

补充:

3、TString 类的定义与实现

3.1. 引入

字符串的基本操作都是基于一维字符数组的, 因此其赋值、比较、连接、求长度等方法均要用函数去实现, 且实现过程中必须注意空间是否足够、最后是不是有'\0'等情况。

3.2. 实现目标

参照第 5 章 5.6 中 string 类的方法, 用比较简洁易懂的方法来实现字符串的基本操作。

3.3. 要求实现的基本操作

3.3.1. 定义对象并初始化:

- ① TString s1; //s1 为 NULL
- ② TString s1("hello"); //s1 为"Hello"
- ③ TString s1="hello"; //s1 为"Hello"
- ④ TString s1("Hello"), s2=s1; //s2 也为"Hello"
- ⑤ char *s = "Hello";
 TString s1 = s; //s1 为"Hello"
- ⑥ char s[] = "Hello";
 TString s1 = s; //s1 为"Hello"

3.3.2. 输入操作(输入以空格/回车结束)

- ① TString s1;
 cin >> s1; //若键盘输入 Hello, 则 s1 得到"Hello"
- ② TString s1;
 cin >> s1; //若键盘输入 Hello 123, 则 s1 得到"Hello"(空格为分隔符)

3.3.3. 输出操作:

- ① TString s1("hello");
 cout << s1; //输出"hello"
- ② TString s1;
 cout << s1; //输出"<NULL>"

3.3.4. 赋值操作:

- ① TString s1("hello"), s2;
 s2=s1; //s2 也为"Hello"
- ② TString s1("Hello");
 s1="Hi"; //s1 为"Hi", 原"Hello"不再保留

3.3.5. 连接操作:

- ① TString s1("tong"), s2("ji"), s3;
 s3 = s1+s2; //s3 为"tongji"
 s3 = s2+s1; //s3 为"jitong"
- ② TString s1("tong"), s3;
 s3 = s1+"ji"; //s3 为"tongji"
 s3 = "ji"+s1; //s3 为"jitong"
- ③ TString s1("tong"), s3;
 char *s="ji";
 s3 = s1+s; //s3 为"tongji"
 s3 = s+s1; //s3 为"jitong"
- ④ TString s1("tong"), s3;
 char s[]="ji";
 s3 = s1+s; //s3 为"tongji"
 s3 = s+s1; //s3 为"jitong"

3.3.6. 比较操作:

- ① TString s1="house", s2="horse";
s1 > s2; (包括其它 5 种比较运算) //返回 0/1
- ② TString s1="house";
s1 > "horse"; (包括其它 5 种比较运算) //返回 0/1
"horse" > s2; (包括其它 5 种比较运算) //返回 0/1
- ③ TString s1="house";
char *s="horse"
s1 > s; (包括其它 5 种比较运算) //返回 0/1
s > s2; (包括其它 5 种比较运算) //返回 0/1
- ④ TString s1="house";
char s[]="horse"
s1 > s; (包括其它 5 种比较运算) //返回 0/1
s > s2; (包括其它 5 种比较运算) //返回 0/1

3.3.7. 求串长度:

- ① TString s1("Hello");
cout << s1.length(); //输出为 5
- ② 定义全局函数 TStringLen(TString &); //形参是否需要 const 自行决定
TString s1("Hello"), s2("123");
char *s3="abcde";
char s4[]="wxyz";
TStringLen(s1+s2); //返回值为 8
TStringLen(s2+s1); //返回值为 8
TStringLen(s1+"pq"); //返回值为 7
TStringLen("pq"+s1); //返回值为 7
TStringLen(s1+s3); //返回值为 10
TStringLen(s3+s1); //返回值为 10
TStringLen(s1+s4); //返回值为 9
TStringLen(s4+s1); //返回值为 9

3.3.8. 取串中某个字符的值/给串中的某个字符赋值:

- ① TString s1("hello");
cout << s1[1]; //输出为 5
- ② TString s1("hello");
s1[0] -= 32;
cout << s1; //输出为 Hello

【要求:】1、程序由三个文件组成, 各文件的说明如下:

10-b3.h: 给出 TString 类的定义及其它需要的定义

10-b3.cpp: 给出 TString 类的所有成员函数的实现及其它需要的全局函数的实现

10-b3-main.cpp: 在 main 函数中给出了 TString 类的测试用例, 不准修改, 不需要提交, 检查作业时会替换本文件

2、可部分参考作业 9-b3, 但 9-b3 中未考虑申请空间不够的情况, 此处必须考虑

3、P. 318-322 的例 10.4 中, 仅比较运算符的重载有参考价值, 定义及构造等完全不同

4、在操作系统的内存允许的情况下, **均不再考虑空间是否够用, 但也不能浪费空间**

例 1: TString s1("Hello"); 则最多允许申请 6 个字节的空间, **不能多申请**

例 2: TString s1; 若 s1 = ***** 或 s1 = s1 + ***** 等语句反复出现时, 不能简单限定空间不超过多少字节, **要无尽利用空间, 直到内存空间被耗尽为止**

5、**不允许**使用系统提供的 string 类

6、最后一个 100MB 累加测试完成后, 用“任务管理器”查看的内存占用不能超过 105MB

D:\demo\部分作业(2015级+2016级-VS2015)\Debug\第10章-TString类(2016级).exe

s1已有长度: 99.2213 MB字节, 本次增加 58369 字节
s1已有长度: 99.2277 MB字节, 本次增加 35380 字节
s1已有长度: 99.3107 MB字节, 本次增加 45295 字节
s1已有长度: 99.3539 MB字节, 本次增加 35299 字节
s1已有长度: 99.3876 MB字节, 本次增加 49236 字节
s1已有长度: 99.4345 MB字节, 本次增加 46151 字节
s1已有长度: 99.4785 MB字节, 本次增加 36485 字节
s1已有长度: 99.5133 MB字节, 本次增加 33153 字节
s1已有长度: 99.5449 MB字节, 本次增加 62599 字节
s1已有长度: 99.6046 MB字节, 本次增加 57508 字节
s1已有长度: 99.6595 MB字节, 本次增加 39710 字节
s1已有长度: 99.6974 MB字节, 本次增加 59187 字节
s1已有长度: 99.7538 MB字节, 本次增加 56338 字节
s1已有长度: 99.8075 MB字节, 本次增加 46230 字节
s1已有长度: 99.8516 MB字节, 本次增加 43201 字节
s1已有长度: 99.8928 MB字节, 本次增加 64016 字节
s1已有长度: 99.9539 MB字节, 本次增加 41002 字节
s1已有长度: 99.993 MB字节, 本次增加 51572 字节
time=379.328
内存分配到达满100MB, 测试结束
本次测试耗时 379.328秒
老师的机器运行VS2015编译的程序, 大约耗时360-380秒
【说明】: 只有相同编译器下的运行时间才有可比性
老师的机器为Intel(R) Core(TM) i5-4210U CPU @1.70GHz 2.40GHz
如果时间相差太大, 除CPU的性能差异外, 还有可能是算法问题
打开任务管理器查看程序内存占用情况...
按回车键继续...

任务管理器

文件(F) 选项(O) 查看(V)

进程 性能 应用历史记录 启动 用户 详细信息 服务

名称	CPU	内存	磁盘	网络
应用 (7)				
Araxis Merge	0%	12.9 MB	0 MB/秒	0 Mbps
Microsoft Visual ...	0%	203.1 MB	0 MB/秒	0 Mbps
Microsoft Word (...)	0.3%	86.1 MB	0 MB/秒	0 Mbps
Task Manager	1.2%	19.6 MB	0 MB/秒	0 Mbps
Windows 资源管...	0%	38.1 MB	0 MB/秒	0 Mbps
第10章-TString类(...)	0%	100.6 MB	0 MB/秒	0 Mbps
腾讯QQ (32 位)	0%	102.4 MB	0 MB/秒	0 Mbps
后台进程 (109)				
64-bit Synaptics ...	0%	1.8 MB	0 MB/秒	0 Mbps
AcroTray (32 位)	0%	0.3 MB	0 MB/秒	0 Mbps
Active Protection ...	0%	0.2 MB	0 MB/秒	0 Mbps
Adobe Acrobat U...	0%	0.2 MB	0 MB/秒	0 Mbps

任务管理器

【作业要求:】

- 1、6月7日前网上提交本次作业
- 2、每题所占平时成绩的具体分值见网页
- 3、超过截止时间提交作业会自动扣除相应的分数, 具体见网页上的说明