1main()

{

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

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

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

}

答案：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prt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].

2.c和c++中的struct有什么不同？

答案：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

3.类的静态成员和非静态成员有何区别？

答案：类的静态成员每个类只有一个，非静态成员每个对象一个

4操作系统中进程调度策略有哪几种？

答案：FCFS(先来先服务)，优先级，时间片轮转，多级反馈

5.纯虚函数如何定义？使用时应注意什么？

答案：virtualvoid f()=0; 是接口，子类必须要实现

6数组和链表的区别

答案：数组：数据顺序存储，固定大小连表：数据可以随机存储，大小可动态改变

7：int main() { intx=3; printf("%d",x); return1; } 问函数既然不会被其它函数调用，为什么要返回1？

答案：mian中，c标准认为0表示成功，非0表示错误。具体的值是某中具体出错信息

8要对绝对地址0x100000赋值，我们可以用 (unsigned int\*)0x100000 =1234; 那么要是想让程序跳转到绝对地址是0x100000去执行，应该怎么做？

答案：\*((void(\*)( ))0x100000 ) ( ); 首先要将0x100000强制转换成函数指针,即: (void(\*)())0x100000 然后再调用它:\*((void (\*)())0x100000)(); 用typedef可以看得更直观些: typedef void(\*)() voidFuncPtr; \*((voidFuncPtr)0x100000)();

9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?

答案：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。每个线程有自己的堆栈。 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

10unsigned short A = 10; printf("~A = %u\n", ~A); charc=128; printf("c=%d\n",c); 输出多少？并分析过程

答案：第一题，～A ＝0xfffffff5,int值 为－11，但输出的是uint。所以输出4294967285 第二题，c＝0x10,输出的是int，最高位为1，是负数，所以它的值就是0x00的补码就是128，所以输出－128。这两道题都是在考察二进制向int或uint转换时的最高位处理。

11直接链接两个信令点的一组链路称作什么?

答案：PPP点到点连接

12unsigned char \*p1; unsigned long\*p2; p1=(unsigned char \*)0x801000; p2=(unsigned long \*)0x810000; 请问p1+5= ; p2+5= ;

答案：801005； 810028。不要忘记了这个是16进制的数字，p2要加40变为16进制就是

12Ethternet链接到Internet用到以下那个协议?

从《计算机网络》这本书的章节设置来看，以太网被排在链路层，我们假设以太网工作在链路层，链路层依靠mac地址寻址，而因特网依靠IP地址寻址，arp协议负责把IP地址解析为mac地址，综上，以太网接入因特网依靠arp协议

13static全局变量与普通的全局变量有什么区别？

全局变量(外部变量)的说明之前再冠以static 就构成了静态的全局变量。全局变量本身就是静态存储方式，静态全局变量当然也是静态存储方式。这两者在存储方式上并无不同。这两者的区别虽在于非静态全局变量的作用域是整个源程序， 当一个源程序由多个源文件组成时，非静态的全局变量在各个源文件中都是有效的。而静态全局变量则限制了其作用域，即只在定义该变量的源文件内有效， 在同一源程序的其它源文件中不能使用它。由于静态全局变量的作用域局限于一个源文件内，只能为该源文件内的函数公用，因此可以避免在其它源文件中引起错误。

14请找出下面代码中的所以错误说明：以下代码是把一个字符串倒序，

如“abcd”倒序后变为“dcba”

1、＃include"string.h"

2、main()

3、{

4、 char\*src="hello,world";

5、 char\*dest=NULL;

6、 intlen=strlen(src);

7、 dest=(char\*)malloc(len);

8、 char\* d=dest;

9、 char\* s=src[len];

10、 while(len--!=0)

11、 d++=s--;

12、 printf("%s",dest);

13、 return 0;

14、}

答案：还要加上＃include<stdio.h> int main(){ char\* src = "hello,world"; int len =strlen(src); char\* dest = (char\*)malloc((len+1)\*sizeof(char)); //要为\0分配一个空间 char\* d = dest; char\* s = &src[len-1]; //指向最后一个字符 while( len-- != 0 )\*d++=\*s--; \*d = 0; //尾部要加\0 printf("%s\n",dest); free(dest);// 使用完，应当释放空间，以免造成内存汇泄露 return 0;}

15设int arr[]={6,7,8,9,10}; int\*ptr=arr; \*(ptr++)+=123; printf(＂%d,%d＂,\*ptr,\*(++ptr));( ) 10分答案：8，

8。这道题目的意义不大，因为在不同的编译器里printf的参数的方向是不一样的，在vc6.0下是从有到左，这里先\*(++ptr) 后\*pt，于是结果为8，8

16在OSI 7 层模型中,网络层的功能有( )

A.确保数据的传送正确无误 B.确定数据包如何转发与路由 C.在信道上传送比特流 D.纠错与流控

17FDDI 使用的是\_\_\_局域网技术。( c)

A.以太网; B.快速以太网; C.令牌环; D.令牌总线。

18.下面那种LAN 是应用CSMA/CD协议的（c)

A.令牌环 B.FDDI C.ETHERNET D.NOVELL

19以下说法错误的是(多) ( bd)

A.中继器是工作在物理层的设备 B.集线器和以太网交换机工作在数据连路层 C.路由器是工作在网络层的设备 D.桥能隔离网络层广播

20.LAN Switch 在网络层次模型中的地位(b )

A.物理层 B.链路层 C.网络层 D.以上都不是

21防火墙是怎么实现的？

22 ipsec为什么是三层的。L2tp为什么是二层的

23ospf中包的ttl值是多少？ 答案：1，因为224.0.0.5/224.0.0.6是本地链路组播

24IP 包头几个字节？加上数据部分几个字节答案：应该是问IPV4，一般20字节，加上源路由选择等后最大60字节，加上数据部分还是根据链路MTU决定

25 在ARM系统中，函数调用的时候，参数是通过哪种方式传递的?

参数<=4时候，通过R0~R3传递，>4的通过压栈方式传递

26中断(interrupt,如键盘中断)与异常(exception,如除零异常)有何区别?

异常：在产生时必须考虑与处理器的时钟同步，实践上，异常也称为同步中断。在处理器执行到由于编程失误而导致的错误指令时，或者在执行期间出现特殊情况(如缺页)，必须靠内核处理的时候，处理器就会产生一个异常。

所谓中断应该是指外部硬件产生的一个电信号，从cpu的中断引脚进入，打断cpu当前的运行；

所谓异常，是指软件运行中发生了一些必须作出处理的事件，cpu自动产生一个陷入来打断当前运行，转入异常处理流程。

27 优先级反转问题在嵌入式系统中是一中严重的问题，必须给与足够重视。

a) 首先请解释优先级反转问题

b) 很多RTOS提供优先级继承策略(Priority inheritance)和优先级天花板策略(Priority ceilings)用来解决优先级反转问题，请讨论这两种策略。

高优先级任务需要等待低优先级任务释放资源，而低优先级任务又正在等待中等优先级任务的现象叫做优先级反转

优先级继承策略(Priority inheritance):继承现有被阻塞任务的最高优先级作为其优先级，任务退出临界区，恢

复初始优先级。

优先级天花板策略(Priority ceilings):控制访问临界资源的信号量的优先级天花板。

优先级继承策略对任务执行流程的影响相对教小，因为只有当高优先级任务申请已被低优先级任务占有的临界资源

这一事实发生时，才抬升低优先级任务的优先级。

28直接链接两个信令点的一组链路称作什么? 　　PPP点到点连接

29给两个变量，如何找出一个带环单链表中是什么地方出现环的？

一个递增一，一个递增二，他们指向同一个接点时就是环出现的地方

30指出下面代码的输出，并解释为什么。

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prt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].

31.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

32列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