

Lesson 3

1.异常处理的顺序：把一段代码放入 `try` 中，如果一切运行正常，那么代码不会进入 `catch` 语句。如果代码运行期间发生异常，那么代码直接跳转至 `catch` 中，`try` 剩余的代码不会被执行。

2.一旦程序发生异常，如果不进行处理，会马上退出(`core dump`)。

3.`catch` 只能捕获特定类型的异常，如果一个异常没有 `catch` 块可以捕捉，那么等价于直接 `throw`，程序退出。

4.一个异常对象最多只能捕获一次。

5.异常捕获的顺序是从上到下，所以最上面的 `catch` 块优先级最高。因此...通用的 `catch` 块必须放在所有 `catch` 的最后面。

6.`pass-by-value` 的特点：

a) 优点是生成的副本不会影响原来的值

b) 缺点是参数进行 `value` 拷贝，带来一定的开销，尤其是对于 C++ 的 `string` 等类型。

7.采用引用交换两个变量的通用写法：

```
typedef int T;
void swap(T &a, T &b){
    T temp = a;
    a = b;
    b = temp;
}
```

8.对于参数列表，应该尽可能的使用 `const` 关键字。

- a) 如果参数中含有 `const`，那么这个参数只读。
- b) 如果参数中是非 `const` 引用，那么参数可以改变，而且我们期望它做出改变。

9.`const` 引用参数提供了两重作用：

- a) 避免了参数拷贝的开销
- b) 保护了参数，使它变为只读。

10.对于原生类型 `int double` 等，如果只读，那么采用简单的传值即可，如果需要改变，可以使用引用，但是最佳做法是传入它的指针。

11.宏函数和内联函数的区别：

- a) 宏函数是在预编译期间进行替换。
- b) 内联函数是在编译期间进行代码扩展。
- c) 内联函数可以看做高级的宏函数，它进行语法检查。
- d) 宏函数只要不进行调用，就不会产生语法检查，而内联函数必须首先经过严格的语法检查才能进行代码扩展。

12.普通函数不能放在头文件中，但是 `inline` 函数必须放在头文件中。

13.内联函数的优点：避免了函数调用的开销，缺点是造成代码膨胀。

14.在 `gcc` 中，`assert` 不加头文件，造成的是链接错误，原因

在于：assert 是个宏函数，不加头文件时，gcc 把 assert 当作一个函数，并且为它推断原型为: `int assert(int);`然后在链接的时候寻找这个函数，所以造成链接错误。

15.在 C 语言中不加头文件，也可以进行正常的函数调用。

16.函数返回值不为 void 的时候，每条路径都必须都返回值。

17.返回值为 value 时，会生成一个临时的 ret 值来存储中间结果。string s = get_str("hello")会先将“hello”的值赋给一个中间临时的 string 对象，然后此对象赋值给 s.

18.结论：返回 const 引用形参的引用时，必须加上 const。原因在于：const 提供了一种语义约束：常量性。const 意味着常量，也就是意味着不允许被修改。而非 const 引用参数是一种期望修改的语义，二者矛盾。

19.绝对不要试图返回局部对象（变量）的指针或者引用。

20.C++中函数的唯一标示（签名， signature）由函数名和形参表组成（这里不完整）。

21.C 不支持函数重载的原因是：C 使用函数名作为 signature。

22.C++程序在生成符号表的时候（symbol table）的时候进行了名字改编（name mangling）。

23.在 C++标准库算法中需要提供的区间全部为[)左闭右开区间。

24.sort 算法在未指定排序规则的情况下（也就是两个参数的版本），默认调用元素的<操作符。

25.C 中用 ASCII 和二进制打开文件的区别是：二进制仅仅把读取的内容当做字节对待，而 ASCII 方式会识别\r\n 等符号。

26.windows 系统中换行符为\r\n，Linux 中为\n。

27.FTP 服务器有两种传输方式：一种是 ASCII 方式，这意味着从 win 到 Linux 系统相互传输文件时，会自己完成\r\n 和\n 的转换。第二种是二进制方式（例如文盲不识别文字），这种仅仅传输字节。

28.标准 C 的 IO 库是用 Linux 系统调用编写的。而在 win 下，C 库是用 win32 API 实现的。

29.标准 C 的 IO 自己实现了一个用户级别的缓冲区。Linux 的 read 函数没有实现缓冲区，如果我们调用 1000 次 read 函数，每次读取一个字符，那么将发生 1000 次系统调用。而我们调用 getchar1000 次，发生系统调用的次数将远小于 1000 次。

30.在 TCP 连接中，如果调用 write(fd, buf, 1000)那么首先发生的是先将用户空间的 buf，拷贝至内核空间。

31.用户态下不能读取内核空间，内核态下也不能读取用户空间数据。