一．String容器

C++标准库定义了一种string类，定义在头文件<string>中。

1. String 和C风格字符串对比：

1.Char\*是一个指针，String是一个类

string封装了char\*，管理这个字符串，是一个char\*型的容器。

2.String封装了很多实用的成员方法

查找find，拷贝copy，删除delete 替换replace，插入insert

3.不用考虑内存释放和越界

string管理char\*所分配的内存。每一次string的复制，取值都由string类负责维护，不用担心复制越界和取值越界等。

三．string容器的常用操作

1. string的构造函数

string();//创建一个空的字符串 例如: string str;

string(const string& str);//使用一个string对象初始化另一个string对象

string(const char\* s);//使用字符串s初始化

string(int n, char c);//使用n个字符c初始化

1. string的基本赋值操作

string& operator=(const char\* s);//char\*类型字符串 赋值给当前的字符串

string& operator=(const string &s);//把字符串s赋给当前的字符串

string& operator=(char c);//字符赋值给当前的字符串

string& assign(const char \*s);//把字符串s赋给当前的字符串

string& assign(const char \*s, int n);//把字符串s的前n个字符赋给当前的字符串

string& assign(const string &s);//把字符串s赋给当前字符串

string& assign(int n, char c);//用n个字符c赋给当前字符串

string& assign(const string &s, int start, int n);//将s从start开始n个字符赋值给字符串

1. string存取字符操作

当访问越界时，[]会直接挂掉，at会抛出异常。

char& operator[](int n);//通过[]方式取字符

char& at(int n);//通过at方法获取字符

1. string拼接操作

string& operator+=(const string& str);//重载+=操作符

string& operator+=(const char\* str);//重载+=操作符

string& operator+=(const char c);//重载+=操作符

string& append(const char \*s);//把字符串s连接到当前字符串结尾

string& append(const char \*s, int n);//把字符串s的前n个字符连接到当前字符串结尾

string& append(const string &s);//同operator+=()

string& append(const string &s, int pos, int n);//把字符串s中从pos开始的n个字符连接到当前字符串结尾

string& append(int n, char c);//在当前字符串结尾添加n个字符c

1. string查找和替换

int find(const string& str, int pos = 0) const; //查找str第一次出现位置,从pos开始查找

int find(const char\* s, int pos = 0) const; //查找s第一次出现位置,从pos开始查找

int find(const char\* s, int pos, int n) const; //从pos位置查找s的前n个字符第一次位置

int find(const char c, int pos = 0) const; //查找字符c第一次出现位置

int rfind(const string& str, int pos = npos) const;//查找str最后一次位置,从pos开始查找

int rfind(const char\* s, int pos = npos) const;//查找s最后一次出现位置,从pos开始查找

int rfind(const char\* s, int pos, int n) const;//从pos查找s的前n个字符最后一次位置

int rfind(const char c, int pos = 0) const; //查找字符c最后一次出现位置

string& replace(int pos, int n, const string& str); //替换从pos开始n个字符为字符串str

string& replace(int pos, int n, const char\* s); //替换从pos开始的n个字符为字符串s

1. string比较操作

compare函数在>时返回 1，<时返回 -1，==时返回 0。

比较区分大小写，比较时参考字典顺序，排越前面的越小。

大写的A比小写的a小。

\*/

int compare(const string &s) const;//与字符串s比较

int compare(const char \*s) const;//与字符串s比较

1. string子串

string substr(int pos = 0, int n = npos) const;//返回由pos开始的n个字符组成的字符串

1. string插入和删除操作

string& insert(int pos, const char\* s); //插入字符串

string& insert(int pos, const string& str); //插入字符串

string& insert(int pos, int n, char c);//在指定位置插入n个字符c

string& erase(int pos, int n = npos);//删除从Pos开始的n个字符

1. string和c风格字符串的转换

在c++中存在一个从const char\*到string的隐式类型转换，却不存在从一个string对象到C\_string的自动类型转换。对于string类型的字符串，可以通过c\_str()函数返回string对象对应的C\_string.

//string 转 char\*

string str = "itcast";

const char\* cstr = str.c\_str();

//char\* 转 string

char\* s = "itcast";

string str(s);