

封装字符串

Problems with C string

- In C
 - String is represented by an array of characters
 - Character arrays are treated differently than ordinary arrays
- Problems: user must know about some irrelevant details
 - Arrays have bounds
 - Meaningful characters must followed by a numerical zero

How C++ solves these problems

- Implement a class and a set of functions, such that the user
 - Using dynamically allocating character arrays to allow strings of any sizes
 - Encapsulating the character array and provide a set of comprehensive functions
- The results:
 - Need not knowing about array bounds and the zero at the end
 - Intuitive usage analogous to fundamental data types

```
1  #include <iostream>
2  using namespace std;
3
4  int length0(const char text[]) {
5      for (int i = 0; ; ++ i) {
6          if (text[i] == '\\0') return i;
7      }
8  }
9
10 void copy0(char to[], const char from[]) {
11     for (int i = 0; ; ++ i) {
12         to[i] = from[i];
13         if (from[i] == '\\0') break;
14     }
15 }
```

```
17 int main() {  
18     // initialize a 'string' variable  
19     int length1 = length0("C L");  
20     char * name1 = new char[length1 + 1];  
21     copy0(name1, "C L");  
22  
23     int length2 = length0("L, C");  
24     char * name2 = new char[length2 + 1];  
25     copy0(name2, "L, C");  
26  
27     cout << name1 << endl; // C L  
28     cout << name2 << endl; // L, C
```

```
30     // assign
31     delete [] name2;
32     int length3 = length0("C.");
33     name2 = new char[length3 + 1];
34     copy0(name2, "C.");
35
36     // append
37     int length4 = length0(name2);
38     int length5 = length0(" L");
39     char * temp = new char[length4 + length5 + 1];
40     copy0(temp, name2);
41     copy0(temp + length4, " L");
42     delete [] name2;
43     name2 = temp;
44     cout << name2 << endl; // C. L
45
46     // clean up
47     delete [] name2;
48     delete [] name1;
49 }
```

面向对象的C程序

```
17 char * assign0(const char text[]) {  
18     int length = length0(text);  
19     char * thisString = new char[length + 1];  
20     copy0(thisString, text);  
21     return thisString;  
22 }  
23  
24 void construct(char * & thisString,  
25               const char text[]) {  
26     thisString = assign0(text);  
27 }
```

```
29 void assign(char * & thisString,
30             const char text[]) {
31     delete [] thisString;
32     thisString = assign0(text);
33 }
34
35 void append(char * & thisString,
36            const char text[]) {
37     int length1 = length0(thisString);
38     int length2 = length0(text);
39     char * temp = new char[length1 + length2 + 1];
40     copy0(temp, thisString);
41     copy0(temp + length1, text);
42     delete [] thisString;
43     thisString = temp;
44 }
```



```
45
46 void destruct(char * thisString) {
47     delete [] thisString;
48 }
49
50 int main() {
51     // initialize a 'string' variable
52     char * name1 = 0;
53     construct(name1, "Cong Liu");
```

```
58      cout << name1 << endl; // Cong Liu
59      cout << name2 << endl; // Liu, Cong
60
61      assign(name2, "C.");
62
63      append(name2, " Liu");
64      cout << name2 << endl; // C. Liu
65
66      destruct(name2);
67      destruct(name1);
68 }
```

C++的面向对象程序

```
1 class String
2 {
3 private:
4     char * array;
5
6     int length0(const char text[]) const {
7         for (int i = 0; ; ++ i) {
8             if (text[i] == '\0') return i;
9         }
10    }
```

```
12     void copy0(char to[], const char from[]) {
13         for (int i = 0; ; ++ i) {
14             to[i] = from[i];
15             if (from[i] == '\\0') break;
16         }
17     }
18
19     void assign0(const char text[]) {
20         if (array != 0) { // != NULL
21             delete [] array;
22         }
23         int length = length0(text);
24         array = new char[length + 1];
25         copy0(array, text);
26     }
```

```
27
28 public:
29     String() {
30         array = (char *)"";
31         array = new char[1];
32         array[0] = '\\0';
33     }
34
35     String(const char text[]) {
36         array = (char *)text;
37         array = 0; // NULL
38         assign0(text);
39     }
40
41     ~String() {
42         delete [] array;
43     }
```

```
45     const char * c_str() const {
46         return array;
47     }
48
49     void assign(const char text[]) {
50         assign0(text);
51     }
52
53     void append(const char text[]) {
54         int length1 = length0(array);
55         int length2 = length0(text);
56         int length = length1 + length2 + 1;
57         char * array2 = new char[length];
58         copy0(array2, array);
59         copy0(array2 + length1, text);
60         delete [] array;
61         array = array2;
62     }
63
64 };
```

```
66 #include <iostream>
67 using namespace std;
68
69 int main() {
70     String name1("Cong Liu");
71     String name2 = "Liu, Cong";
72     cout << name1.c_str() << endl; // Cong Liu
73     cout << name2.c_str() << endl; // Liu, Cong
74
75     name2.assign("C.");
76     name2.append(" Liu");
77     cout << name2.c_str() << endl; // C. Liu
78 }
```