

四川大学

2005 年攻读硕士学位研究生入学考试试题

考试科目：计算机基础（微机原理、C 语言、操作系统）

科目代码：874#

适用专业：计算机系统结构、计算机软件与理论
计算机应用

（试题共 5 页）

（答案必须写在答题纸上，写在试题上不给分）

微机原理

一. 简述：（20 分，每题 5 分）

1. 说明 CPU 中 READY 引脚和时序 T_w 的关系。
2. 说明 CPU 标志位 FLAG 中，IF 位和 CPU 引脚 NMI, INTR 的关系。
3. 说明 8237 DMAC 中，压缩时序的功能。
4. 在 CPU 与外设的数据传送中，直接传送方式是否使用联络线，为什么？

二. 绘制单管动态存储器的电原理图，并说明其工作过程。（10 分）

三. 叙述 8259 中断控制器中，中断结束命令 EOI 的功能。并且说明其分类以及相应的使用场合。（10 分）

四. 8086 CPU 为核心可构成最大组态和最小组态，比较两种组态在构成上的异同。（10 分）

C 语言程序设计(50 分)

一. 读程序, 写执行结果 (8 分)

1.

```
#include <stdio.h>
int main( )
{   int a=4,b=3,c=1,x;
    x = a>b;
    printf("%d\n",x);
    x = --a+b++;
    printf("%d\n",x);
    x = a>b;
    printf("%d\n",x);
    x = c&&b==c;
    printf("%d\n",x);
    return 0;
}
```

2.

```
#include <stdio.h>
int main( )
{   int i,j;
    for(i=0; i<8; i+=2)
    {   for(j=0; j<(8-i)/2; j++)
        printf("+");
        for(j=0; j<=i; j++)
            printf("*");
        printf("\n");
    }
    return 0;
}
```

二. 按要求完成函数。(28 分)

1. 一种最简单的字符串加密方法是将字符串的每一字符加上一个偏移量 offset (不考虑溢出)。请编写相应的编解码函数, 其原型分别为:

char *encode(char *s, int offset);

char *decode(char *s, int offset);

2. 编写函数将两个节点值递增的单向链表归并为一个。要求利用原节点空间。
链表节点定义为:

```
struct node
{   int value;
    struct node *pNext;
};
```

函数原型为:

struct node *merge(struct node *a, struct node *b);

其中 a, b 为要归并的两个单向链表头节点指针, 函数返回值为归并后的单向链表头节点指针。

三. 编写程序模拟数字 LED 显示。(14 分)

编写程序完成如下功能: 用户键盘输入最多四位的整数, 在屏幕上以类似 LED 方式显示。如用户输入为:

8234

则输出:

```
-- -- --  
| | | | |  
| | | | |  
-- -- --  
| | | | |  
| | | | |  
-- -- --
```

其中每个数字字模均为 7 行×4 列。0—9 十个数字的字模顺序地以文本形式存放在文件 C:\font.txt 中。

要求:

1. 输出的各数字间有 1 个空格;
2. 你的程序可以使用 C 语言的标准库函数, 不需要写出头文件;
3. 程序中应有必要的注释。

操作系统 (共计: 50 分)

一. 填空题 (有 (1) 至 (14) 空, 每空 1 分, 共 14 分)

1. 操作系统最基本的特征是 (1) 和 (2), 最主要的任务是 (3)。
2. 在首次适应算法中, 空闲区应以 (4) 的次序链接;
在最佳适应算法中, 空闲区应以 (5) 的次序链接。
3. 程序的并发执行具有与程序的顺序执行不同的特征, 这些特征分别是 (6), (7), (8)。
4. 文件存贮空间的分配可采取多种方式, 其中 (9) 方式可使文件顺序访问的效率最高; (10) 方式则可解决文件存贮空间中的碎片问题, 但却不支持对文件的随机访问; 而 UNIX 采用的则是 (11) 方式。
5. S 为死锁状态的充要条件是 (12), 该充要条件称为死锁定理。
6. 目录的作用在于实现 (13); 目前广泛采用的目录结构是 (14)。

二. 简答题 (每小题 4 分, 共 16 分)

1. 何谓多道程序技术? 实现多道程序技术应解决哪些问题?
2. 何谓死锁? 产生死锁的原因和必要条件是什么?
3. 试从调度性, 并发性, 拥有资源及系统开销方面对进程和线程进行比较。
4. 何谓系统调用? 它与一般的过程调用有何区别?

三. 应用题 (每小题 5 分, 共 20 分)

1. 某车站售票厅, 任何时间最多可容纳 100 名购票者进入, 当售票厅中少于 100 名购票者时, 则厅外的购票者可立即进入, 否则需在外面等待。若把一个购票者看作一个进程, 请回答以下问题:

- (1) 用 PV 操作管理这些并发进程时, 应怎样定义信号量? 写出信号量的初值以及信号量各种取值的含义。
- (2) 根据所定义的信号量, 把应执行的 PV 操作填入下列方框中, 以保证进程能够正确地并发执行。

Cobegin process $p_i (i=1, 2, \dots, n)$

begin

进入售票厅;

购票;

退出;

end

Coend

- (3) 若欲购票者最多为 n 个人, 写出信号量可能的变化范围 (最大值和最小值)
2. 若系统有同类资源 m 个, 被 n 个进程共享, 试问: 当 $m > n$ 和 $m \leq n$, 每个进程最多可申请多少个这类资源而使系统一定不会发生死锁?
3. 已知某分页系统, 主存容量为 64k, 页面大小为 1k, 对一个 4 页大的作业, 其 0, 1, 2, 3 页分别被分配到主存的 2, 4, 6, 7 块中。
 - (1) 将十进制的逻辑地址 3500, 4500 转换成物理地址。
 - (2) 以十进制逻辑地址 3500 为例画出地址变换过程图。

4. 某移动臂磁盘的柱面由外向里从 0 开始顺序编号, 假定当前磁头停在 100 号柱面而且移动方向是向外的, 现有一个请求队列在等待访问磁盘, 访问的柱面号分别为: 190、10、160、80、90、125、30、20、140 和 25。请给出分别采用最短寻找时间优先和电梯调度算法处理上述请求的次序, 并分别计算出它们的平均寻道长度。



王道论坛

www.cskaoan.com

1 一. 读程序, 写执行结果

2 1.

3

4 1

5 6

6 0

7 0

8

9 2.

10

11 +++++*

12 +++++*

13 +++++*

14 +++++*

15

16 二. 按要求完成函数

17 1.

18 char *encode(char *s, int offset)

19 {

20 char *p = s;

21 while (*p)

22 {

23 *p += offset;

24 }

25 return s;

26 }

27 char *decode(char *s, int offset)

28 {

29 char *p = s;

30 while (*p)

31 {

32 *p -= offset;

33 }

34 return s;

35 }

36

37 2.

38 struct node

39 {

40 int value;

41 struct node *pNext;

42 }

43 struct node *merge(struct node *a, struct node *b)

44 {

45 struct node *p;

46 struct node *q;

47 struct node *t;

48 if (a->value <= b->value)

49 {

50 p = a;

51 q = b;

52 }

53 else

54 {

```

61     p = b;
62     q = a;
63 }
64 t = p;
65
66 while (q)
67 {
68     if (p->pNext == NULL)
69     {
70         p->pNext = q;
71         break;
72     }
73
74     if (q->value < p->pNext->value)
75     {
76         struct node *k = q->pNext;
77         q->pNext = p->pNext;
78         p->pNext = q;
79         q = k;
80
81         continue;
82     }
83
84     p = p->pNext;
85 }
86
87 return t;
88 }
89
90

```

三. 编写程序完成模拟数字LED显示

```

91 char font[10][7][5];
92 int n;
93 int digit[4];
94 int length = 0;
95
96 /* 装载字体文件 */
97 void load_fonts()
98 {
99     int i, j, k;
100     FILE *file;
101
102     file = fopen("c:\\font.txt", "r");
103     if (file == NULL)
104     {
105         printf("打开字体文件 font.txt 错误\n");
106         exit(1);
107     }
108
109     for (k = 0; k < 10; k++)
110     {
111         for (i = 0; i < 7; i++)
112         {
113             char line[10];
114             fgets(line, 10, file);
115             for (j = 0; j < 4; j++)
116             {
117                 font[k][i][j] = line[j];
118             }
119         }
120     }

```

```

121         font[k][i][4] = 0;
122     }
123 }
124
125 fclose(file);
126 }
127
128 /* 将输入的整数解码为单个的数字 */
129 void decode()
130 {
131     int i;
132     int m = n;
133     for (i = 0; i < 4; i++)
134     {
135         digit[i] = m % 10;
136         m = m / 10;
137
138         if (m==0)
139             break;
140     }
141
142     length = i + 1;
143     if (length >= 4)
144         length = 4;
145 }
146
147 /* 显示LED */
148 void display()
149 {
150     int i, j;
151     for (i = 0; i < 7; i++)
152     {
153         for (j = length-1; j >= 0; j--)
154         {
155             printf("%s", font[digit[j]][i]);
156             printf(" ");
157         }
158         printf("\n");
159     }
160 }
161
162 int main()
163 {
164
165     load_fonts();
166
167     scanf("%d", &n);
168
169     decode();
170
171     display();
172 }
173

```


操作系统试题答案

一. 填空题 (每空 1 分, 共 14 分)

1. (1) 并发; (2) 资源共享; (3) 管理资源。
2. (4) 空闲区地址从小到大; (5) 空闲区大小从小到大 (递增)。
3. (6) 间断性; (7) 失去封闭性; (8) 不可再现性。
4. (9) 连续分配; (10) 隐式链接分配; (11) 混合 (索引) 分配;
5. (12) 当且仅当 S 状态的资源分配图是不可完全简化的。
6. (13) 按名存取; (14) 树形目录结构。

二. 简答题 (每小题 4 分, 共 16 分)

1. 答: 多道程序技术是指在内存中同时存放若干个作业, 并使它们共享系统资源, 同时运行的技术。

实现此技术需要解决的问题:

- (1) 如何为每道程序分配主存空间; (2) CPU 的调度和分配; (3) I/O 设备的调度和分配; (4) 信息共享和保护; (5) 在计算机系统中必须设置一组使彼此间能协调运行的软件, 用以对上述问题进行妥善、有效地处理。
2. 答: 死锁是指多个进程因竞争资源而形成的一种僵局, 若无外力的作用, 这些进程将无法再向前推进。

产生死锁的原因是竞争资源和进程推进顺序非法。

产生死锁的必要条件是: 互斥条件, 请求和保持条件, 不剥夺条件和环路等待条件。

3. 答: 进程和线程之间就上述问题比较如下:

(1) 调度性: 在传统的 OS 中, 拥有资源的基本单位和独立调度, 分配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, 而把进程作为资源拥有的基本单位。

(2) 并发性: 在引入线程的 OS 中, 不仅进程间可以并发执行, 而且在一个进程的多个线程间也可以并发执行, 因此它比传统的 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、一组寄存器和堆栈) 外,

本身基本不拥有系统资源，但它可访问其隶属进程的资源。

(4) 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，如内存空间和 I/O 设备等；进程切换时所保存和设置的现场信息也要明显多于线程，因此，OS 在创建、撤消、切换进程时所付出的开销将明显大于线程。另外，由于隶属于同一进程的多个线程共享同一地址空间和该进程的所有已打开的文件，从而使它们之间的同步和通信的实现也比进程更方便。

4. 答：系统调用是 OS 提供给用户程序的唯一接口，即它是 OS 内核中提供的一些系统子程序。用户可通过特殊的系统调用命令（也称作访管指令）来调用这些子程序，从而使用户在自己的程序中可获得 OS 提供的服务，如：打开文件，创建子进程等。

系统调用与一般的过程调用的区别主要有以下几点：

(1) 运行在不同的系统状态：一般的调用程序和被调用的程序都运行在相同的状态——系统态或用户态；而对系统调用，其调用程序是运行在用户态，而被调用程序则是运行在系统态。

(2) 通过软中断进入：一般的过程调用可通过过程调用语句直接由调用过程转向被调用过程；而系统调用则必须通过执行系统调用命令（也称作访管指令），由软中断（或陷入机制）转向相应的系统调用处理程序，同时 CPU 的执行状态将从用户态转换为系统态。

(3) 返回问题：一般的过程调用在被调用过程执行完后，将直接返回到调用过程继续执行；而对系统调用，如果用抢占方式，则在被调用过程执行完后，必须先对要求运行的进程做优先权分析，只当调用进程仍具最高优先权时，才返回到调用进程继续执行；否则，将引起重新调度。

三、应用题（每小题 5 分，共 20 分）

1. 解：(1) 应定义一个信号量 S ， S 的初值为 100，

当 $0 < S \leq 100$ 时，允许厅外的购票者进入；

当 $S=0$ 时，厅内已有 100 人，欲购票者暂不能进入；

当 $S < 0$ 时， $|S|$ 表示等待进入者的人数；

(2) 用 PV 操作管理时保证进程正确执行的程序如下：


```

Cobegin    process  $p_i (i=1, 2, 3, \dots, n)$ 
            begin  $p(s)$ :
                进入售票厅;
                购票;
                退出;
                 $v(s)$ 
            end;
Coend;
    
```

(2) 若购票者最多为 n 人, 则信号量 S 的变化范围: $m-n \leq s \leq 100$

2. 解: 假设每个进程最多可以申请 x 个资源, 为保证系统不发生死锁, 应该使下列不等式成立:

$$n(x-1)+1 \leq m$$

解上述不等式: $nx \leq n+m-1$

$$x \leq 1 + \frac{m-1}{n}$$

于是可解得: $x = \begin{cases} 1 & \text{当 } m \leq n \text{ 时} \\ 1 + \lceil \frac{m-1}{n} \rceil & \text{当 } m > n \text{ 时} \end{cases}$

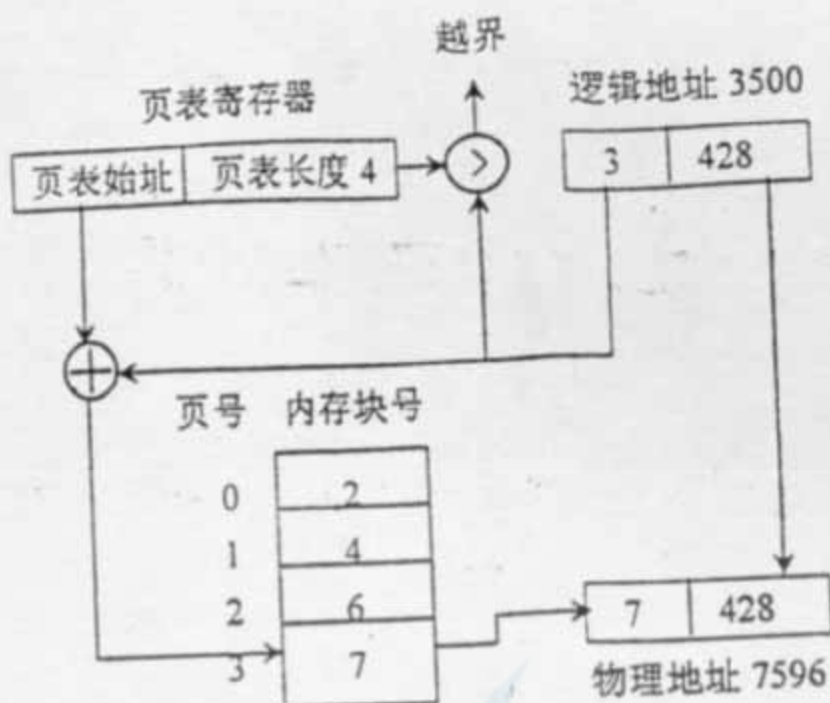
3. 解: (1) 对上述逻辑地址, 可先计算出它们的页号和页内地址, 然后通过页表转换成对应的物理地址。

①逻辑地址 3500: $p = \lfloor 3500/1k \rfloor = 3$, $d = \lfloor 3500/1k \rfloor_{\text{取余}} = 428$, 由页号可查页表找到对应的物理块号为 7, 故物理地址为: $7*1k+428=7596$

②逻辑地址 4500: $p = \lfloor 4500/1k \rfloor_{\text{取整}} = 4$, $d = \lfloor 4500/1k \rfloor_{\text{取余}} = 404$

因为页号 $p=4$ 不小于页表长度 4, 就产生越界中断。

(2) 逻辑地址 3500 的地址变换过程如下图所示:



4. 解：处理上述请求的次序以及平均寻道时间如下表示：

采用最短寻找时间优先算法时处理各请求的次序为：

90、80、125、140、160、190、30、25、20、10

平均寻道时间为：

$$[(100-90)+(90-80)+(125-80)+(140-125)+(160-140)+(190-160) \\ +(190-30)+(30-25)+(25-20)+(20-10)] \div 10 = 21$$

采用电梯调度算法时的次序为：

90、80、30、25、20、10、125、140、160、190

平均寻道时间为：(10+10+50+5+5+10+115+15+20+30)÷10=27