linux服务器开发架构师 Lv2
2019年07月02日 阅读 9468

关注

60道30K+C++工程师面试必问面试题

1.在C++ 程序中调用被C 编译器编译后的函数，为什么要加extern “C”?

答：首先，extern是C/C++语言中表明函数和全局变量作用范围的关键字，该关键字告诉编译器，其声明的函数和变量可以在本模块或其它模块中使用。

通常，在模块的头文件中对本模块提供给其它模块引用的函数和全局变量以关键字extern声明。extern "C"是连接申明(linkage declaration),被extern "C"修饰的变量和函数是按照C语言方式编译和连接的。作为一种面向对象的语言，C++支持函数重载，而过程式语言C则不支持。函数被C++编译后在符号库中的名字与C语言的不同。例如，假设某个函数的原型为：void foo(int x, int y);该函数被C编译器编译后在符号库中的名字为_foo，而C++编译器则会产生像_foo_int_int之类的名字。这样的名字包含了函数名、函数参数数量及类型信息，C++就是靠这种机制来实现函数重载的。

所以，可以用一句话概括extern “C”这个声明的真实目的:解决名字匹配问题，实现C++与C的混合编程。

2.头文件中的ifndef/define/endif有什么作用?

答：这是C++预编译头文件保护符，保证即使文件被多次包含，头文件也只定义一次。

3. ＃include<file.h> 与 ＃include "file.h"的区别?

答：前者是从标准库路径寻找和引用file.h，而后者是从当前工作路径搜寻并引用file.h。

4.评价一下C/C++各自的特点

答：C语言是一种结构化语言，面向过程，基于算法和数据结构，所考虑的是如何通过一个过程或者函数从输入得到输出；

5．const 有什么用途?

答：在C/C++中，（1）可以定义const常量，（2）修饰函数的返回值和形参；

在C++中，还可以修饰函数的定义体，定义类的const成员函数。被const修饰的东西受到强制保护，可以预防意外的变动，提高了程序的健壮性。

6．const和#define有什么区别?

答：（1）const和#define都可以定义常量，但是const用途更广。

（2）const 常量有数据类型，而宏常量没有数据类型。编译器可以对前者进行类型安全检查。而对后者只进行字符替换，没有类型安全检查，并且在字符替换可能会产生意料不到的错误。

（3）有些集成化的调试工具可以对const 常量进行调试，但是不能对宏常量进行调试。

7．关于sizeof小结的。

答：sizeof计算的是在栈中分配的内存大小。

（1）sizeof不计算static变量占得内存；

（2）32位系统的指针的大小是4个字节，64位系统的指针是8字节，而不用管指针类型；

（3）char型占1个字节，int占4个字节，short int占2个字节

long int占4个字节，float占4字节，double占8字节，string占4字节

一个空类占1个字节，单一继承的空类占1个字节，虚继承涉及到虚指针所以占4个字节

（4）数组的长度：

若指定了数组长度，则不看元素个数，总字节数=数组长度*sizeof（元素类型）

若没有指定长度，则按实际元素个数类确定

Ps：若是字符数组，则应考虑末尾的空字符。

在默认情况下，为方便对结构体内元素的访问和管理，当结构体内元素长度小于处理器位数的时候，便以结构体内最长的数据元素的长度为对齐单位，即为其整数倍。若结构体内元素长度大于处理器位数则以处理器位数为单位对齐。

（6） unsigned影响的只是最高位的意义，数据长度不会改变，所以sizeof（unsigned int）=4

（7） 自定义类型的sizeof取值等于它的类型原型取sizeof

（8） 对函数使用sizeof，在编译阶段会被函数的返回值的类型代替

（9） sizeof后如果是类型名则必须加括号，如果是变量名可以不加括号，这是因为sizeof是运算符

（10） 当使用结构类型或者变量时，sizeof返回实际的大小。当使用静态数组时返回数组的全部大小，sizeof不能返回动态数组或者外部数组的尺寸

8．sizeof与strlen的区别?

答： （1）sizeof的返回值类型为size_t（unsigned int）；

（2）sizeof是运算符，而strlen是函数；

（3）sizeof可以用类型做参数，其参数可以是任意类型的或者是变量、函数，而strlen只能用char*做参数，且必须是以'\0'结尾；

（4）数组作sizeof的参数时不会退化为指针，而传递给strlen是就退化为指针；

（5）sizeo是编译时的常量，而strlen要到运行时才会计算出来，且是字符串中字符的个数而不是内存大小；

9．指针和引用的区别?

答：指针和引用都提供了间接操作对象的功能。

（1） 指针定义时可以不初始化，而引用在定义时就要初始化，和一个对象绑定，而且一经绑定，只要引用存在，就会一直保持和该对象的绑定；

（2） 赋值行为的差异：指针赋值是将指针重新指向另外一个对象，而引用赋值则是修改对象本身；

10．数组和指针的区别?

答：（1）数组要么在全局数据区被创建，要么在栈上被创建；指针可以随时指向任意类型的内存块；

（2）修改内容上的差别：

char a[] = “hello”;

a[0] = ‘X’;

char *p = “world”; // 注意p 指向常量字符串

p[0] = ‘X’; // 编译器不能发现该错误，运行时错误

(3)用运算符sizeof 可以计算出数组的容量（字节数）。sizeof(p),p 为指针得到的是一个指针变量的字节数，而不是p 所指的内存容量。C++/C 语言没有办法知道指针所指的内存容量，除非在申请内存时记住它。注意当数组作为函数的参数进行传递时，该数组自动退化为同类型的指针。

11.空指针和悬垂指针的区别?

答：空指针是指被赋值为NULL的指针；delete指向动态分配对象的指针将会产生悬垂指针。

（1） 空指针可以被多次delete，而悬垂指针再次删除时程序会变得非常不稳定；

（2） 使用空指针和悬垂指针都是非法的，而且有可能造成程序崩溃，如果指针是空指针，尽管同样是崩溃，但和悬垂指针相比是一种可预料的崩溃。

12.C++中有malloc/free，为什么还有new/delete?

答：malloc/free是C/C++标准库函数，new/delete是C++运算符。他们都可以用于动态申请和释放内存。

对于内置类型数据而言，二者没有多大区别。malloc申请内存的时候要制定分配内存的字节数，而且不会做初始化；new申请的时候有默认的初始化，同时可以指定初始化；

对于类类型的对象而言，用malloc/free无法满足要求的。对象在创建的时候要自动执行构造函数，消亡之前要调用析构函数。由于malloc/free是库函数而不是运算符，不在编译器控制之内，不能把执行构造函数和析构函数的任务强加给它，因此，C++还需要new/delete。

答：当类中有指针成员时，一般有两种方式来管理指针成员：一是采用值型的方式管理，每个类对象都保留一份指针指向的对象的拷贝；另一种更优雅的方式是使用智能指针，从而实现指针指向的对象的共享。

智能指针的一种通用实现技术是使用引用计数。智能指针类将一个计数器与类指向的对象相关联，引用计数跟踪该类有多少个对象共享同一指针。

每次创建类的新对象时，初始化指针并将引用计数置为1；当对象作为另一对象的副本而创建时，拷贝构造函数拷贝指针并增加与之相应的引用计数；对一个对象进行赋值时，赋值操作符减少左操作数所指对象的引用计数（如果引用计数为减至0，则删除对象），并增加右操作数所指对象的引用计数；调用析构函数时，构造函数减少引用计数（如果引用计数减至0，则删除基础对象）。

14.面向对象技术的基本概念是什么，三个基本特征是什么?

答：基本概念：类、对象、继承； 基本特征：封装、继承、多态。

封装：将低层次的元素组合起来形成新的、更高实体的技术；

继承：广义的继承有三种实现形式：实现继承、可视继承、接口继承。

多态：允许将子类类型的指针赋值给父类类型的指针

15.C++空类默认有哪些成员函数?

答：默认构造函数、析构函数、复制构造函数、赋值函数

16.哪一种成员变量可以在一个类的实例之间共享?

答：static静态成员变量

17.继承层次中，为什么基类析构函数是虚函数?

答：编译器总是根据类型来调用类成员函数。但是一个派生类的指针可以安全地转化为一个基类的指针。这样删除一个基类的指针的时候，C++不管这个指针指向一个基类对象还是一个派生类的对象，调用的都是基类的析构函数而不是派生类的。如果你依赖于派生类的析构函数的代码来释放资源，而没有重载析构函数，那么会有资源泄漏。

18.为什么构造函数不能为虚函数?

19.如果虚函数是有效的，那为什么不把所有函数设为虚函数?

答：不行。首先，虚函数是有代价的，由于每个虚函数的对象都要维护一个虚函数表，因此在使用虚函数的时候都会产生一定的系统开销，这是没有必要的。

20.构造函数可以是内联函数

21.什么是多态？多态有什么作用?

答：多态就是将基类类型的指针或者引用指向派生类型的对象。多态通过虚函数机制实现。

多态的作用是接口重用。

22.重载和覆盖有什么区别?

答：虚函数是基类希望派生类重新定义的函数，派生类重新定义基类虚函数的做法叫做覆盖；

重载就在允许在相同作用域中存在多个同名的函数，这些函数的参数表不同。重载的概念不属于面向对象编程，编译器根据函数不同的形参表对同名函数的名称做修饰，然后这些同名函数就成了不同的函数。

重载的确定是在编译时确定，是静态的；虚函数则是在运行时动态确定。

23.公有继承、受保护继承、私有继承

答：（1）公有继承时，派生类对象可以访问基类中的公有成员，派生类的成员函数可以访问基类中的公有和受保护成员；

（2）私有继承时，基类的成员只能被直接派生类的成员访问，无法再往下继承；

（3）受保护继承时，基类的成员也只被直接派生类的成员访问，无法再往下继承。

24.公有继承时基类受保护的成员，可以通过派生类对象访问但不能修改。

25.有哪几种情况只能用构造函数初始化列表而不能用赋值初始化?

答：const成员，引用成员

26.什么是虚指针?

答：虚指针或虚函数指针是虚函数的实现细节。带有虚函数的每一个对象都有一个虚指针指向该类的虚函数表。

27.C++如何阻止一个类被实例化？一般在什么时候将构造函数声明为private?

答：（1）将类定义为抽象基类或者将构造函数声明为private；

（2）不允许类外部创建类对象，只能在类内部创建对象

28.main函数执行之前会执行什么？执行之后还能执行代码吗?

答：（1）全局对象的构造函数会在main函数之前执行；

（2）可以，可以用_onexit 注册一个函数，它会在main 之后执行;

如果你需要加入一段在main退出后执行的代码，可以使用atexit()函数，注册一个函数。

语法：

```
#include <stdlib.h>

#include <stdio.h>

int atexit(void (*function")(void));

void fn1( void ), fn2( void ), fn3( void );

int main( void )

{

atexit(fn1);

atexit( fn2 );

printf( "This is executed first.\n" );

}
```

```
{
printf( " This is\n" );
}
void fn2()
{
printf( " executed next." );
}
```

结果：

This is executed first.

This is executed next.

29.请描述进程和线程的区别？

答：（1）进程是程序的一次执行，线程是进程中的执行单元；

（2）进程间是独立的，这表现在内存空间、上下文环境上，线程运行在进程中；

（3）一般来讲，进程无法突破进程边界存取其他进程内的存储空间；而同一进程所产生的线程共享内存空间；

（4）同一进程中的两段代码不能同时执行，除非引入多线程。

30.进程间如何通信？

答：信号、信号量、消息队列、共享内存

31.在网络编程中涉及并发服务器，使用多进程与多线程的区别？

答：（1）线程执行开销小，但不利于资源管理和保护；进程则相反，进程可跨越机器迁移。

（3）线程产生的速度快，线程间通信快、切换快；

（4）线程的资源利用率比较好；

（5）线程使用公共变量或者资源时需要同步机制。

32.说一下TCP 3次握手、4次挥手的全过程。

33.TCP和UDP有什么区别。

答：

TCP——传输控制协议,提供的是面向连接、可靠的字节流服务。

当客户和服务器彼此交换数据前，必须先在双方之间建立一个TCP连接，之后才能传输数据。TCP提供超时重发，丢弃重复数据，检验数据，流量控制等功能，保证数据能从一端传到另一端。

UDP——用户数据报协议，是一个简单的面向数据报的传输层协议。UDP不提供可靠性，它只是把应用程序传给IP层的数据报发送出去，但是并不能保证它们能到达目的地。由于UDP在传输数据报前不用在客户和服务器之间建立一个连接，且没有超时重发等机制，故而传输速度很快.

TCP协议和UDP协议的一些特性区别如下：

1.TCP协议在传送数据段的时候要给段标号；UDP 协议不需要。

2.TCP协议可靠；UDP协议不可靠。

3.TCP协议是面向连接；UDP协议采用无连接。

4.TCP协议负载较高,采用虚电路；UDP协议低负载。

5.TCP协议的发送方要确认接受方是否收到数据段(3次握手协议)。

6.TCP协议采用窗口技术和流控制。

34.如何编写套接字?

35.调用函数时要进行参数压栈，一般情况下顺序是从最右边参数往左压栈。

答：（1）栈区：由编译器自动分配和释放，存放函数的参数值、局部变量的值等；

（2）堆：一般由程序员分配和释放，存放动态分配的变量；

（3）全局区（静态区）：全局变量和静态变量存放在这一块，初始化的和未初始化的分开放；

（4）文字常量区：常量字符串就放在这里，程序结束自动释放；

（5）程序代码区：参访函数体的二进制代码。

37.请讲述堆和栈的区别。

答：（1）申请方式不同。栈上有系统自动分配和释放；堆上有程序员自己申请并指明大小；

（2）栈是向低地址扩展的数据结构，大小很有限；堆是向高地址扩展，是不连续的内存区域，空间相对大且灵活；

（3）栈由系统分配和释放速度快；堆由程序员控制，一般较慢，且容易产生碎片；

38.全局变量放在数据段，内部变量static int count；放在数据段，内部变量char *p="AAA"，p的位置在堆栈上，指向的空间的位置数据段，内部变量char *p=new char；p的位置堆，指向的空间的位置数据段

39.字符数组与字符串的比较：最明显的区别是字符串会在末尾自动添加空字符。

40.函数指针相关概念（C++学习笔记）

41.类使用static成员的优点，如何访问?

答：优点：

（1）static 成员的名字是在类的作用域中，因此可以避免与其他类的成员或全局对象名字冲突；

（2）可以实施封装。static 成员可以是私有成员，而全局对象不可以；

（3） static 成员是与特定类关联的，可清晰地显示程序员的意图。

static 数据成员必须在类定义体的外部定义(正好一次)，static 关键字只能用于类定义体内部的声明中义不能标示为static. 不像普通数据成员，static成员不是通过类构造函数进行初始化，也不能在类的声明中

静态数据成员初始化的格式为：

＜数据类型＞ ＜类名＞::＜静态数据成员名＞=＜值＞

类的静态数据成员有两种访问形式：

＜类对象名＞.＜静态数据成员名＞ 或 ＜类类型名＞::＜静态数据成员名＞

42. static数据成员和static成员函数

答：（1）static数据成员：

static数据成员独立于该类的任意对象而存在；每个static数据成员是与类关联的对象，并不与该类的对象相关联。Static数据成员（const

static数据成员除外）必须在类定义体的外部定义。不像普通数据成员，static成员不是通过类的构造函数进行初始化，而是应该在定义时进行初始化。

（2）static成员函数：

Static成员函数没有this形参，它可以直接访问所属类的static成员，不能直接使用非static成员。因为static成员不是任何对象的组成部分，所以static成员不能被声明为const。同时，static成员函数也不能被声明为虚函数。

43.static成员变量定义放在cpp文件中，不能放在初始化列表中。Const static成员可就地初始化。

44.如何引用一个已经定义过的全局变量?

答：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量，假定你将那个变量写错了，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。

44.static关键字的作用。

答：static总是使得变量或对象的存储形式变成静态存储，连接方式变成内部连接，对于局部变量（已经是内部连接了），它仅改变其存储方式；对于全局变量（已经是静态存储了），它仅改变其连接类型

45.奈奎斯特定理

47．多态类中的虚函数表是 Compile-Time，还是 Run-Time时建立的?

答案：虚拟函数表是在编译期就建立了,各个虚拟函数这时被组织成了一个虚拟函数的入口地址的数组。而对象的隐藏成员--虚拟函数表指针是在运行期--也就是构造函数被调用时进行初始化的，这是实现多态的关键。

48. 一个父类写了一个 virtual 函数，如果子类覆盖它的函数不加 virtual ,也能实现多态?

在子类的空间里，有没有父类的这个函数，或者父类的私有变量? (华为笔试题)

答案：只要基类在定义成员函数时已经声明了 virtue关键字，在派生类实现的时候覆盖该函数时，virtue关键字可加可不加，不影响多态的实现。子类的空间里有父类的所有变量(static除外)。

49. 完成字符串拷贝可以使用 sprintf、strcpy 及 memcpy 函数，请问这些函数有什么区别

，你喜欢使用哪个，为什么?

答案：这些函数的区别在于 实现功能以及操作对象不同。

（1）strcpy 函数操作的对象是字符串，完成从源字符串到目的字符串的拷贝功能。

（2）sprintf 函数操作的对象不限于字符串：虽然目的对象是字符串，但是源对象可以是字符串、也可以是任意基本类型的数据。这个函数主要用来实现（字符串或基本数据类型）向字符串的转换功能。如果源对象是字符串，并且指定 %s 格式符，也可实现字符串拷贝功能。

（3）memcpy 函数顾名思义就是内存拷贝，实现将一个内存块的内容复制到另一个内存块这一功能。内存块由其首地址以及长度确定。程序中出现的实体对象，不论是什么类型，其最终表现就是在内存中占据一席之地（一个内存区间或块）。因此，memcpy

的操作对象不局限于某一类数据类型，或者说可适用于任意数据类型，只要能给出对象的起始地址和内存长度信息、并且对象具有可操作性即可。鉴于memcpy 函数等长拷贝的特点以及数据类型代表的物理意义，memcpy 函数通常限于同种类型数据或对象之间的拷贝，其中当然也包括字符串拷贝以及基本数据类型的拷贝。

对于字符串拷贝来说，用上述三个函数都可以实现，但是其实现的效率和使用的方便程度不同：

• strcpy 无疑是最合适的选择：效率高且调用方便。

• memcpy 虽然高效，但是需要额外提供拷贝的内存长度这一参数，易错且使用不便；并且如果长度指定过大的话（最优长度是源字符串长度 + 1），还会带来性能的下降。其实 strcpy 函数一般是在内部调用 memcpy 函数或者用汇编直接实现的，以达到高效的目的。因此，使用 memcpy 和 strcpy 拷贝字符串在性能上应该没有什么大的差别。

对于非字符串类型的数据的复制来说，strcpy 和 snprintf 一般就无能为力了，可是对 memcpy 却没有什么影响。但是，对于基本数据类型来说，尽管可以用 memcpy 进行拷贝，由于有赋值运算符可以方便且高效地进行同种或兼容类型的数据之间的拷贝，所以这种情况下 memcpy 几乎不被使用 。memcpy 的长处是用来实现（通常是内部实现居多）对结构或者数组的拷贝，其目的是或者高效，或者使用方便，甚或两者兼有。

50. 应用程序在运行时的内存包括代码区和数据区，其中数据区又包括哪些部分？

答：对于一个进程的内存空间而言，可以在逻辑上分成 3个部份：代码区，静态数据区和动态数据区。

动态数据区一般就是“堆栈”。 栈是一种线性结构，堆是一种链式结构。进程的每个线程都有私有的“栈”。

全局变量和静态变量分配在静态数据区，本地变量分配在动态数据区，即堆栈中。程序通过堆栈的基地址和偏移量来访问本地变量。

51. C++函数中值的传递方式有哪几种?

答：三种传递方式为：值传递、指针传递和引用传递。

52. C++里面是不是所有的动作都是main()引起的？如果不是，请举例.

比如全局变量的初始化，就不是由main函数引起的

举例： class A{};

A a; //a的构造函数限执行

int main() {}

53. 下列哪两个是等同的

int b;

B const* int a = &b;

C const int* const a = &b;

D int const* const a = &b;

54. 内联函数在编译时是否做参数类型检查?

答：内联函数要做参数类型检查, 这是内联函数跟宏相比的优势。

55. 全局变量和局部变量有什么区别？实怎么实现的？操作系统和编译器是怎么知道的?

（1）生命周期不同：

全局变量随主程序创建和创建，随主程序销毁而销毁

局部变量在局部函数内部，甚至局部循环体等内部存在，退出就不存在； 内存中

分配在全局数据区

（2）使用方式不同：通过声明后全局变量程序的各个部分都可以用到；局部变量只能在局部使用，分配在栈区

操作系统和编译器通过内存分配的位置来知道的，全局变量分配在全局数据段并且在程序开始运行的时候被加载。局部变量则分配在堆栈里面 。

56. 有 A 、 B 、 C 、 D 四个人，要在夜里过一座桥。他们通过这座桥分别需要耗时 1 、 2 、 5 、 10 分钟，只有一支手电，并且同时最多只能两个人一起过桥。请问，如何安排，能够在 17 分钟内这四个人都过桥?

Solution:关键是时间最长的两个人必须同时过桥

The First Time： A(1) 和 B(2) 过桥， A(1) 返回 Cost： 1+2

The Second Time： C(5) 和 D(10) 过桥， B(2) 返回 Cost： 10+2

The Third Time A(1) 和 B(2) 过桥 Cost： 2

57. static全局变量与普通的全局变量有什么区别？static局部变量和普通局部变量有什么区别？static函数与普通函数有什么区别？

答：static全局变量与普通全局变量区别：static全局变量只初使化一次，防止在其他文件单元中被引用;

static局部变量和普通局部变量区别：static局部变量只被初始化一次，下一次依据上一次结果值；

static函数与普通函数区别：static函数在内存中只有一份，普通函数在每个被调用中维持一份拷贝。

58. 程序的局部变量存在于（堆栈）中，全局变量存在于（静态区 ）中，动态申请数据存在于（ 堆）中。

59. 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?

c用宏定义，c++用inline

60. 有1,2,....一直到n的无序数组,求排序算法,并且要求时间复杂度为O(n),空间复杂度O(1),使用交换,而且一次只能交换两个数。

复制代码

```
#include<iostream.h>

Using namespace std;

int main(){

int a[] = {10,6,9,5,2,8,4,7,1,3};

int len = sizeof(a) / sizeof(int);

int temp;

for(int i = 0; i < len; )

{

temp = a[a[i] - 1];

a[a[i] - 1] = a[i];

a[i] = temp;
```

```
i++;

}

for (int j = 0; j < len; j++)

cout<<a[j]<<",";

return 0;

}
```

文章分类 代码人生 文章标签 C++

linux服务器开... Lv2
获得点赞 3 · 获得阅读 10,297

相关推荐

吴尼玛 · 23小时前 · C++
Linux命令行参数解析——getopt_long

Lucy_FinTech社区 · 1天前 · C++
手把手教你用C++搭建操作系统

首页 探索掘金

青玉白露 · 3天前 · C++

C/C++杂谈(1)指针常量、常量指针

3

doodlewind · 1月前 · JavaScript / C++

写给前端的手动内存管理基础入门（一）返璞归真：从引用类型到裸指针

137 16

彭丑丑 · 13天前 · C++

NDK | C++ 复习笔记

7 2

God23Bin · 2天前 · C++

C语言-字符串函数的实现（一）之strlen

vivo互联网技术 · 1月前 · Android / C++

Android NativeCrash 捕获与解析

42 5

彭丑丑 · 1月前 · C++

NDK | C 语言复习笔记

16 4

carrot__lsp · 9天前 · C++

C++ Day14-其他语法 C 智能指针

1 1

ruki · 10天前 · C++

xmake v2.5.3 发布，支持构建 linux bpf 程序和 Conda 包集成

1 1