



bbs.kaoyan.com

1.用预处理指令**#define**声明一个常数，用以表明1年中有多少秒（忽略闰年问题）

```
#define SECONDS_PER_YEAR (60 * 60 * 24 * 365)UL
```

2.嵌入式系统中经常要用到无限循环，你怎么样用**C**编写死循环呢？ `while(1){}`或者`for(;;){}`

3.用变量**a**给出下面的定义

a)一个整型数（An integer）

b)一个指向整型数的指针（A pointer to an integer）

c)一个指向指针的指针，它指向的指针是指向一个整型数（
A pointer to a pointer to an integer）

d)一个有10个整型数的数组（An array of 10 integers）

e)一个有10个指针的数组，该指针是指向一个整型数的。（An array of 10 pointers to integers）

f)一个指向有10个整型数数组的指针（A pointer to an array of 10 integers）

g)一个指向函数的指针，该函数有一个整型参数并返回一个整型数 (**A pointer to a function that takes an integer as an argument and returns an integer**)

h)一个有10个指针的数组，该指针指向一个函数，该函数有一个整型参数并返回一个整型数 (**An array of ten pointers to functions that take an integer argument and return an integer**)

答案是：

a) `int a;` // An integer

b) `int *a;` // A pointer to an integer

c) `int **a;` // A pointer to a pointer to an integer

d) `int a[10];` // An array of 10 integers

e) `int *a[10];` // An array of 10 pointers to integers

f) `int (*a)[10];` // A pointer to an array of 10 integers

g) `int (*a)(int);` // A pointer to a function a that takes an integer argument and returns an integer

h) `int (*a[10])(int);` // An array of 10 pointers to functions that take an integer argument and return an integer

4.关键字**static**的作用是什么？

这个简单的问题很少有人能回答完全。在C语言中，关键字**static**有三个明显的作用：

1)在函数体，一个被声明为静态的变量在这一函数被调用过程中维持其值不变。

2)

在模块内（但在函数体外），一个被声明为静态的变量可以被模块内所用函数访问，但不能被模块外其它函数访问。它是一个本地的全局变量。

3)

在模块内，一个被声明为静态的函数只可被这一模块内的其它函数调用。那就是，这个函数被限制在声明它的模块的本地范围内使用。

大多数应试者能正确回答第一部分，一部分能正确回答第二部分，同是很少的人能懂得第三部分。这是一个应试者的严重的缺点，因为他显然不懂得本地化数据和代码范围的好处和重要性。

5.关键字**const**有什么含意?表示常量 不可以修改的变量。

```
const int a;
```

```
int const a;
```

```
const int *a;
```

```
int * const a;
```

```
int const * a const;
```

前两个的作用是一样，a是一个常整型数。第三个意味着a是一个指向常整型数的指针（也就是，整型数是不可修改的，但指针可以）。第四个意思a是一个指向整型数的常指针（也就是说，指针指向的整型数是可以修改的，但指针是不可修改的）。最后一个意味着

a

是一个指向常整型数的常指针（也就是说，指针指向的整型数是不可修改的，同时指针也是不可修改的）。如果应试者能正确回答这些问题，那么他就给我留下了一个好印象。顺带提一句，也许你可能会问，即使不用关键字

const，也还是能很容易写出功能正确的程序，那么我为什么还要如此看重关键字const呢？我也如下的几下理由：

1)关键字const

的作用是为给读你代码的人传达非常有用的信息，实际上，声明一个参数为常量是为了告诉了用户这个参数的应用目的。如果你曾花很多时间清理其它人留下的垃圾，你就会很快学会感谢这点多余的信息。（当然，懂得用

const的程序员很少会留下的垃圾让别人来清理的。)

2)通过给优化器一些附加的信息，使用关键字const也许能产生更紧凑的代码。

3)合理地使用关键字const

可以使编译器很自然地保护那些不希望被改变的参数，防止其被无意的代码修改。简而言之，这样可以减少bug的出现。

6.程序什么时候应该使用线程，什么时候单线程效率高。

答：1. 耗时的操作使用线程，提高应用程序响应。

2. 并行操作时使用线程，如C/S架构的服务器端并发线程响应用户的请求。

3. 多CPU系统中，使用线程提高CPU利用率

4. 改善程序结构。一个既长又复杂的进程可以考虑分为多个线程，成为几个独立或半独立的运行部分，这样的程序会利于理解和修改。

其他情况都使用单线程。

7.一般数据库若出现日志满了，会出现什么情况，是否还能使用？

答：只能执行查询等读操作，不能执行更改，备份等写操作，原因是任何写操作都要记录日志。也就是说基本上处于不能使用的状态。

第三次握手：客户端收到服务器的SYN + ACK包，向服务器发送确认包ACK(ack=k+1)，此包发送完毕，客户端和服务器进入ESTABLISHED状态，完成三次握手。

8.ICMP是什么协议,处于哪一层?答：Internet控制报文协议，处于网络层（IP层）（ping命令基于这个协议）

9.winsock建立连接的主要实现步骤?

答：服务器端：socket()建立套接字，绑定（bind）并监听（listen），用accept（）等待客户端连接。

客户端：socket()建立套接字，连接（connect）服务器，连接上后使用send()和recv（），在套接字上写读数据，直至数据交换完毕，closesocket()关闭套接字。

服务器端：accept

（）发现有客户端连接，建立一个新的套接字，自身重新开始等待连接。该新产生的套接字使用

send()和recv（）写读数据，直至数据交换完毕，closesocket()关闭套接字。

10.IP组播有那些好处？

答：Internet

上产生的许多新的应用，特别是高带宽的多媒体应用，带来了带宽的急剧消耗和网络拥挤问题。组播是一种允许一个或多个发送者（组播源）发送单一的数据包到多个接收者（一次的，同时的）的网络技术。组播可以大大的节省网络带宽，因为无论有多少个目标地址，在整个网络的任何一条链路上只传送单一的数据包。所以说组播技术的核心就是针对如何节约网络资源的前提下保证服务质量。

11.引用与指针有什么区别？

1)引用必须被初始化，指针不必。 2)

引用初始化以后不能被改变，指针可以改变所指的对象。

3)不存在指向空值的引用，但是存在指向空值的指针。

12.TCP/IP建立连接的过程?(3-way shake)

答：在TCP/IP协议中，TCP协议提供可靠的连接服务，采用三次握手建立一个连接。

第一次握手：建立连接时，客户端发送syn包(syn=j)到服务器，并进入SYN_SEND状态，等待服务器确认；

第二次握手：服务器收到syn包，必须确认客户的SYN（ack=j+1），同时自己也发送一个SYN包（syn=k），即SYN+ACK包，此时服务器进入SYN_RECV状态；

13、局部变量能否和全局变量重名？

答：能，局部会屏蔽全局。要用全局变量，需要使用"::"

局部变量可以与全局变量同名，在函数内引用这个变量时，会用到同名的局部变量，而不会用到全局变量。对于有些编译器而言，在同一个函数内可以定义多个同名的局部变量，比如在两个循环体内都定义一个同名的局部变量，而那个局部变量的作用域就在那个循环体内。

14、如何引用一个已经定义过的全局变量？

答：extern

可以用引用头文件的方式，也可以用extern

关键字，如果用引用头文件方式来引用某个在头文件中声明的全局变理，假定你将那个变写错了，那么在编译期间会报错，如果你用

extern方式引用时，假定你犯了同样的错误，那么在编译期间不会报错，而在连接期间报错。

15.描述实时系统的基本特性 在特定时间内完成特定的任务，实时性与可靠性。

16.全局变量和局部变量在内存中是否有区别？如果有，是什么区别？

全局变量储存在静态数据库，局部变量在堆栈。

17.什么是平衡二叉树？ 左右子树都是平衡二叉树且左右子树的深度差值的绝对值不大于1。

18.堆栈溢出一般是由什么原因导致的？ 没有回收垃圾资源。

19.什么函数不能声明为虚函数？ constructor函数不能声明为虚函数。

20.冒泡排序算法的时间复杂度是什么？ 时间复杂度是 $O(n^2)$ 。

21.写出float x与“零值”比较的if语句。 `if(x<0.000001&& x>-0.000001)`

22.Internet采用哪种网络协议？该协议的主要层次结构？

Tcp/Ip协议 主要层次结构为：应用层/传输层/网络层/数据链路层/物理层。

23.Internet物理地址和IP地址转换采用什么协议？ ARP (Address Resolution Protocol)

(地址解析協議)

24.IP地址的编码分为哪俩部分？ IP地址由两部分组成，网络号和主机号。不过是要和“子网掩码”按位与上之后才能区分哪些是网络位哪些是主机位。

25.用户输入M,N值，从1至N开始顺序循环数数，每数到M输出该数值，直至全部输出。写出C程序。

循环链表，用取余操作做

26.不能做switch()的参数类型是： switch的参数不能为实型。

27.请写出下列代码的输出内容

```
# include<stdio.h>
```

```
main()
```

```
{
```

```
int a,b,c,d;
```

```
a=10;
```

```
b=a++;
```

```
c=++a;
```

```
d=10*a++;
```

```
printf("b, c, d: %d, %d, %d", b, c, d) ;
```

```
return 0;
```

```
}
```

答：10， 12， 120

28.写出下列代码的输出内容

```
# include<stdio.h>
```

```
int inc(int a)
```

```
{
```

```
return(++a);
```

```
}
```

```
int multi(int*a,int*b,int*c)
```

```
{
```

```
return(*c=*a**b);
```

```
}
```

```
typedef int(FUNC1)(int in);
```

```
typedef int(FUNC2) (int*,int*,int*);

void show(FUNC2 fun,int arg1, int*arg2)

{

    INCp=&inc;

    int temp =p(arg1);

    fun(&temp,&arg1, arg2);

    printf("%d\n",*arg2);

}

main()

{

    int a;

    show(multi,10,&a);

    return 0;
```

}

1.对于一个频繁使用的短小函数,在C语言中应用什么实现,在C++中应用什么实现?答: c
用宏定义, c++用inline

2.直接链接两个信令点的一组链路称作什么?答: PPP点到点连接

3.软件测试都有那些种类?黑盒: 针对系统功能的测试 白盒: 测试函数功能, 各函数接口

4.确定模块的功能和模块的接口是在软件设计的那个阶段完成的?概要设计阶段

三.选择题:

1.Ethternet链接到Internet用到以下那个协议? A.HDLC;B.ARP;C.UDP;D.TCP;E.ID

2.属于网络层协议的是: A.TCP;B.IP;C.ICMP;D.X.25

3.Windows消息调度机制是: A.指令队列;B.指令堆栈;C.消息队列;D.消息堆栈;

4.unsigned short hash(unsigned short key)

{

```
return (key>>)%256
```

```
}
```

请问hash(16),hash(256)的值分别是: A.1.16;B.8.32;C.4.16;D.1.32

四.找错题:

1.请问下面程序有什么错误?

```
int a[60][250][1000],i,j,k;
```

```
for(k=0;k<=1000;k++)
```

```
for(j=0;j<250;j++)
```

```
for(i=0;i<60;i++)
```

```
    a[i][j][k]=0;
```

把循环语句内外换一下

2.#define Max_CB 500

```
void LmiQueryCSmd(Struct MSgCB * pmsg)
```

```
{  
  
unsigned char ucCmdNum;  
  
.....  
  
for(ucCmdNum=0;ucCmdNum<Max_CB;ucCmdNum++)  
  
{  
  
.....;  
  
}
```

死循环

3.以下是求一个数的平方的程序,请找出错误:

```
#define SQUARE(a)((a)*(a))  
  
int a=5;  
  
int b;  
  
b=SQUARE(a++);
```

4.typedef unsigned char BYTE

```
int examply_fun(BYTE gt_len; BYTE *gt_code)
```

```
{
```

```
    BYTE *gt_buf;
```

```
    gt_buf=(BYTE *)MALLOC(Max_GT_Length);
```

```
    .....
```

```
    if(gt_len>Max_GT_Length)
```

```
    {
```

```
        return GT_Length_ERROR;
```

```
    }
```

```
    .....
```

```
}
```

五.问答题:

1.IP Phone的原理是什么?IPV6

2.TCP/IP通信建立的过程怎样，端口有什么作用？三次握手，确定是哪个应用程序使用该协议

3.1号信令和7号信令有什么区别，我国某前广泛使用的是那一种？

4.列举5种以上的电话新业务？

1.进程和线程的差别。

线程是指进程内的一个执行单元,也是进程内的可调度实体.与进程的区别:

(1)调度：线程作为调度和分配的基本单位，进程作为拥有资源的基本单位

(2)并发性：不仅进程之间可以并发执行，同一个进程的多个线程之间也可并发执行

(3)

拥有资源：进程是拥有资源的一个独立单位，线程不拥有系统资源，但可以访问隶属于进程的资源

.

(4)

系统开销：在创建或撤消进程时，由于系统都要为之分配和回收资源，导致系统的开销明显大于创建或撤消线程时的开销。

2. Heap与stack的差别。

Heap是堆，stack是栈。Stack的空间由操作系统自动分配/释放，Heap上的空间手动分配/释放。

Stack空间有限，Heap是很大的自由存储区。C中的malloc函数分配的内存空间即在堆上,C++中对应的是new操作符。

程序在编译期对变量和函数分配内存都在栈上进行，且程序运行过程中函数调用时参数的传递也在栈上进行。

3.如果只想让程序有一个实例运行，不能运行两个。像winamp一样，只能开一个窗口，怎样实现？

用内存映射或全局原子（互斥变量）、查找窗口句柄...FindWindow，互斥，写标志到文件或注册表,共享内存。

4.存储过程是什么？有什么用？有什么优点？

就是一堆sql

的集合，可以建立非常复杂的查询，编译运行，所以运行一次后，以后再运行速度比单独执行SQL快很多

5.网络编程中设计并发服务器，使用多进程与多线程，请问有什么区别？

1，进程：子进程是父进程的复制品。子进程获得父进程数据空间、堆和栈的复制品。

2

，线程：相对与进程而言，线程是一个更加接近于执行体的概念，它可以与同进程的其他线程共享数据，但拥有自己的栈空间，拥有独立的执行序列。

两者都可以提高程序的并发度，提高程序运行效率和响应时间。

线程和进程在使用上各有优缺点：线程执行开销小，但不利于资源管理和保护；而进程正相反。同时，线程适合于在SMP机器上运行，而进程则可以跨机器迁移。

6. Windows下的内存是如何管理的？

7.用一个语句实现x是否为2的若干次幂的判断：

```
int i = 512;
```

```
cout << boolalpha << ((i & (i - 1)) ? false : true) << endl;
```

8.下面三个有什么区别？

char * const p;char const * p; const char *p

答：char * const p; //常量指针，p的值不可以修改

char const * p; //指向常量的指针，指向的常量值不可以改

const char *p; //和char const *p一样

9.一个32位的机器,该机器的指针是多少位

指针是多少位只要看地址总线的位数就行了。80386以后的机子都是32的数据总线。所以指针的位数就是4个字节了。

10.main()

{

int a[5]={1,2,3,4,5};

int *ptr=(int *)(&a+1);

printf("%d,%d",*(a+1),*(ptr-1));

```
}
```

输出： 2,5

*(a+1) 就是a[1], *(ptr-1)就是a[4],执行结果是2, 5

&a+1不是首地址+1, 系统会认为加一个a数组的偏移, 是偏移了一个数组的大小 (本例是5个int)

```
int *ptr=(int *)(&a+1);
```

则ptr实际是&(a[5]),也就是a+5

原因如下:

&a是数组指针, 其类型为int (*)[5];

而指针加1要根据指针类型加上一定的值,

不同类型的指针+1之后增加的大小不同

a是长度为5的int数组指针, 所以要加5*sizeof(int)

所以ptr实际是a[5]

但是ptr与(&a+1)类型是不一样的(这点很重要)

所以prt-1只会减去sizeof(int*)

a,&a的地址是一样的,但意思不一样,a是数组首地址,也就是a[0]的地址,&a是对象(数组)首地址,a+1是数组下一元素的地址,即a[1],&a+1是下一个对象的地址,即a[5]。

11.交换两个变量的值,不使用第三个变量。即a=3,b=5,交换之后a=5,b=3;

a = a + b;b = a - b;a = a - b;//第一种方法

a = a^b;b = a^b;a = a^b; //第二种.

a^=b^=a^=b; //第三种,后两种只能用于字符和整型

12.c和c++中的struct有什么不同?

c和c++中struct的主要区别是c中的struct不可以含有成员函数,而c++中的struct可以。c++中struct和class的主要区别在于默认的存取权限不同,struct默认为public,而class默认为private

13.举几种进程的同步机制,并比较其优缺点。

原子操作信号量机制、自旋锁、管程,会合,分布式系统

14.进程之间通信的途径:共享存储系统、消息传递系统、管道（以文件系统为基础）

15.进程死锁的原因:资源竞争及进程推进顺序非法

16.死锁的4个必要条件:互斥、请求保持、不可剥夺、环路

17.死锁的处理：鸵鸟策略、预防策略、避免策略、检测与解除死锁

18.操作系统中进程调度策略有哪几种？ FCFS(先来先服务)，优先级，时间片轮转，多级反馈

19.数组和链表的区别数组：数据顺序存储，固定大小；

链表：数据可以随机存储，大小可动态改变

20.ISO的七层模型是什么？ tcp/udp是属于哪一层？ tcp/udp有何优缺点？

应用层、表示层、会话层、运输层、网络层、物理链路层、物理层、tcp /udp属于运输层

TCP服务提供了数据流传输、可靠性、有效流控制、全双工操作和多路复用技术等。

与TCP不同，UDP并不提供对IP协议的可靠机制、流控制以及错误恢复功能等。由于UDP比较简单，UDP头包含很少的字节，比TCP负载消耗少。

tcp:提供稳定的传输服务，有流量控制，缺点是包头大，冗余性不好。

udp:不提供稳定的服务，包头小，开销小。

面试题:线程与进程的区别和联系?线程是否具有相同的堆栈? dll是否有独立的堆栈?

(1) 进程是死的, 只是一些资源的集合, 真正的程序执行都是线程来完成的, 程序启动的时候操作系统就帮你创建了一个主线程。

(2) 每个线程有自己的堆栈。

(3) DLL中有没有独立的堆栈, 这个问题不好回答, 或者说这个问题本身是否有问题。因为DLL中的代码是被某些线程所执行, 只有线程拥有堆栈, 如果DLL中的代码是EXE中的线程所调用, 那么这个时候是不是说这个DLL没有自己独立的堆栈? 如果DLL中的代码是由DLL自己创建的线程所执行, 那么是不是说DLL有独立的堆栈?

以上讲的是堆栈, 如果对于堆来说, 每个DLL有自己的堆, 所以如果是从DLL中动态分配的内存, 最好是从DLL中删除, 如果你从DLL中分配内存, 然后在EXE中, 或者另外一个DLL中删除, 很有可能导致程序崩溃。

■ 结尾