C++符号修饰

C++拥有类、继承、虚机制、重载、名称空间等这些特性，尽管函数名相同，但是参数列表不同，这是C++重载的最简单方式；

为了支持C++这些复杂的特性，人们发明了符号修饰（Name Decoration）或符号改编的机制。

首先看下面的这段代码：

int func(int);

float func(float);

class C {

int func(int);

class C2 {

int func(int);

}

}

namespace N {

int func(int);

class C {

int func(int);

}

}

这段代码有6个同名函数func，只不过它们的返回类型和参数所在的名称空间不同。我们引入一个术语

叫做函数签名，函数签名包含了一个函数的信息，包括函数名、它的参数类型、它所在的类和名称空间及

其他信息。函数签名用于识别不同的函数，就像签名用于识别不同的人一样，函数的名字只是函数签名的一部分。



所有符号都以“\_Z”开头，对于嵌套的名字（在名称空间或在类里面的），后面紧跟“N”，然后是各个名称空间和类的名字，每个名字前是名字字符串长度，再以“E”结尾。比如名称修饰以后就是\_ZN1N1C4funcE。对于一个函数来说，它的参数列表紧跟在“E”后面，对于int类型就是字母“i”。所以整个N::C::func(int)的函数签名就是\_ZN1N1C4funcEi。

签名和名称修饰机制不光被使用在函数中，C++的全局变量和静态变量都有同样的机制。

不同编译器产商的名称修饰方式可能不同。