目录

[C和C++的区别 1](#_Toc536054883)

[快排 2](#_Toc536054884)

[写一个strcpy 2](#_Toc536054885)

[new和malloc的区别 2](#_Toc536054886)

[指针和引用的区别 2](#_Toc536054887)

[什么是虚函数及用法 4](#_Toc536054888)

[隐藏覆盖重载 4](#_Toc536054889)

[静态库和动态库的区别 4](#_Toc536054890)

[查看共享库的命令 4](#_Toc536054891)

[进程间通信的方式 4](#_Toc536054892)

[僵死进程 4](#_Toc536054893)

[同步线程的方法 4](#_Toc536054894)

[TCP和UDP的区别 5](#_Toc536054895)

[select、epoll、poll的区别 5](#_Toc536054896)

[Et Lt的区别 5](#_Toc536054897)

[const define static 的用法 5](#_Toc536054898)

[linux基本命令 5](#_Toc536054899)

[有名管道无名管道 5](#_Toc536054900)

## C和C++的区别

C语言是面向过程的语言，就是分析出问题的步骤，将这些步骤一一实现，使用时依次调用。

优点：性能比面向对象高，因为类调用时需要实例化，开销比较大，比较消耗资源;比如[单片机](https://www.baidu.com/s?wd=%E5%8D%95%E7%89%87%E6%9C%BA&tn=24004469_oem_dg&rsv_dl=gh_pl_sl_csd)、嵌入式开发、 Linux/Unix等一般采用面向过程开发，性能是最重要的因素。   
缺点：没有面向对象易维护、易复用、易扩展。

C++是面向对象的语言，将问题分解成各个对象，建立对象的目的不是为了完成某一个步骤，而是为了描述某一事物在解决问题的步骤中的行为。

优点：易维护、易复用、易扩展，由于面向对象有封装、继承、多态性的特性，可以设计出低耦合的系统，使系统 更加灵活、更加易于维护   
缺点：性能比面向过程低

一些关键字的细微区别：

（1）struct：在C语言中struct定义的变量中不能有函数，而在C++中可以有函数。

（2）malloc ：malloc函数的返回值为void\*，在C语言中可以赋值给任意类型的指针，在C++中必须强制类型转换，否则报错。

（3）struct和class：class是对struct的扩展，struct默认的访问权限是public，而class默认的访问权限是private。

返回值

C语言如果没有写返回值，那么默认返回值为int

而C++必须指定返回值，若没有返回值则必须加上void

参数列表

C语言中如果没有指定参数列表，那么编译器会默认可以接受多个参数

C++中如果没有指定参数列表，那么就不会接受任何参数

函数默认值

在定义函数时，C++可以在形参中设计一个默认值，以便调用函数形参缺省时使用。但C语言不支持。

内联函数inline

C++可以使用inline来修饰函数，与宏类似，但又有区别，inline在编译期间处理，有类型和安全的检查。在函数调用点展开，没有开栈淸栈的开销，速度快。这也是C语言不具备的。

函数重载

C语言中产生函数符号的规则是根据名称产生，这也就注定了C语言不存在函数重载的概念。而C++生成符号则考虑了函数名、参数个数、参数类型。属于静多态

const

C语言中的const：被修饰后不能做左值，可以不初始化，但是之后没有机会再初始化。不可以当数组的下标，可以通过指针修改。简单来说，它和普通变量的区别只是不能做左值而已。其他地方都是一样的。

C++中的const：真正的常量。定义的时候必须初始化，可以用作数组的下标。const在C++中的编译规则是替换（和宏很像），所以它被看作是真正的常量。

作用域

C语言只有全局作用域、局部作用域

C++有类作用域、局部作用域、名字空间作用域

## 快排

## 写一个strcpy

## new和malloc的区别

属性

new/delete是C++关键字，需要编译器支持。malloc/free是库函数，需要头文件支持。

参数

使用new操作符申请内存分配时无须指定内存块的大小，编译器会根据类型信息自行计算。而malloc则需要显式地指出所需内存的尺寸。

返回类型

new操作符内存分配成功时，返回的是对象类型的指针，类型严格与对象匹配，无须进行类型转换，故new是符合类型安全性的操作符。而malloc内存分配成功则是返回void \* ，需要通过强制类型转换将void\*指针转换成我们需要的类型。

分配失败

new内存分配失败时，会抛出bac\_alloc异常。malloc分配内存失败时返回NULL。

自定义类型

        new会先调用operator new函数，申请足够的内存（通常底层使用malloc实现）。然后调用类型的构造函数，初始化成员变量，最后返回自定义类型指针。delete先调用析构函数，然后调用operator delete函数释放内存（通常底层使用free实现）。

         malloc/free是库函数，只能动态的申请和释放内存，无法强制要求其做自定义类型对象构造和析构工作。

重载

C++允许重载new/delete操作符，特别的，布局new的就不需要为对象分配内存，而是指定了一个地址作为内存起始区域，new在这段内存上为对象调用构造函数完成初始化工作，并返回此地址。而malloc不允许重载。

内存区域

new操作符从自由存储区（free store）上为对象动态分配内存空间，而malloc函数从堆上动态分配内存。自由存储区是C++基于new操作符的一个抽象概念，凡是通过new操作符进行内存申请，该内存即为自由存储区。而堆是操作系统中的术语，是操作系统所维护的一块特殊内存，用于程序的内存动态分配，C语言使用malloc从堆上分配内存，使用free释放已分配的对应内存。自由存储区不等于堆，如上所述，布局new就可以不位于堆中。

## 指针和引用的区别

(1)当引用被创建时，它必须被初始化。而指针则可以在任何时候被初始化。

(2)一旦一个引用被初始化为指向一个对象，它就不能被改变为对另一个对象的引用。而指针则可以在任何时候指向另一个对象。

(3)不可能有NULL引用。必须确保引用是和一块合法的存储单元关联。

进一步解析：

指针和引用都是 C++ 中的基本语法成份，它们既有联系，也有不同之处。

它们都是地址的概念，其中指针指向一块内存，它的内容是所指内存的地址；而引用是某块内存的别名，具体来说，指针是一个变量的地址，引用是一个变量的别名。

但它们的不同之处也很明显，体现在以下方面：

* 指针是一个实体，而引用仅是个别名；
* 引用必须被初始化，指针不必；

* 引用只能在定义时被初始化一次，之后不可变；指针可以改变所指的对象；

* 可以有const指针，但是没有const引用；

* 不存在指向空值的引用，但是存在指向空值的指针，即引用不能为空，指针可以为空；

* “sizeof 引用”得到的是所指向的变量（对象）的大小，而“sizeof 指针”得到的是指针本身（所指向的变量或对象的地址）的大小；

* 指针和引用的自增（++）运算意义不一样；

* 程序为指针变量分配内存区域，而引用不需要分配内存区域；

* 指针可以有多级，但是引用只能是一级，例如int \*\*p是合法的，而 int &&a是不合法的；

* 指针和引用作为函数参数进行传递时也不同。用指针传递参数，可以实现对实参进行改变的目的；在将引用作为函数参数进行传递时，实质上传递的是实参本身，而不是实参的一个拷贝，因此对形参的修改其实是对实参的修改。

## 什么是虚函数及用法

## 隐藏覆盖重载

隐藏：继承关系下，基类和派生类同名的方法。派生类下隐藏了基类中所有同名的函数

覆盖：继承关系下，基类中和派生类中同名同参的方法。派生类中同名同参的虚函数会覆盖基类中同名同参的虚函数

## 静态库和动态库的区别

## 查看共享库的命令

## 进程间通信的方式

管道 共享内存 信号量 消息队列 套接字

## 僵死进程

僵死进程：子进程先于父进程结束，父进程没有调用wait获取子进程的退户码

处理僵死进程：1.使用wait 2.使用signal（SIGCHLD，SIG\_IGN）(只能在liunx下使用)

## 同步线程的方法

信号量 互斥锁 条件变量 读写锁

## TCP和UDP的区别

## select、epoll、poll的区别

## Et Lt的区别

## const define static 的用法

## linux基本命令

## 有名管道无名管道

Pipe无名管道（只能在父子进程间使用）

Mkfifo 有名管道

有名管道的属性信息是存在磁盘中的，如属主，创建时间，权限等，名字等