阅读：10,690　　作者：解学武

串的定长顺序存储结构（C语言）详解版

上一节　　下一节

我们知道，顺序存储结构（顺序表）的底层实现用的是数组，根据创建方式的不同，数组又可分为静态数组和动态数组，因此顺序存储结构的具体实现其实有两种方式。

> 通常所说的数组都指的是静态数组，如 str[10]，静态数组的长度是固定的。与静态数组相对应的，还有动态数组，它使用 malloc 和 free 函数动态申请和释放空间，因此动态数组的长度是可变的。

串的定长顺序存储结构，可以简单地理解为采用 "固定长度的顺序存储结构" 来存储字符串，因此限定了其底层实现只能使用静态数组。

使用定长顺序存储结构存储字符串时，需结合目标字符串的长度，预先申请足够大的内存空间。

例如，采用定长顺序存储结构存储 "data.biancheng.net"，通过目测得知此字符串长度为 18（不包含结束符 '\0'），因此我们申请的数组空间长度至少为 18，用 C 语言表示为：

```
char str[18] = "data.biancheng.net";
```

下面这段 C 语言代码给大家完美地展示了使用定长顺序存储结构存储字符串：

```
01.  #include<stdio.h>
02.  int main()
03.  {
04.      char str[20]="data.biancheng.net";
05.      printf("%s\n",str);
06.      return 0;
07.  }
```

根据实际情况，实现代码可包含一些函数，用于实现某些具体功能，如求字符串的长度等，由于这些知识都是学习编程语言的基础内容，因此不再过多赘述。

上一节　　下一节