# STL-string

## 1.1.string源码解析

### 1.1.1 string定义

C++标准库中关于string的定义在stringfwd.h头文件中，linux操作系统下路径(以下所有代码路径都是linux操作系统下)为：/usr/include/c++/4.8.2/bits，具体定义如下截图：

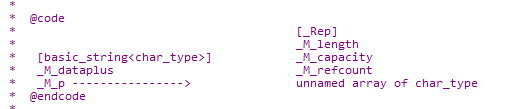


由如上截图中的内容可知，string的定义其实是basic\_string模板类对char的一个模板实例化。

basic\_string模板类定义在basic\_string文件中，具体路径为(/usr/include/c++/4.8.2/bits)。关于basic\_string模板类的具体实现放在了basic\_string.tcc当中，c++里面模板的实现不能放在.cpp文件中，必须写在头文件中，如果模板函数实现较复杂，就会导致头文件臃肿和杂乱，这里可以看到stl里面方法，就是把较复杂的实现放在.tcc文件里面，然后当做头文件来包含，我们在写模板代码的时候也可以以此为参考。

### 1.1.2 basic\_string成员定义

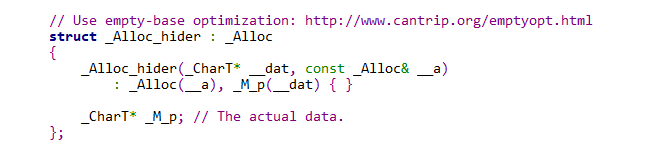
首先，在basic\_string.h顶头注释中我们看到关于几个主要的类成员定义，如下截图：



这里其实是介绍了basic\_string的内存布局，从起始地址出开始，\_M\_length表示字符串的长度、\_M\_capacity是最大容量、\_M\_refcount是引用计数，\_M\_p指向实际的数据。值得注意的是引用计数，说明该版本的string实现采用了copy-on-write的方式来减少无意义的内存拷贝。

由上图中的定义我们进行计算std::string的大小应该为24字节（没有数据）或者32字节（有数据）。其实不然，C++对象的大小是根据非静态成员的大小极其个数进行计算的，静态成员变量和成员函数是不计算在内的，因此根据basic\_string.h中的定义可以看出，非静态成员变量只有一个，如下截图：

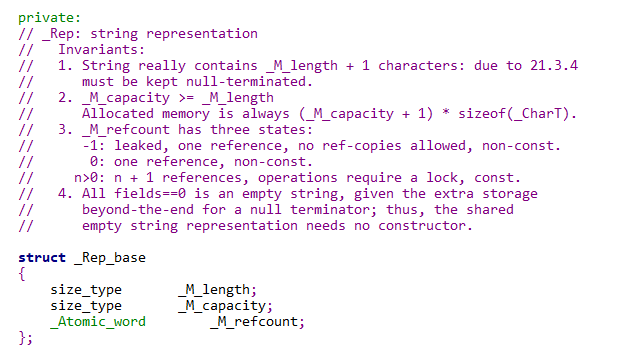
关于\_Alloc\_hider的定义(/usr/include/c++/4.8.2/bits/basic\_string.h)如下图：



从上图我们可以看出，\_Alloc\_hider是个结构体类型，其中存放的就是前面提到的\_M\_p变量，即模板类实际指向的数据体。\_Alloc\_hider的基类\_Alloc是一个分配器，无成员变量，大小为0。则sizeof(std::string)的大小为8（64位系统）/4（32位系统）。

如果计算std::string所花费的总空间字节数应该为sizeof(std::string) + sizeof(‘ ‘) + sizeof(\_Rep) = 33个字节。至于sizeof(‘ ‘)和sizeof(\_Rep)后面会提到，即必须以null结尾多出来的一个字节大小和其他3个变量的所占字节大小，其中sizeof(std::string)的大小为栈空间大小，这一点要区分开来。

由上我们已经解析了basic\_string的四个主要成员中的一个，至于另外三个成员的定义如下(\_M\_length、\_M\_capacity、\_M\_refcount)：



从上图我们可以看出，这三个成员通过一个结构体来进行管理，而不是直接作为类成员进行管理，猪样就只需要定义一个\_Rep指针即可，最大限度的减小了string对象的大小，减小了对象拷贝的消耗。

从上图注释1.String really contains \_M\_length + 1 characters: due to 21.3.4 must be kept null-terminated.中，我们可以看出，string始终由\_M\_length+1个字符，必须保证以null进行结尾。（2.3.4注释后面继续解释）

### 1.1.3 basic\_string构造函数