通过设置定时器来定时唤醒，当进程运行时，它会把内存中所有的已占用分区紧缩在一起，即按照内存地址的顺序，将第二个已占用分区的起始地址调整到第一个已占用分区结束的地址(通过更改每个分区对应的内存地址来实现)，依次处理第三个、第四个.....。这样执行完成后，所有可用内存均合并为一个连续的空闲分区。

评分规则：以上答案是本题的最基础答案，课堂上曾详细讲解过 Minix 中动态分区的实现方式，以本题的基础答案为基准，第一问占总分的 30%，第二问占总分的 40%，第三问占总分的 30%，根据答案情况酌情给分。

13. 文件系统设计：文件格式对许多信息系统的应用效率有着极为重要的影响，根据应用环境的特点来设计文件的逻辑结构是系统设计人员必须具备的素质，假设你现在就是一名非常优秀的文件格式设计人员，你面临着不同应用环境下的文件格式设计问题，请尝试回答如下问题：

- 1) 请简要说明你所了解的文件逻辑结构组织方式有哪几种？
- 2) 某种应用情境下，文件很少被修改但是需要极为频繁的随机访问，请设计一种文件逻辑结构及其数据访问方法；
- 3) 某种应用情境下，文件频繁的被修改而且需要频繁的随机访问，请设计一种文件逻辑结构及其数据访问方法；

注意：请从访问速度、磁盘存储空间利用效率和易于使用（内容添加/删除/修改）等方面论述你的设计的合理性。

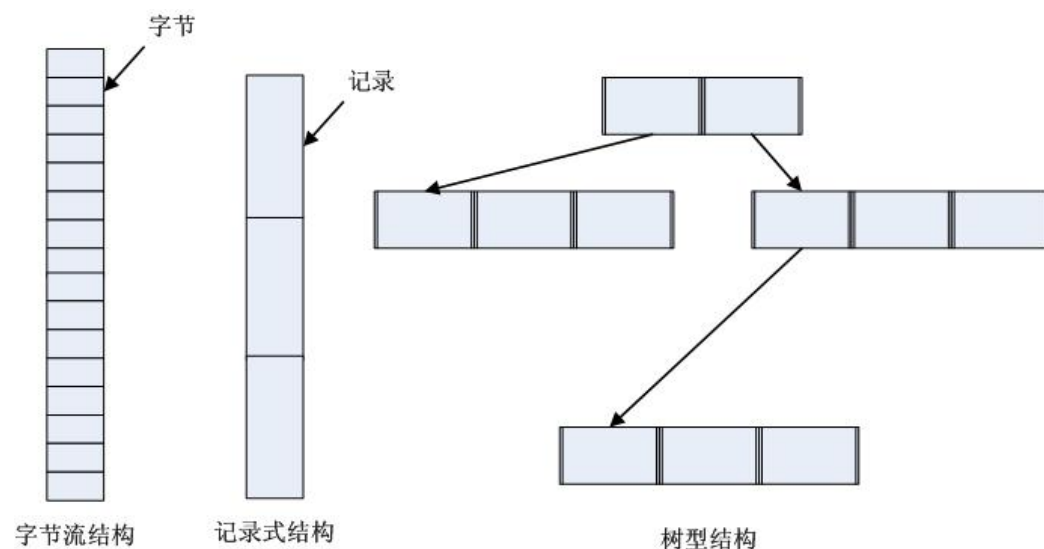
（本题默认分值：10 分）

1) 文件的逻辑结构的组织方式大体上分为三种

第一种是字节流式的逻辑结构，该逻辑结构难以进行快速的随机访问，每一次修改时都涉及到整个字节流的插入删除操作；

第二种是基于记录（每个记录大小一样）的逻辑结构，该逻辑结构能够进行快速的随机访问，但对记录进行增删改时，需要对整个记录序列进行处理；

第三种是基于树形（树中每个节点的大小可不同）的逻辑结构，也是最接近面向对象思想的逻辑结构，随机访问性能高于字节流式结构，低于记录式结构；但文件内容的修改效率极高。



2) 在这种情况下，文件很少被修改，因此需要注意的就是提高文件的随机访问速度。

此时最适合采用记录式文件结构，可为每个文件设计一个记录索引表，该索引表中存储了文件中每条记录的编号，长度和硬盘起始位置。

当对文件进行随机访问时，首先读出索引表内容，然后计算（文件内访问地址/记录长度）获得记录编号，以其为索引查询索引表获得该记录的起始地址，再用（文件内访问地址%记录长度）获得该记录内的偏移地址。

当对文件进行修改时，主要的修改操作设计增、删、改，每一种操作的设计简要描述如下：

第一种，在某个位置增加一个记录，首先通过文件索引表找到该位置所对应的记录编号，记录式文件结构的特点是每条记录定长，因此增加一条记录的操作在索引表中就意味着为新纪录分配一个编号，这个新纪录如果在文件尾部则效率最高，如果在文件头部或中间，则需要更新其后所有记录的编

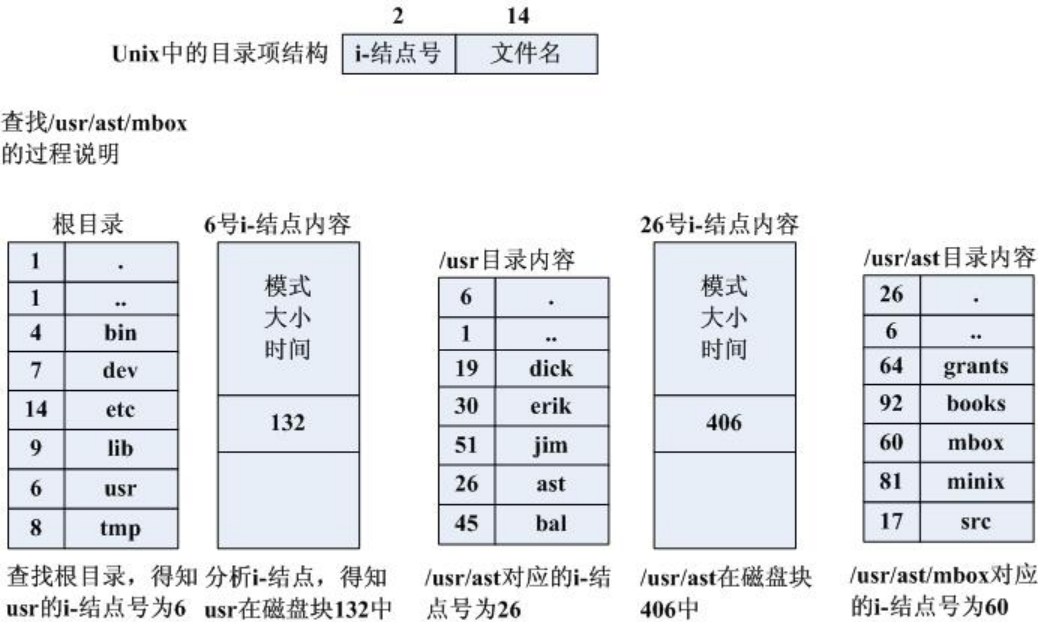


号及起始地址。

第二种，删除某个记录，在索引表中删除一个记录的操作是将该记录之后的所有记录向前移动一个记录长度的偏移地址。

第三种，修改某个记录，利用随机访问迅速的找到该记录的地址，然后在记录内进行内容修改操作。

3) 在这种情况下，文件会被频繁修改，而且频繁访问。因此最好的选择是采用树形逻辑结构方式，在课堂上曾讲解过 I-Node 文件管理方式，可遵循 I-Node 思想来设计文件系统，如下图所示。



评分规则：第一问中应回答三种文件逻辑结构的组织形式，每一种回答正确则占总分的 10%，第一问占总分的 30%；

第二问中应采用记录式文件逻辑结构+索引表结构，如果选择正确则获得总分的 15%，在使用记录式文件结构的基础上，如果文件读写设计思路正确，则获得总分的 10%，第二问占总分的 25%；

第三问中应采用树形文件逻辑结构，最佳设计思路是采用 i-node 节点的方式进行设计，如果选择正确则获得总分的 15%，如果文件读写思路正确，则获得总分的 30%，第三问占总分的 45%