暨南大學 JINAN UNIVERSITY

2020 年招收攻读硕士学位研究生入学考试试题（B）

****

招生专业与代码：计算机系统结构 081201、计算机软件与理论 081202、计算机应用技术 081203、电子信息(专业学位) 085400

考试科目名称及代码：计算机基础综合 848

考生注意：所有答案必须写在答题纸（卷）上，写在本试题上一律不给分。

第一部分 数据结构（75 分）

一、 单项选择题(每题 2 分，共 20 分)

1. 含有 m 个结点的二叉树链式存储结构中空指针的个数为（　　）。

A．2m　　B．m-1　　C．m+1　　D．m

2. 下列排序算法中元素的移动次数和关键字的初始排列次序无关的是（　　）。

A. 快速排序　　B. 插入排序　　C. 选择排序　　D. 希尔排序

3. 一个栈的进栈序列是 a b c d e，则栈的输出序列不可能的是（　　）。

A．a b c d e　　B．e d c b a　　C．d e c b a　　D．d c e a b

4. 需要的辅助空间最多的排序算法为（　　）。

A. 归并排序　　B. 快速排序

C. 基数排序　　D. 堆排序

5. 哈希表的平均查找长度说法错误的是（　　）。

A. 与处理冲突方法有关而与表的长度无关

B. 与选用的哈希函数有关

C. 与哈希表的饱和程度有关

D. 与表中填入的记录数有关

6. 有 n 个顶点 、e 条边且使用了邻接表存储的有向图进行深度优先遍历，其算法的时间复杂度是（　　）。

A. O（n+e）　　B. O（n^2）　　C. O（n+2e）　　D. O（n*e）

7. 已知一个长度为 11 的顺序表，其元素按关键字有序排列，若采用折半查找查找一个其中不存存在的元素，则关键字的比较次数最多是（　　）。

A．3　　B．4　　C．5　　D．6

8. 一棵完全二叉树上有 3001 个结点，其中叶子结点的个数是（　　）。

A. 1500　　B.1501　　C. 1000　　D.1001

9. 若一棵二叉树度为 2 的结点有 18 个,度为 1 的结点有 10 个,则度为 0 的结点个数是(　　)。

A. 46　　B. 28　　C. 19　　D. 17

10. m 阶 B-树是一棵（　　）。

A .m 叉排序树　　B. m-1 叉平衡排序树　　C. m 叉平衡排序树　　D. m+1 叉平衡排序树

二、 填空题(每空 2 分，共 14 分)

1. 已知一棵二叉树的中序遍历序列为GDHBAECIF，后序遍历序列为GHDBEIFCA，那么先序遍历序序列为_____。

2. 若某记录的关键字序列是（491，77，572，16，996，101，863，258，689，325），以第一个关键字为枢轴，写出采用快速排序算法第一趟排序的结果_____。

3. 将对称矩阵 A[8][8]的下三角部分逐行存储到起始地址为 2000 的内存单元中，已知每个元素占 4 个单元，假设第一个元素是 A[0][0]，则 A[4][6]的地址是_____。

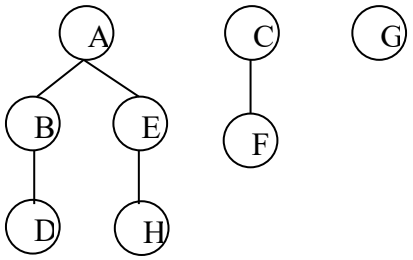
4. 在顺序表中插入一个元素，需要平均移动表中一半元素，具体移动元素的个数与_____有关。

5. 在哈希查找方法中，要解决两方面的问题，它们是_____和_____。

6. 循环队列中，Q.rear == Q.front表示循环队列空，表示循环队列满的条件是_____。

三、简答题（共 3 小题，每题 7 分，共 21 分）

1. 将下面的森林转换为二叉树（3 分），并给出该二叉树的中序线索链表（4 分）。



2. 设 Huffman 编码的长度不超过 4，若已对两个字符编码为 01 和 11，则最多还可以对多少个字符编码，为什么？(7 分)

3. 假设图的顶点是 A、B、C、D、E，请根据下面的邻接矩阵画出相应的有向图（3 分），然后画出图的邻接表和逆邻接表（4 分）。

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

四、编写算法（共 2 小题，每题 10 分，共 20 分）

1. 试编写一个算法完成下面的功能：对于输入的任意一个非负十进制整数，输出与其等值的八进制数。（10 分）

2. 试编写一个算法，在有向图 G 中，判定从顶点 Vi 到顶点 Vj 是否有通路。（10 分）

第二部分 操作系统（75 分）

五、判断题(每小题 1 分，共 10 分，正确的打 √，错误的打×)

1. 系统调用中的被调用程序运行在系统态。
2. 银行家算法采用了死锁预防的方法。
3. 文件系统采用树形目录结构可以节省内存空间。
4. 虚存管理允许用户程序大于主存容量，而且还可以提高系统的吞吐量。
5. SPOOLing 系统实现了设备的独立性。

6. 分时系统的时间片越小，用户的满意度就越高。
7. 管程每次只允许一个进程进入。
8. 操作系统既可看作虚拟机，也可看作资源管理器。
9. 在作业调度时，采用最高响应比优先的作业调度算法可以得到最短的作业平均周转时间。
10. 并行程序设计中，使用信号量比使用管程更能保证程序的正确性。

六、 填空题 (每小题 1 分，共 10 分)

1. 对于速率为 9.6KB/s 的数据通信而言，如果设置一个具有 8 位的缓冲寄存器，则 CPU 中断时间和响应时间分别大约为__（1）__ 、__（2）__。
2. 如果计算机连接了三个同类型的激光打印机及五个同类型的喷墨打印机，需要安装的驱动程序数目是__（3）__。
3. 在具有 n 个进程的系统中，允许 m 个进程(n≥m≥1)同时进入它们的临界区，其信号量 S 的值的变化范围是__（4）__，处于等待状态的进程数最多有__（5）__个。
4. 动态分区的__（6）__算法可以使内存中的空闲分区分布得更均匀。
5. UNIX 的目录项由文件名和__（7）__构成。
6. 若干事件在同一时间间隔内发生称为__（8）__ 。
7. 虚拟存储器具有__（9）__、__（10）__ 和虚拟性三大特征。

七、 单选题(每小题 1 分，共 10 分)

1. 请求调页系统中，如下算法中，(　) 淘汰自上次访问以来经历时间最长的页面。
   A. FIFO　　B. OPT　　C. NRU　　D. LRU
2. 下列进程调度算法中，(　) 可能会出现进程长期得不到调度的情况。
   A. 静态优先权法　　B. 抢占式调度中采用动态优先权法
   C. 分时处理中的时间片轮转调度算法　　D. 非抢占式调度中采用 FIFO 算法
3. 分时系统中，CPU 进程切换需要 3ms，为使得 100 个用户均能在 1 秒内得到响应，可以选择的时间片是（　　）。
   A. 2ms　　B. 50 ms　　C. 10ms　　D. 7 ms
4. 磁盘的 I/O 控制主要采取(　) 方式。
   A. 程序 I/O　　B. 中断　　C. DMA　　D. SPOOLing
5. 系统产生死锁是指(　) 。
   A. 系统发生重大故障
   B. 若干进程同时处于阻塞状态
   C. 请求的资源数大于系统提供的资源数
   D. 若干进程等待被其他进程所占用而又不可能被释放的资源
6. 通道又称 I/O 处理机，它用于实现(　) 之间的信息传输。
   A. CPU 与外存　　B. CPU 与外设　　C. 内存与外存　　D. 内存与外设
7. 下面叙述正确的是(　) 。
   A. 程序段是进程存在的唯一标志
   B. 系统通过 PCB 来控制和管理进程，用户可以从 PCB 中读出与本身运行状态相关的信息
   C. 当进程有执行状态变为就绪状态时，CPU 现场信息必须被保存在 PCB 中
   D. 当进程申请 CPU 得不到满足时，它将处于阻塞状态
8. 在没有快表的情况下，分页系统要访问(　)次内存。
   A. 1　　B. 2　　C .3　　D. 4
9. 计算机操作系统中，若 WAIT、SIGNAL 操作的信号量 S 初值为 3，当前值为-4，则表示当前

有(　　) 个等待信号量 S 的进程。

A. 1　　B. 2　　C. 3　　D. 4

10. 有 10 个进程共享 5 个打印机，若信号量 S 的当前值是-2，则当前有(　　)个进程提出了打印请求？

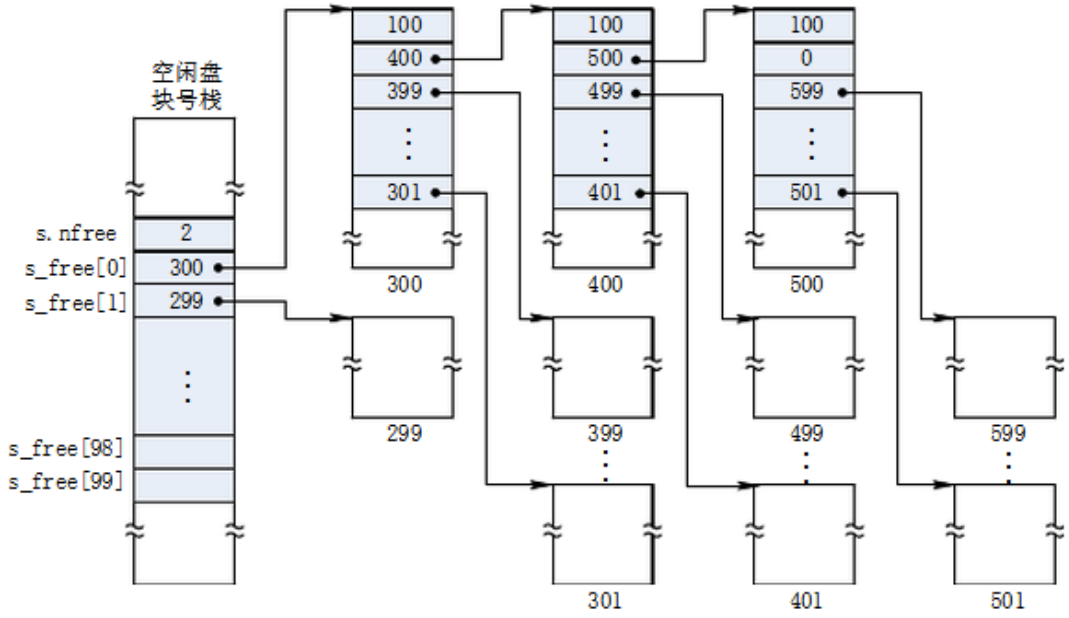
A. 10　　B. 7　　C. 5　　D. 2

八、 简答题(每小题 5 分，共 25 分)

1. 什么是文件目录、目录文件，各起什么作用？
2. 多级树形目录的文件系统，怎样才能提高查找文件的速度?
3. 多线程系统与传统多进程系统相比有哪些优点？
4. 分页存储管理和分段存储管理的主要区别有哪些？
5. 用伪代码或文字描述 fork()系统调用是如何创建进程的。

九、 应用题(每小题 10 分，共 20 分)

1. 某类 Unix 系统采用成组链接法来管理磁盘的空闲空间，目前磁盘的状态图如下（10 分）：



（1）该磁盘中目前还有多少个空闲盘块？(4 分)

（2）给出该系统的磁盘块分配及回收算法（流程图或描述）。(6 分）

2. 分析下面给出的表达式的并行性，并用信号量机制实现该表达式的并行计算。(10 分）

$$(3 * a * b + 4) / (c + d)^{(e-f)}$$

版本：v221222

本答案为非官方版本，仅根据本人水平所写。

若有错误，敬请谅解与斧正。

联系方式：qq1035775998

买家：

【腾讯文档】2020 年 848 纠错文档

https://docs.qq.com/sheet/DS0xHWWNwd1ZGam9a?tab=1ctotg

2020 年 848 计算机基础综合 答案

数据结构

1. 答案给出的页数都来自于《严书》电子版。
2. 因为没有正确答案，所以给所的答案并不完全正确，自己做的时候要保持质疑的态度。
3. 如果有校正会及时更新。
4. 您如果发现错误，请及时与我们联系，我们会给予奖励。

一、 单项选择题

答案：

1 - 5 C C D C D

6 - 10 A B B C C

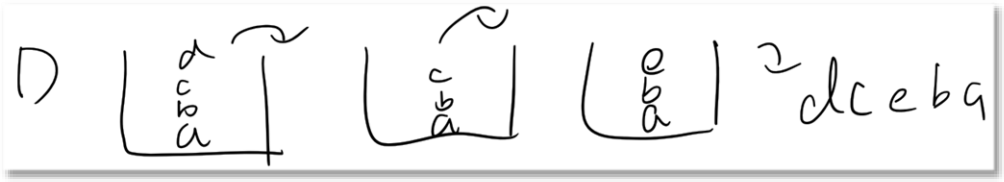
解析：

1. m 个结点的二叉树有 2m 个指针，m-1 条边，每条边用掉 1 个指针，因此空指针个数为 2m – (m - 1) = m + 1。
2. 我认为这道题应该是元素的比较次序才对。

   如果是元素的移动次数那应该是基数排序，没有答案。

   但如果是元素的比较次序，那就是选择排序，从头到尾不断与后面每一位（如果有）进行比较。

3.



b 和 a 顺序反了

4. 采用链式基数排序，空间复杂度为 O(n + rd)。

   归并排序 O(n)，快速排序 O(logn)，堆排序 O(1)

5. （2）查找过程中需和给定值进行比较的关键字的个数取决于下列三个因素：哈希函数，处理冲突的方法和哈希表的装填因子。

   从以上分析可见，哈希表的平均查找长度是 α 的函数，而不是 n 的函数。由此，不管 n 多大，我们总可以选择一个合适的装填因子以便将平均查找长度限定在一个范围内。

6. 遍历 n 个顶点和 e 条边，因此时间复杂度为 O(n + e)。

7. 根到结点 9-10 的路径。因此，折半查找在查找不成功时和给定值进行比较的关键字个数最多也不超过 $\lfloor \log_2 n \rfloor + 1$。

   相当于再往关键字下画两个空结点，找这些最远空结点。

8. 边的公式：$2n_2 + n_1 = 3001 - 1$，完全二叉树的n_1只能为 0 或 1，根据整除公式可知$n_1 = 0, n_2 = 1500$。又因为$n_0 = n_2 + 1$，因此叶子结点 1501。（3001 – 1500 也行）

9. $n_0 = n_2 + 1$

10. 不是 B，是 C。

    B。根据 m 阶 B 树的定义，树中的每个结点最多有 m 棵子树，所以是 m 叉树；而且 B 树严格限制所有失败结点在同一层次上，因此是高度平衡的；B 树是一种多路搜索树，根结点中的每个关键码的值都大于它左边子树上所有结点的关键码的值，同时小于它右边子树上所有结点的关键码的值，且根结点内所有关键码是从小到大有序排列的。

二、 填空题

答案：

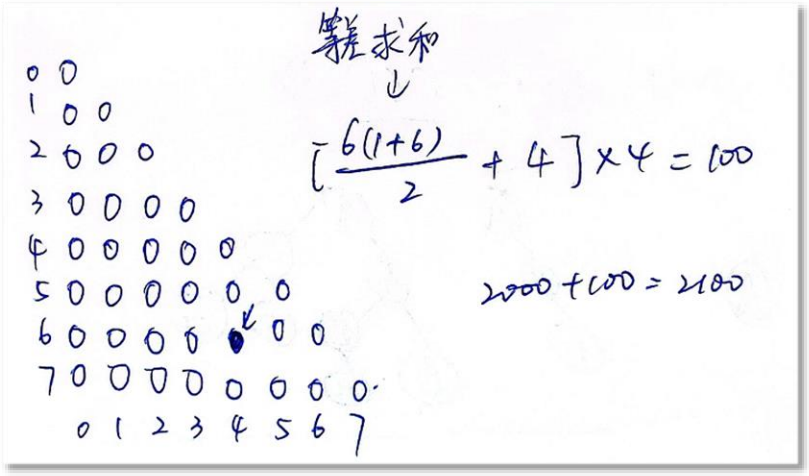
1. ABDGHCEFI
2. 325 77 258 16 101 494 863 996 689 572
3. 2100
4. 插入元素在表中的位置
5. 如何构造哈希函数 如何解决冲突
6. Q.front == (Q.rear + 1) % MAXSIZE

解析：

1. 先看后序找根（最后），然后再从中序分，之后重复直到划分结束。
2. 跟着这个严书中的算法来就好。

```
int Partition (SqList &L, int low, int high) {
  // 交换顺序表 L 中子表 r[low..high]的记录,枢轴记录到位,并返回其所在位置,此时
  // 在它之前(后)的记录均不大(小)于它。
  L.r[0] = L.r[low];                          // 用子表的第一个记录作枢轴记录
  pivotkey = L.r[low].key;                    // 枢轴记录关键字
  while (low<high) {                          // 从表的两端交替地向中间扫描
    while (low<high && L.r[high].key>= pivotkey) --high;
    L.r[low] = L.r[high];                     // 将比枢轴记录小的记录移到低端
    while (low<high && L.r[low].key<= pivotkey) ++low;
    L.r[high] = L.r[low];                     // 将比枢轴记录大的记录移到高端
  }
  L.r[low] = L.r[0];                          // 枢轴记录到位
  return low;                                 // 返回枢轴位置
}// Partition
```

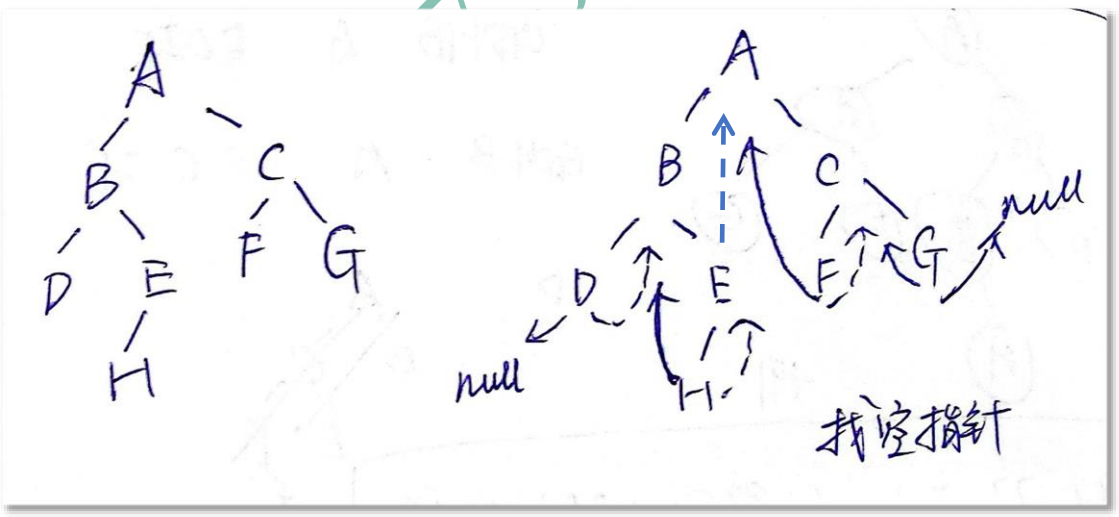
3. 硬算，实在不会就画图。因为是下三角，所以 A[4][6] = A[6][4]。（起始地址为 2000，26-1=25 , 25×4）



5. 803 真题。

三、 简答题

1.

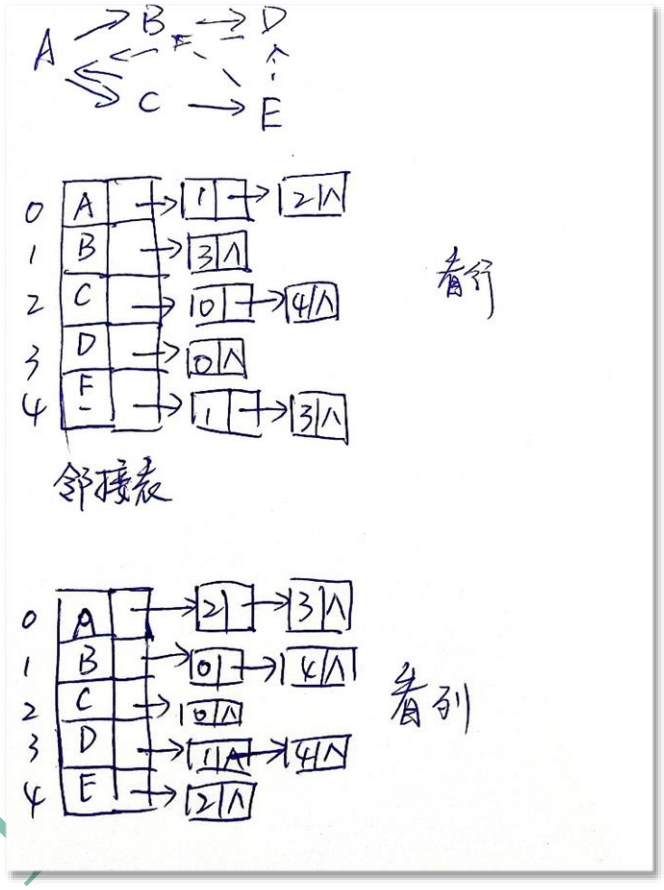


2. 8 个。Huffman 编码为前缀编码，要求编码不能为其他关键字前缀，也就意味着关键字不能为树的中间结点。换句话说，关键字只能作为叶子结点。

四个编码的树意味这个树有五层。因为第三层已有两个叶子结

点，而 Huffman 又不存在度为 1 的结点，所以第三层剩下两个度为 2 的结点。为了最大化叶子结点的个数，第四层可以多 4 个度为 2 的节点，因此，共有 8 个叶子结点给字符编码。

3.



四、 编写算法

1. 严书 P48

```
void conversion () {
  // 对于输入的任意一个非负十进制整数,打印输出与其等值的八进制数
  InitStack(S);        // 构造空栈
  scanf ("%d",N);
  while (N) {
    Push(S, N % 8);
    N = N/8;
  }
  while (!StackEmpty(s)) {
    Pop(S,e);
    printf ( "%d", e );
  }
}// conversion
```

2. 深度遍历和广度遍历都可以

   深度遍历。

   严书 P169

```
// - - - 算法 7.4 和 7.5 使用的全局变量 - - -
    Boolean visited[MAX];          // 访问标志数组
    Status ( * VisitFunc)(int v);  // 函数变量

    void DFSTraverse(Graph G, Status ( * Visit)(int v)) {
                                   // 对图 G 作深度优先遍历。
      VisitFunc = Visit;           // 使用全局变量 VisitFunc,使 DFS 不必设函数指针参数
      for (v = 0; v<G.vexnum; ++v) visited[v] = FALSE;  // 访问标志数组初始化
      for (v = 0; v<G.vexnum; ++v)
        if (!visited[v])  DFS(G, v);  // 对尚未访问的顶点调用 DFS
    }
```

算法 7.4

```
void DFS(Graph G, int v) {
  // 从第 v 个顶点出发递归地深度优先遍历图 G。
  visited[v] = TRUE;  VisitFunc(v);  // 访问第 v 个顶点
  for ( w = FirstAdjVex(G, v);  w>=0;  w = NextAdjVex(G, v, w) )
    if (!visited[w])  DFS(G, w);     // 对 v 的尚未访问的邻接顶点 w 递归调用 DFS
}
```

访问该顶点后，查找 visited[j]。

广度遍历

```
void BFSTraverse(Graph G, Status ( * Visit)(int v)) {
    // 按广度优先非递归遍历图 G。使用辅助队列 Q 和访问标志数组 visited。
    for (v = 0; v<G.vexnum; ++v)  visited[v] = FALSE;
    InitQueue(Q);                          // 置空的辅助队列 Q
    for ( v = 0;  v<G.vexnum;  ++v )
      if ( !visited[v]) {                  // v 尚未访问
        visited[v] = TRUE; Visit(v);
        EnQueue(Q, v);                     // v 入队列
        while (!QueueEmpty(Q))  {
          DeQueue(Q, u);                   // 队头元素出队并置为 u
          for ( w = FirstAdjVex(G, u);  w>=0;  w = NextAdjVex(G, u, w))
            if (! Visited[w])  {           //w 为 u 的尚未访问的邻接顶点
              Visited[w] = TRUE;  Visit(w);
              EnQueue(Q,W);
            } // if
          }// while
    }// if
}// BFSTraverse
```

同样也要查找 visited[j]。

操作系统

1. 答案给出的页数都来自于《题解》电子版。
2. 因为没有正确答案，所以给所的答案并不完全正确，自己做的时候要保持质疑的态度。
3. 如果有校正会及时更新。
4. 您如果发现错误，请及时与我们联系，我们会给予奖励。

五、 判断题

答案：

1 - 5 √ × × √ ×

6 – 10 × √ √ × ×

解析：

2. 死锁避免 P67
3. 多了表反而增加了内存占用。应该是提高检索速度和方便用户组织和使用自己的文件。P149
5. 设备独立性是指用户程序独立于物理设备，在驱动之上，实现逻辑与物理的转换。而 spooling 技术应该是实现了虚拟设备，单个物理设备可以让多个进程共享。
6. 太小也不好。因为调度也是需要时间的，不能让调度占用时间比例太大。
9. 短作业优先算法。

10.开发管程的目的就是更好的保证程序的正确性。

六、 填空题

答案

1. 0.1ms 0.0125ms
2. 2
3. [m - n, m] n-m
4. 循环首次适应
5. 指向该文件的索引结点指针
6. 并发
7. 多次性 对换性

解析：

1. 题解 P142 原题答案解析（B 是 Bit）

   8（位） / [9.6 ×1024（速率）] × 1000（时间换算） ≈ 0.8ms(K 是大 K，通信速率和内存的单位要分清，还有考试记得带计算器)

   响应时间就是 1/9.6 ×1024 × 1000 ≈ 0.1ms,处理 1 位的时间去处理 8 位。

没标注的话，B 我认为其实是 Byte.

所以答案应该是

8（位）/[8（字节换算成位）× 9.6 ×1024（速率））] × 1000（时间换算） ≈ 0.1ms

0.1ms / 8 = 0.0125ms 处理 1 位的时间去处理 8 位。

2. P145
3. P58
4. P88

(2) 循环首次适应算法。该算法由首次适应算法演变而成，它将空闲分区按起始地址递增的顺序排成循环链表，每次分配均从上次分配的位置之后开始查找。它使内存中的空闲分区分布得更均匀，从而减少查找空闲分区的开销，但会使存储器中缺乏大的空闲分区。

5. P148

中读出文件的物理地址等信息。因此，有些系统，如 UNIX 系统，便采用把文件名和文件描述信息分开的办法，即，将文件描述信息单独形成一个称为索引结点的数据结构，存放在外存的索引结点区，而组成文件目录的目录项中仅有文件名和指向该文件所对应的索引结点的指针。这样，便可大大减少文件目录所占的磁盘块数，从而加快检索目录的速度。

7. P109

虚拟存储器具有以下主要特征：

(1) 多次性。与常规存储管理的“一次性”相反，虚拟存储器将一个作业分成多次调入内存，多次性是虚拟存储器最重要的特征。

(2) 对换性。与常规存储管理的“驻留性”相反，在作业运行期间，虚拟存储器允许将那些暂不使用的程序或数据从内存调至对换区，待以后需要时再调入内存，从而有效地提高内存利用率。

(3) 虚拟性。虚拟存储器对内存的扩充是逻辑上的，用户所看到的大容量只是一种感觉，并不实际存在，因此是虚拟的。虚拟性是实现虚拟存储器的目标。

另外需要说明的是，虚拟存储器必须建立在离散分配的基础上，因此其实现方式也可分成请求分页、请求分段和请求段页式等方式。在实现过程中，虚拟存储器必须得到一定的硬件支持，如：系统必须具有一定容量的内存和较大容量的外存，必须提供请求分页(段)的页(段)表机制，以及缺页(段)中断机构和地址变换机构；还需要得到实现请求调页(段)的软件以及实现页(段)置换的软件的支持。

以上都是书上内容，辅导真的要好好看！

七、 单选题

答案：

1 - 5 D A D C D

6 - 10 D C B D B

解析：

1. 定义，要认识这些英文。NRU 是 CLOCK 算法。

3. A、D 都行，但 A 的使用时间短于切换时间，所以不合适。

4. P142

7. A. PCB

   B. PCB 对用户是透明的，只有系统能读

   C. P38

在创建进程时，系统将为它配置一个 PCB，在进行进程调度时，系统将根据 PCB 中的状态和优先级等信息来选择新进程，然后将老进程的现场信息保存到它的 PCB 中，再根据新进程 PCB 中所保存的处理机状态信息来恢复运行的现场；执行中的进程，如果需要访问文件或者需要与合作进程实现同步或通信，也都需要访问 PCB；当进程因某种原因而暂停执行时，也必须将断点的现场信息保存到它的 PCB 中；当进程结束时，系统将回收它的 PCB。可见，在进程的整个生命期中，系统总是通过其 PCB 对进程进行控制和管理，亦即，系统是根据其 PCB 而不是任何别的什么而感知到某进程的存在，所以说，PCB 是进程存在的唯一标志。

D. 就绪

8. P106 原题

9. 10. P23 第一段最后一句

在记录型信号量中，S->value 的初值表示系统中某类资源的数目，因而又称为资源信号量。每次对它进行 wait 操作意味着申请一个单位的该类资源，signal 操作意味着归还一个单位的该类资源。当 S->value＞0 时，它的值表示系统中该类资源当前可用的数目；S->value≤0 时，表示该类资源已分配完毕，其绝对值表示系统中因申请该类资源而阻塞在 S->list 队列上的进程数目。

八、 简答题

1. 文件目录：文件控制块的有序集合。对这些文件实施有效的管理，可以实现“按名存取”，并提高对文件的检索速度，同时实现文件的共享和重名。

   目录文件：目录以文件的方式存放在外存上。作为文件以供访问，并对文件目录进行管理。

   P148

7.1.3 文件目录

在一个计算机系统中，通常存储有大量的文件。为了能对这些文件实施有效的管理，必须将它们妥善地组织起来，这主要是通过文件目录来实现的。对目录管理的要求是能够实现“按名存取”，能够提供快速的目录查询手段以提高对文件的检索速度，并能为文件的共享和重名提供方便。

1. 文件控制块、目录项和索引结点

文件控制块(FCB)是 OS 用来描述和控制文件的一个数据结构，其中最基本的内容是文件名和文件的物理地址，其他的内容通常有文件的逻辑结构、文件的物理结构、文件的长度、文件的存取权限、文件的建立日期和时间、文件最后一次修改的日期和时间、文件的连接计数及文件主标识符等文件属性信息。

文件控制块与文件一一对应，文件控制块的有序集合被称作目录，其中的每个文件控制块被称为目录项。目录通常也是以文件的方式存放在外存上，故也被称作目录文件。

2. ① 将需要经常查找文件的最近公共父结点设置成当前目录或工作目录，即设立相对路径对文件进行查找。P149

在树形目录结构中，从根目录到任何数据文件，都只有一条唯一的通路，用“/”依次地将这条通路上的所有目录文件名和数据文件名连接起来，便构成了文件的绝对路径名。例如，图 7.2 中的 UserB 用户可用路径名“/UserB/Doc/F”来访问他的 F 文件。通常，一个用户进程在一给定的时间内所访问的文件仅局限于某个文件目录之下，为了简化文件的查找过程，可将该文件目录设置成“当前目录”或“工作目录”，以后用户进程对各文件的访问都可相对于“当前目录”而进行，而将当前目录到数据文件之间的所有目录文件名(不包括当前目录文件名)与数据文件名用“/”依次连接起来，便构成了文件的相对路径名。如用户 UserB 的当前目录是 Doc，则他可使用相对路径名“F”来访问自己的 F 文件。

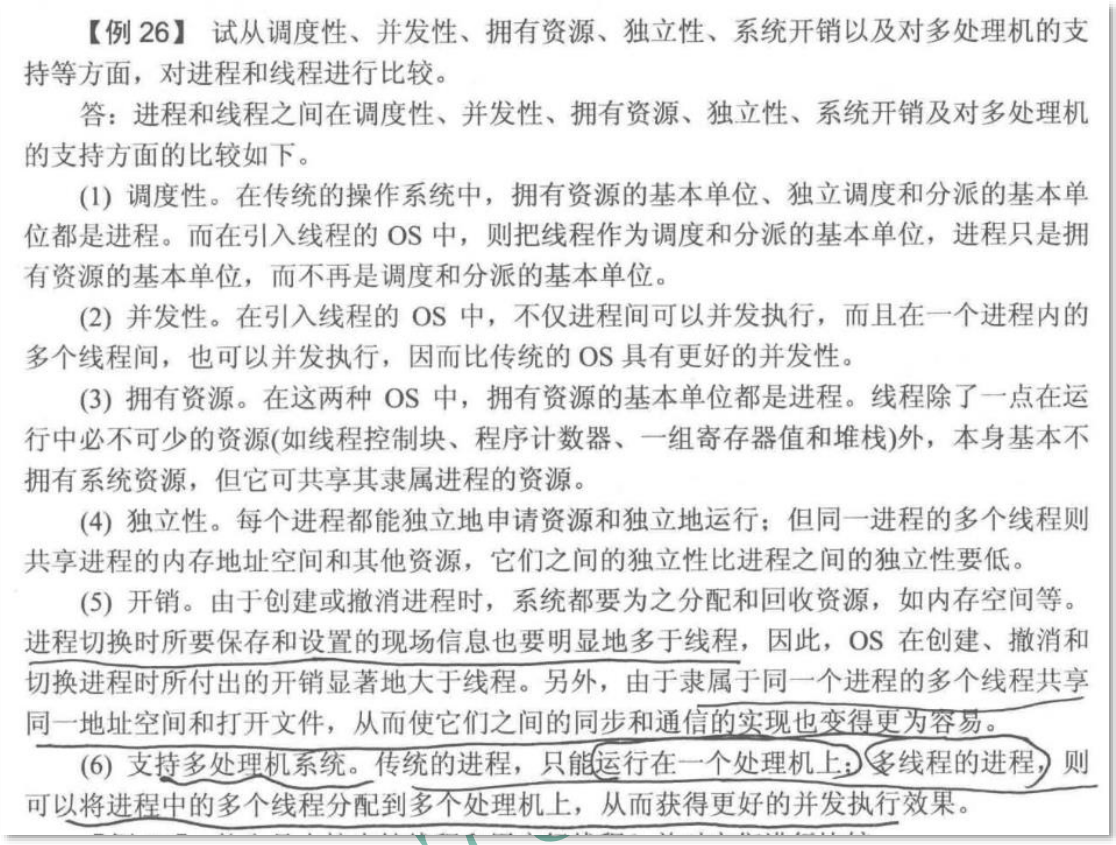
上面答案其实够了，以下为补充。

② 选取好的文件存储结构，提高文件访问速度。

③ 提高磁盘 I/O 速度。

④ 利用链接方式共享文件。

3. P55



【例 26】 试从调度性、并发性、拥有资源、独立性、系统开销以及对多处理机的支持等方面，对进程和线程进行比较。

答：进程和线程之间在调度性、并发性、拥有资源、独立性、系统开销及对多处理机的支持方面的比较如下。

(1) 调度性。在传统的操作系统中，拥有资源的基本单位、独立调度和分派的基本单位都是进程。而在引入线程的 OS 中，则把线程作为调度和分派的基本单位，进程只是拥有资源的基本单位，而不再是调度和分派的基本单位。

(2) 并发性。在引入线程的 OS 中，不仅进程间可以并发执行，而且在一个进程内的多个线程间，也可以并发执行，因而比传统的 OS 具有更好的并发性。

(3) 拥有资源。在这两种 OS 中，拥有资源的基本单位都是进程。线程除了一点在运行中必不可少的资源(如线程控制块、程序计数器、一组寄存器值和堆栈)外，本身基本不拥有系统资源，但它可共享其隶属进程的资源。

(4) 独立性。每个进程都能独立地申请资源和独立地运行；但同一进程的多个线程则共享进程的内存地址空间和其他资源，它们之间的独立性比进程之间的独立性要低。

(5) 开销。由于创建或撤消进程时，系统都要为之分配和回收资源，如内存空间等。进程切换时所要保存和设置的现场信息也要明显地多于线程，因此，OS 在创建、撤消和切换进程时所付出的开销显著地大于线程。另外，由于隶属于同一个进程的多个线程共享同一地址空间和打开文件，从而使它们之间的同步和通信的实现也变得更为容易。

(6) 支持多处理机系统。传统的进程，只能运行在一个处理机上；多线程的进程，则可以将进程中的多个线程分配到多个处理机上，从而获得更好的并发执行效果。

a) 线程为基本调度单位，需要资源少，创建、销毁、切换开销小。

b) 线程间通信更加灵活（可以在进程内共享占用的内存空间和打开的文件）

c) 多线程的进程可以将进程分配给多个处理机，从而实现更好的并发效果。

4. P93

分页系统和分段系统有许多相似之处，比如：都采用离散分配方式来提高内存利用率，都要通过地址变换机构来实现地址变换。但在概念上两者是完全不同的，它们的区别主要表现在以下三个方面：

(1) 页是信息的物理单位，分页是为了提高内存的利用率。段则是信息的逻辑单位，它含有一组其意义相对完整的信息。分段是为了能更好地满足用户的需要。

(2) 页的大小固定且由系统决定。段的长度不固定，且由用户所编写的程序决定。

(3) 分页的地址空间是一维的，程序员只需利用一个记忆符，便可表示一个地址。分段的地址空间是二维的，程序员在标识一个地址时，既需给出段名，又需给出段内地址。

5. 我认为的答案。

进程创建原语的主要任务是创建进程控制块 PCB。具体操作过程是：先从 PCB 集合中申请一个空闲的 PCB，再为新进程分配内存等资源，并根据父进程提供的参数和分配到的资源情况来对 PCB 进行初始化，最后将新进程插入就绪队列。

文字里可以加 fork()。

九、 应用题

1. P175 原题

   a) 301

   b) P165

当系统要为用户分配文件所需的盘块时，若第一组不只一块，则将超级块中的空闲盘块数减 1，并将空闲盘块号栈栈顶的盘块分配出去；若第一组只剩一块且栈顶的盘块号不是结束标记 0，则先将该块的内容(记录有下一组的盘块数和盘块号)读到超级块中，然后再将该块分配出去；否则，若栈顶的盘块号为结束标记 0，则表示该磁盘上已无空闲盘块可供分配。

在系统回收空闲盘块时，若第一组不满 100 块，则只需将回收块的块号填入超级块的空闲盘块号栈栈顶，并将其中的空闲盘块数加 1；若第一组已有 100 块，则必须先将超级块中的空闲盘块数和空闲盘块号写入回收块中，然后将盘块数 1 和回收块的块号记入超级块中。

2. 根据运算符的优先级安排各表达式的运算顺序，如下图。

① $S_1 = 3*a*b+4$

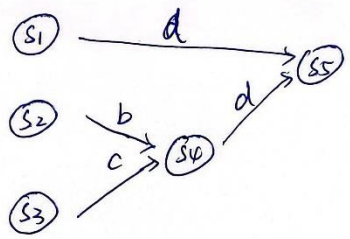
$S_2 = c+d$

$S_3 = e-f$

$S_4 = (c+d)^{(e-f)}$

$S_5 = (3*a*b+4)/(c+d)^{(e-f)}$

② 画出前趋图



③ 算法描述

semaphore a=b=c=d=0;

S1() { 计算a*b*3+4; signal(a); }

S2() { 计算c+d ; signal(b); }

S3() { 计算e-f ; signal(c); }

S4() { wait(b); (wait)(c); 计算(c+d)^(e-f); signal(d); }

~~S5() { signal(a); signal(d); 计算(3*(a*b)+4)/(c+d)^(e-f); }~~

S5() { wait(a); wait(d); 计算(3*a*b+4)/(c+d)^(e-f); }