

全面贯彻“跨越135分”辅导理念

全国硕士研究生入学统一考试



计算机历年真题全真解析



黄宝书

- ★历年考试真题答案解析
- ★重点、难点、考点梳理
- ★解题思路技巧全面分析
- ★名校名师
- ★紧扣大纲
- ★自我检测

最新版



北京邮电大学出版社
www.buptpress.com

全国硕士研究生入学统一考试 计算机历年真题全真解析·黄宝书

跨考教育计算机教研室 编

北京邮电大学出版社
·北京·

内 容 简 介

“计算机考研跨越 135 分必备”系列包括四个分册:(1)《全国硕士研究生入学统一考试计算机基础综合辅导·蓝宝书》;(2)《全国硕士研究生入学统一考试计算机考研核心习题集·绿宝书》;(3)《全国硕士研究生入学统一考试计算机历年真题全真解析·黄宝书》;(4)《全国硕士研究生入学统一考试计算机全真模拟题及答案·红宝书》。每一个分册的编写都凝聚了跨考教育教授们多年的研究心血。

本书以历年真题为基础,结合最新计算机考研统考大纲以及编者多年从事命题、阅卷、培训辅导的实际经验编写而成,编写过程中对还未统考前的各名校真题进行了删选,同时尽量保持了各名校真题的原貌及其风格。本书汇集了北大、清华、中科院、北邮、北航、浙大、哈工大、电子科大、川大、武大等名校计算机专业的往年真题,同时也收录了最近统考真题并一一进行了详细解析。希望考生能从中提取精华,受到启示,获得收益。

本书不仅特别适合在硕士研究生入学考试中参加理工类科目考试的考生,也适合各大院校学习理工类高级课程的师生,对于参加高级职称考试及其他相关专业人员来说,本书也是一本宝贵的学习和了解计算机课程的参考资料。

图书在版编目(CIP)数据

全国硕士研究生入学统一考试计算机历年真题全真解析:黄宝书/跨考教育计算机教研室编.--北京:北京邮电大学出版社,2011.7

ISBN 978-7-5635-2650-5

I. ①全… II. ①跨… III. ①电子计算机—研究生—入学考试—题解 IV. ①TP3-44

中国版本图书馆 CIP 数据核字(2011)第 118860 号

书 名: 全国硕士研究生入学统一考试计算机历年真题全真解析·黄宝书

作 者: 跨考教育计算机教研室

责任编辑: 姚 顺

出版发行: 北京邮电大学出版社

社 址: 北京市海淀区西土城路 10 号(邮编:100876)

发 行 部: 电话: 010-62282185 传真: 010-62283578

E-mail: publish@bupt.edu.cn

经 销: 各地新华书店

印 刷: 北京忠信诚胶印厂

开 本: 787 mm×1 092 mm 1/16

印 张: 19.25

字 数: 479 千字

版 次: 2011 年 7 月第 1 版 2011 年 7 月第 1 次印刷

ISBN 978-7-5635-2650-5

定 价: 36.50 元

· 如有印装质量问题,请与北京邮电大学出版社发行部联系 ·

前 言

真题是命题组专家多年辛勤研究的结晶,真题是练习题中的精华,反映了大纲的基本要求,又蕴含着命题的指导思想和发展趋势,是广大考生了解全国硕士研究生入学考试最直接的一手资料。最新历年真题最大限度地体现了考试大纲的基本精神,是检验考生对考试大纲的理解和对基础知识掌握的标尺。

为了更好地帮助考生复习,本书以历年真题为基础,结合最新考研计算机统考大纲以及编者多年从事命题、阅卷、培训辅导的实际经验编写而成,编写过程对统考前各名校的真题进行了删选,同时尽量保持了各名校真题的原貌及其风格。

本书汇集了北京大学、清华大学、中国科学院、北京邮电大学、浙江大学、哈尔滨工业大学等名校计算机专业历年的真题,同时也收录了最新全国统考的真题。试卷题目设计新颖,涉及知识面广泛,解题方法独到,甚至很有可能以直接或间接的方式预测以后的计算机统考真题。这些真题从各方面来讲都有着极高的使用价值。另外,加大习题的训练量是提高计算机专业课考研分数的唯一途径,因此,考生可以此书为训练习题,平时多加练习,定可大获裨益。

本书采用了分地区编排的套题结构,以满足不同地区考生的需求。整体由两部分组成:各名校历年真题、真题答案解析。真题答案解析部分参照了多部大学经典教材和教学参考书,由各名校知名专家对历年真题进行解析,详细介绍了解题方法及规范,对开阔解题思路,增加解题技巧,提高应试水平,有很大的帮助。

由于编者水平有限,书中难免存在不足之处,敬请业界有关专家和广大读者不吝指正。如遇到疑难问题,可通过以下方式与我们联系:bjbaba@263.net。

跨考教育计算机教研室

目 录

北京地区

清华大学 2008 年硕士研究生入学考试试题	1
清华大学 2008 年硕士研究生入学考试试题参考答案	5
清华大学 2006 年硕士研究生入学考试试题	13
清华大学 2006 年硕士研究生入学考试试题参考答案	15
中国科学院研究生院 2007 年硕士研究生入学考试试题	18
中国科学院研究生院 2007 年硕士研究生入学考试试题参考答案	26
中国科学院—中国科学技术大学 2005 年硕士研究生入学考试试题	39
中国科学院—中国科学技术大学 2005 年硕士研究生入学考试试题参考答案	41
中国科学院—中国科学技术大学 2004 年硕士研究生入学考试试题	45
中国科学院—中国科学技术大学 2004 年硕士研究生入学考试试题参考答案	49
北京大学 2008 年硕士研究生入学考试试题	54
北京大学 2008 年硕士研究生入学考试试题参考答案	57
北京大学 2005 年硕士研究生入学考试试题	62
北京大学 2005 年硕士研究生入学考试试题参考答案	66
北京大学 2004 年硕士研究生入学考试试题	70
北京大学 2004 年硕士研究生入学考试试题参考答案	76
北京航空航天大学 2008 年硕士研究生入学考试试题	81
北京航空航天大学 2008 年硕士研究生入学考试试题参考答案	85
北京航空航天大学 2007 年硕士研究生入学考试试题	91
北京航空航天大学 2007 年硕士研究生入学考试试题参考答案	94
北京邮电大学 2006 年硕士研究生入学考试试题	100
北京邮电大学 2006 年硕士研究生入学考试试题参考答案	105
北京理工大学 2007 年硕士研究生入学考试试题	111
北京理工大学 2007 年硕士研究生入学考试试题参考答案	118

华东地区

浙江大学 2005 年硕士研究生入学考试试题	126
浙江大学 2005 年硕士研究生入学考试试题参考答案	133
浙江大学 2004 年硕士研究生入学考试试题	139

浙江大学 2004 年硕士研究生入学考试试题参考答案	145
上海交通大学 2007 年硕士研究生入学考试试题	150
上海交通大学 2007 年硕士研究生入学考试试题参考答案	155
上海交通大学 2004 年硕士研究生入学考试试题	164
上海交通大学 2004 年硕士研究生入学考试试题参考答案	168
厦门大学 2006 年硕士研究生入学考试试题	173
厦门大学 2006 年硕士研究生入学考试试题参考答案	175
东南大学 2005 年硕士研究生入学考试试题	179
东南大学 2005 年硕士研究生入学考试试题参考答案	180

西南地区

四川大学 2005 年硕士研究生入学考试试题	184
四川大学 2005 年硕士研究生入学考试试题参考答案	187
电子科技大学 2008 年硕士研究生入学考试试题	194
电子科技大学 2008 年硕士研究生入学考试试题参考答案	199
重庆邮电大学 2007 年硕士研究生入学考试试题	210
重庆邮电大学 2007 年硕士研究生入学考试试题参考答案	215

东北地区

哈尔滨工业大学 2007 年硕士研究生入学考试试题	224
哈尔滨工业大学 2007 年硕士研究生入学考试试题参考答案	228
大连理工大学 2008 年硕士研究生入学考试试题	239
大连理工大学 2008 年硕士研究生入学考试试题参考答案	241

其他地区

武汉大学 2006 年硕士研究生入学考试试题	247
武汉大学 2006 年硕士研究生入学考试试题参考答案	251
华南理工大学 2007 年硕士研究生入学考试试题	259
华南理工大学 2007 年硕士研究生入学考试试题参考答案	262
2010 年全国统考考试试题	271
2010 年全国统考考试试题参考答案	280
2009 年全国统考考试试题	290
2009 年全国统考考试试题参考答案	297

北京地区



清华大学

2008年硕士研究生入学考试试题

	数据结构	计算机组成原理	操作系统	计算机网络
疑问				

★ ★ ★ 数 据 结 构 ★ ★ ★

一、证明题

1. 什么样的有向无环图有唯一的拓扑有序序列,并证明。

二、计算题

1. 有 n 个结点的二叉树最大高度、最小高度分别是多少?
2. 一棵有 n 个结点的树有 m 个叶结点,如果用左兄弟-右子女表示法,则有多少个结点的右指针域为空?
3. 哈夫曼树中,有 n 个叶结点,问一共有多少个结点?
4. 有 n 个结点的树的不同排列形式有多少种?

三、简答题

给定一个文件有 1 000 000 个记录,每个 200 B,记录中关键码大小为 50 B,页面大小为 4 KB,现以 B+树(最大关键码复刻)方式组织该文件,尽量使每结点拥有尽可能多的关键码,已知每个指针占用 5 B。问:

1. 该 B+树有多少个叶结点,共有多少层?
2. 该 B+树共有多少个索引结点?
3. 每次搜索要读盘多少次?

四、算法设计题

1. 给定 $A[n]$,设计一个算法,重排数组,使得奇数都在数组前半部分,偶数都在后半部分。要求时间复杂度 $O(n)$ 。

函数头: void exstorage(int A[], int n)

2. 重新设计一个直接选择算法函数,采用递归方式。

函数头: void selectsort(int A[], int left, int right)

操作系统

一、简答题

1. 简述磁盘 I/O 操作的时间组成部分, 阐述优化磁盘调度策略的目标。
2. 什么是内碎片和外碎片?
3. 内核线程和用户线程有什么区别? 各自有什么特点?
4. 什么是内核模式和用户模式? 为什么系统要设置这两种模式?

二、计算题

已知系统为 32 位实地址, 采用 48 位虚拟地址, 页面大小为 4 KB, 页表项大小为 8 个字节, 每段最大为 4 GB。

1. 系统将采用多少级页表? 页内偏移多少位?
2. 假设系统采用一级页表, TLB 命中率为 98%, TLB 访问时间为 10 ns, 内存访问时间为 100 ns, 并假设当 TLB 访问失败时才开始访问内存, 问平均页面访问时间是多少?
3. 如果是二级页表, 页面平均访问时间是多少?
4. 每用户最多可以有多少个段? 段内采用几级页表?
5. 如果要满足访问时间 ≤ 120 ns, 那么命中率至少需要多少?

三、PV 操作题

给定一个全局数组 $a[n]$ $b[n]$, 然后是 $T_1 \sim T_{n-1}$ 共 $n-1$ 个线程, 线程为代码如下:

```
Ti(){
    a = g(a, a[i - 1]);
    b = f(a);
}
```

其中 g 和 f 函数的作用是通过输入参数, 进行一系列运算后返回。相当于 T_i 以 a 和 $a[i-1]$ 为输入参数, a 和 b 为输出。

要求使用 PV 原语, 实现 $T_1 \sim T_{n-1}$ 的并发互斥, 尽量保证最大限度的并发。

($a[i-1]$ 为 T_{i-1} 线程的结果)

四、进程同步问题

假设当前处于非抢占调度策略, 进程只有两种方式可以放弃 CPU, 一个是主动调用系统调度函数 `sysnc()`, 此时进程主动放弃 CPU; 另一个方式是当进程执行 I/O 操作时, 系统将调度下一个进程。试分析如下三种进程对, 何时会出现不符合下列原则, 并说明原因:

1) 空闲则入; 2) 有限等待; 3) 保证互斥。

第一种:

```
Thread1(){
    sysnc();
    ---- critical section ----
    g = g + b;
```

```

    f = g - a;           // 临界区不能被打断
    -----critical section-----
}
Thread2(){
    -----critical section-----
    g = g + b;
    f = g - a;
    -----critical section-----
}

```

第二种:

```

Thread1(){
    sysnc();
    -----critical section-----
    g = g + b;
    f = g - a;
    -----critical section-----
}
Thread2(){
    -----critical section-----
    g = g + b;
    f = g - a;
    -----critical section-----
    sysnc();
}

```

第三种:

```

Thread1(){
    sysnc();
    -----critical section-----
    g = g + b;
    fstring = printf(...);      // 调用 I/O;
    f = g - a;
    -----critical section-----
}
Thread2(){
    sysnc();
    -----critical section-----
    g = g + b;
    f = g - a;
    -----critical section-----
}

```

计算机组成原理

一、填空题

1. 写出一1.125的IEEE754 32位标准的浮点数_____。
2. 控制器部件由哪五部分组成:_____、_____、_____、_____、_____。
3. 五级指令流水线是由哪五部分组成:IF、_____、_____、_____、_____。

二、下列指令能否用单字指令(字长为12位)实现。

1. 4条三寄存器指令。
2. 255条单寄存器指令。
3. 16条0寄存器指令。

三、输入/输出方式都有哪几种?请简要叙述各自特点。

四、1. 在虚拟页式系统中,虚拟地址的位数48位,可用的最大主存空间位128 GB,每页大小4 KB。设计多级页表(不要假设有TLB存在)。

2. 系统中为什么要设计TLB?画图表示出虚拟地址到真实地址的转换。



清华大学

2008年硕士研究生入学考试试题

参考答案

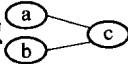
数据结构

一、证明题

1. 从离散数学的角度来看,拓扑排序就是由某集合上的一个偏序得到该集合上的一个全序。直观来说,偏序即集合中仅部分元素间可比较(存在某些元素间无法比较),全序即集合中所有元素间均可比较。拓扑排序的任务是在这个偏序上得到一个全序,即得到一个完成整个项目的各步骤的序列。排序依赖的原则就是各个步骤之间的优先关系。拓扑排序得到的序列不一定是唯一的,因为某些步骤间没有规定优先关系(这就是偏序的特点),在拓扑排序的时候人为地加入一些规则,使得到的序列为满足偏序关系的一个全序。

由此我们可以得到结论:当有向无环图满足全序关系时,其拓扑排序序列是唯一的。

证明:(反证法)偏序是满足自反性、反对称性、传递性的序。

若拓扑序列不唯一,则这个图的拓扑序列至少有两个。设其中的两个序列为 $a \cdots b \cdots, b \cdots a \cdots c$ 。在这个图的某部分必存在类似  的图,可以看出此部分图满足偏序关系,但不是全序关系。而本题的前提是:图为一个全序关系图,故与本题的前提矛盾。故假设不成立。

二、计算题

1. 最大高度: n (注:每层一个结点,故 n 个结点的树的最大高度为 n)。

最小高度: $\log_2 n$ (向下取整)+1(注: n 个结点的完全二叉树的高度最小)。

2. 当采用左兄弟-右子女表示法时,每个结点的指针域为两个,故共有 $2n$ 个指针域。由二叉树的性质 $n_0 = n_2 + 1$ 可知, $n_2 = n_0 - 1 = m - 1$,故右指针为空的结点个数为 $n - m + 1$ 个。

3. 在哈夫曼树中,只有度为 0 和度为 2 的结点,又度为 0 和度为 2 的结点间存在 $n_0 = n_2 + 1$ 关系,则哈夫曼树共有 $n + n - 1 = 2n - 1$ 个结点。

4. 有 n 个结点的二叉树的不同排列形式有 $1/(n+1) * C_{2n}^n$ 种。又因为树与二叉树可以相互转换,一棵树可以转换为唯一的一棵没有右子树的二叉树,故具有 n 个结点有不同形态的树的数目和具有互不相似的二叉树的数目相同。故具有 n 个结点的树的不同排列形式有 $1/n * C_{2(n-1)}^{n-1}$ 种。

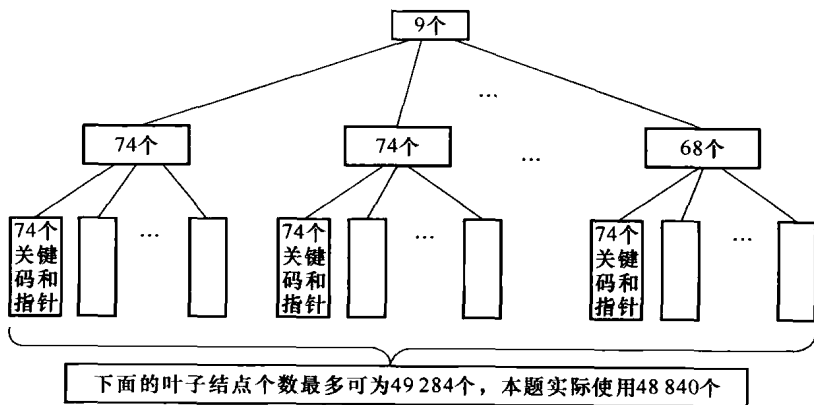
三、简答题

[分析] B+树的定义为:

- (1) 有 n 棵子树的结点中含有 n 个关键字;
- (2) 所有的叶子结点中包含了全部关键字的信息及指向这些关键字记录的指针,且叶子结点本身以关键字的大小自小而大顺序链接;
- (3) 所有的非终端结点可以看成是索引部分,结点中仅含有其子树(根结点)中的最大(或最小)关键字。

B+树的插入仅在叶子结点进行,当结点中的关键字个数大于 m 时要分裂成两个结点,它们所含关键字的个数分别为 $(m+1)/2$ 向上取整和 $(m+1)/2$ 向下取整。

整个文件实际大小为: $1\,000\,000 \times 200\text{ B}$;共需要页面个数为 $1\,000\,000 \times 200\text{ B} / 4\text{ KB} = 48\,828.125$,实际需要页面数为 48 829 个。故叶结点个数为 48 829 个;每个叶子结点需要的空间由关键码(50 B)和指针(5 B)共同决定。故每页上可以存放的结点个数为 $4\text{ KB} / 55\text{ B} = 74$,故该 B+树为 74 叉树。48 829 个记录依次插入到 B+树中,会形成如下所示的 B+树:



$(49\,284 - 48\,829) / 74 = 6.1$, 向上取整为 7。

进而可得 3 个小题的答案:

1. 该 B+树共有 $74 \times 74 \times 8 + 68 \times 74 = 48\,840$ 个叶子结点,由图可见共有 3 层。
2. 该 B+树有 $48\,840 + 9 + 1 = 48\,850$ 个索引结点。
3. 每次搜索要读盘次数为 4。因为每次搜索都是走了一个从根结点到叶子结点的路径。

四、算法设计题

```
1. #include "stdio.h"
   #include "iostream.h"
   #define n 10

   int A[n];
   main()
   {
       int i = 0, j, temp;
       printf("please enter the numbers:\n"); // 为数组输入数据
       for(i = 0; i < n; i++)
           scanf("%d", &A[i]);
```

```

i = 0;
while(A[i] % 2 == 1 && i < n)
    { i++; } // 找到第一个偶数, 第一个偶数
// i 始终指向前半部分的第一个偶数, j 若为奇数则与 i 号元素相交换
for(j = i + 1; j < n; j++)
    if(A[i] % 2 == 0 && A[j] % 2 == 1) { temp = A[j]; A[j] = A[i]; A[i] = temp; i++; }
for(i = 0; i < n; i++)
    printf("%d\n", A[i]);
system("pause");
}

```

2. 本题实质上是一个快速排序的实现, 将整个序列进行一趟快速排序, 分为两部分。程序代码如下:

```

#include <iostream>
using namespace std;

void swap(int &a, int &b) // 交换两个元素的值
{
    int temp;
    temp = a;
    a = b;
    b = temp;
}

int Partition(int a[], int low, int high)
// 在分割产生的序列部分内进行直接选择排序
{
    int pivot = a[low];
    while(low < high)
    {
        while(low < high && a[high] >= pivot) -- high;
        swap(a[low], a[high]);
        while(low < high && a[low] <= pivot) ++ low;
        swap(a[high], a[low]);
    }
    return low;
}

void QSort(int a[], int low, int high) // 递归实现快速排序
{

```

```

if(low<high)
{
    //第一趟直接选择排序,将整个序列分为两部分
    int pivotloc = Partition(a,low,high);
    QSort(a,low,pivotloc-1);           //分别对分割产生的部分进行快速排序
    QSort(a,pivotloc+1,high);
}
}

void main()
{
    int a[10] = {9,8,7,6,5,4,3,2,1,0};
    QSort(a,0,10);                     //调用函数
    for(int i = 0; i < 10; i++)         //输出排序结果
        cout << a[i] << " ";
    cout << endl;
}

```

操作系统

一、简答题

1. 磁盘 I/O 操作的时间分为:寻道时间、旋转延迟时间、传输时间。其中寻道时间是指把磁臂移动到指定的磁道所经历的时间。旋转延迟时间是指指定扇区移动到磁头下面所经历的时间。传输时间是指把数据从磁盘读出或向磁盘写入数据所经历的时间。

磁盘调度算法的目标是使磁盘的平均寻道时间最短。

2. 内碎片是指内存空间由于不断地被分配与释放,中间会出现一些很难再分配给其他用户使用的空间,这些空间就是内存碎片。相应的外存空间中也会出现一些无法利用的硬盘空间,这些就是外碎片。为了更好地利用存储空间,每隔一段时间要进行碎片整理,使得这些未被利用的空间可以合并成一个较大的空间供用户使用。

3. 区别:

内核线程:由操作系统内核创建和撤销;内核维护进程及线程的上、下文信息以及线程切换;一个内核线程由于 I/O 操作而阻塞,不会影响其他线程的运行。Windows NT 和 Windows 2000/XP 支持内核线程。

用户线程:由应用进程利用线程库创建和管理;不依赖于操作系统核心;不需要用户态/核心态切换,速度快。操作系统内核不知道多线程的存在,因此一个线程阻塞将使得整个进程(包括它的所有线程)阻塞。由于这里的处理器时间片分配是以进程为基本单位,所以每个线程执行的时间相对减少。

特点:

内核线程:在多处理器系统中,内核能够同时调度统一进程中多个线程并行执行;如果

进程中的一个线程被阻塞了,内核可以调度该进程中的其他线程占有处理器运行,也可以运行其他进程中的线程;内核支持线程具有很小的数据结构和堆栈,线程切换速度较快,开销较小;内核本身也可以采用多线程技术,提高系统的运行速度和效率。

用户线程:线程切换不需要转换到内核空间,从而节省了模式切换的开销也节省了内核的宝贵资源;调度算法可以是进程专用的,也可以用于进程中的线程调度;用户线程实现与操作系统平台无关。

4. 内核模式是操作的一种高特权模式,其中的程序代码能直接访问所有内存(包括所有的用户模式进程和应用程序的地址空间)和硬件。也称为“管理员模式”、“保护模式”或“Ring 0”。而用户模式相对于内核模式,则是一种低特权的模式,并不能访问所有的内存空间,又称为普通模式。

用户模式和内核模式的区别是用户模式下不可调用 Win32 内核函数,只有操作系统才可调用,应用程序无法调用,其实是基于安全考虑。如 callball 函数就只能在内核模式下运行。

二、计算题

1. 逻辑地址空间中的页面数为 $2^{48}/4\text{ KB}=2^{36}$,而实地址空间中的页面数为 $2^{32}/4\text{ KB}=1\text{ M}$,所以要采用多级页表, $2^{36}/1\text{ MB}=2^{16}$,故可以分成两级页表。第一级页表中存放 2^{16} 个页表项,第二级页表中存放 2^{20} 个页表项,正好可以表示逻辑地址空间中的所有页面的映射关系。

由于页面的大小为 4 KB,故页内偏移采用 12 位。

逻辑地址描述如下图:

第一级页号(16 位)	第二级页号(20 位)	页内地址(12 位)
-------------	-------------	------------

2. 采用一级页表时:平均页面访问时间 $=98\% \times 10 + (1-98\%) \times 100 = 11.8\text{ ns}$

3. 采用二级页表时:平均页面访问时间 $=98\% \times 10 + (1-98\%) \times [98\% \times 10 + (1-98\%) \times 100] = 10.036\text{ ns}$

4. 每个段的大小为 4 GB,而逻辑空间大小为 2^{48} ,则系统可分成 $2^{48}/4\text{ GB}=2^{16}$ 个分段。

每个分段中的页面数量为 $4\text{ GB}/4\text{ KB}=1\text{ MB}$,可以转化为 32 位实地址空间中的页面号,故采用一级页表就可以了,此时的逻辑地址可用下图表示:

段号(16 位)	段内页号(20 位)	页内地址(12 位)
----------	------------	------------

5. 仍旧采用题中的假设:TLB 访问时间为 10 ns,内存访问时间为 100 ns,并假设当 TLB 访问失败时才开始访问内存,这时设命中率为 h ,则平均访存时间为 $t=h \times 10 + (1-h) \times [h \times 10 + (1-h) \times 100]$ 。要使得访存时间 $\leq 120\text{ ns}$,则要求 $t \leq 120$,此时得到: $t \geq 0.118$ (另一个根舍掉)故命中率需要至少为 0.118。

三、PV 操作题

这个操作实际上是一个生产者消费者问题, T_i 看成是 T_{i+1} 的生产者, T_{i+1} 看成是 T_i 的消费者,设置一个数组 $\text{empty}[n]$,初始 $\text{empty}[0]$ 始终为 1,其他为 0,用来表示 $a[i]$ 中的数据是否产生完毕,mutex 标志 a 是否被访问,初始为 1,则使用 PV 操作后:

```

Ti(){
    Wait(mutex);
    Wait(empty[i-1]);
    a = g(a,a[i-1]);
    b = f(a);
    signal(empty[i%n]);
    signal(mutex);
}
    
```

四、进程同步问题

第一种线程对:不符合有限等待原则。

当第一个线程调用 sysnc 释放 CPU 后,第二个线程获得 CPU 会一直执行,线程没有释放 CPU 的过程,这样它一直占用 CPU,使得线程一在释放 CPU 之后一直处于等待状态,即出现无限等待的情况(还有可能是线程二先占用 CPU,这样它就一直占用并未释放,使得线程一永远无法执行)。

第二种线程对:不符合空闲则入的原则。

这是由于线程二中调用了 sysnc 操作释放 CPU,这时调用线程一,而线程一也是释放 CPU 使得可能出现关键区空闲却没有线程访问的情况。

第三种线程对:不符合保证互斥原则。

在两个线程运行的过程中,第二个线程会先释放 CPU,此时第一个运行。但是在第一个运行的过程中它会释放 CPU(但这时它已经在访问关键区了)。线程二进入关键区运行,直至释放 CPU,此时线程一继续运行。但是这是关键区中的数据是被线程二访问后的结果,使得线程一的结果与预计结果不同,并未保证访问关键区这个操作是个原子操作。并未保证互斥,可能出现同时访问关键区。

计算机组成原理

一、填空题

1.

1	0000 000	1	001 0000 0000 0000 0000 0000
---	----------	---	------------------------------

IEEE754 的阶码是用移码表示的,浮点数部分有一个隐藏的 1。

2. 时序发生器、指令寄存器、指令译码器、地址寄存器、微操作控制器

3. ID、EXE、MEM、WB

二、计算机中一般有寄存器 16 个,故需要 4 位来表示这 16 个寄存器。

1. 4 条指令需要 2 位来表示。故 4 条三寄存器指令的操作码部分需要 2 位,三寄存器使用 $4 \times 3 = 12$ 位来寻址。故字长 12 位是不能表示 4 条三寄存器指令的。

2. 255 条单寄存器指令可以用单字指令(字长 12 位)来表示。8 位操作码+4 位寄存器地址。

3. 16 条 0 寄存器指令可以用单字指令(字长 12 位)来表示。4 位操作码。

三、输入/输出方式有:程序查询方式、程序中断方式、DMA 方式、通道方式。各自特点为:

程序查询方式:在这种方式下,输入/输出指令或询问指令测试一台设备的忙闲标志位,决定主存储器和外围设备是否交换一个字符或一个字。一旦 CPU 启动 I/O 设备,便不断查询 I/O 的准备情况,终止原程序的执行。CPU 在反复查询过程中,浪费了宝贵的 CPU 时间;另外,I/O 准备就绪后,CPU 参与数据的传送工作,此时 CPU 也不能执行原程序。可见 CPU 和 I/O 设备串行工作,使主机不能充分发挥效率,外围设备也不能得到合理使用,整个系统的效率很低。

程序中断方式:中断机构引入后,外围设备有了反映其状态的能力,仅当操作正常或异常结束时才中断中央处理机。由于输入/输出操作直接由中央处理器控制,每传送一个字符或一个字,都要发生一次中断。因而,仍然消耗大量中央处理器时间。程序中断方式消除了程序查询方式的忙式测试,提高了 CPU 资源的利用率,但是在响应中断请求后,必须停止现行程序转入中断处理程序并参与数据传输操作。

DMA 方式:主存和 I/O 设备之间有一条数据通路,在主存和 I/O 设备之间成块传送数据过程中,不需要 CPU 干预,实际操作由 DMA 直接执行完成。DMA 不仅设有中断机构,还增加了 DMA 传输控制机构。DMA 方式与程序中断方式相比,又减少了 CPU 对 I/O 的干预,已经从字(字节)为单位的干预减少到以数据块为单位的干预。而且每次 CPU 干预时,并不要做数据副本,仅仅需要发一条启动 I/O 指令以及完成 I/O 结束中断处理。但是每发出一次 I/O 指令,只能读写一个数据块,如果用户希望一次读写多个离散的数据块,并能把它们传送到不同的主存区域,或相反时,则需要由 CPU 分别发出多条启动 I/O 指令及进行多次 I/O 中断处理才能完成。

通道方式:通道方式是 DMA 方式的发展,它又进一步减少了 CPU 对 I/O 操作的干预,减少对多个不连续的数据块而不是仅仅一个数据块及有关管理和控制的干预。通道又称输入/输出处理器。它能完成主存储器和外围设备之间的信息传送,与中央处理器并行地执行操作。

四、1. 多级页表的设计:有 48 位虚拟地址,页面为 4 KB(2^{12}),那么 2^{48} 的虚地址空间由 64 MB(2^{26})个页组成。若以每个页表项占用 4 个字节计算,则需要占用 128 MB(2^{28})连续内存空间存放页表。这还是一个进程的地址空间,系统中有许多进程,这样做存储开销太大。为此页表和页面一样也要进行分页。

将页表 2^{28} 分页, $2^{28}/2^{12}=2^{16}$ 个表项,每个表项 4B,则为 $2^{18}=128$ KB。将 128 KB 再分页,为 $2^{18}/2^{12}=2^6$ 个表项,每个表项 4B,则占用 $2^8=256$ B。至此已经很小了,不用再分了。故此多级页表为二级页表。二级页表示意图为:

