

C 上机

字符串及字符串函数

张晓平

武汉大学数学与统计学院

homepage: xpzhang.me

2018 年 5 月 27 日

目录

C 上机

张晓平

目录

字符串与字符串函数

1 字符串与字符串函数

例

设计并测试一个函数 `my_strchr()`，其功能是搜索第一个参数指定的字符串，在其中查找由第二个参数指定的字符首次出现的位置。

- 若找到，返回指向这个字符的指针；
- 若没有找到，返回 `NULL`。

```
1 #include <stdio.h>
2 char * my_strchr(const char * str, char c)
3 ;
4 int main(int argc, char * argv[])
5 {
6     char * p;
7
8     if(argc != 3) {
9         fprintf(stderr, "Usage: %s string
10         character\n", argv[0]);
11         return 1;
12     }
13     p = my_strchr(argv[1], argv[2][0]);
```

```
14  if (p == NULL)
15      printf("Cannot find the character %c!\n", argv[2][0]);
16  else
17      printf("Find the character %c!\n", *p)
18      ;
19  return 0;
20 }
21
22 char * my_strchr(const char * str, char c)
23 {
24
25     while(*str && *str != c)
26         str++;
```

```
27 |     if (*str == c)
28 |         return ((char *) str);
29 |     return NULL;
30 | }
```

例

编写一个函数 `is_within()`，判断指定字符是否在指定字符串中。

- 若在其中，则返回 `true`;
- 否则返回 `false`。

```
1 #include<stdio.h>
2 #include<stdbool.h>
3 bool is_within(const char * str, int c);
4
5 int main(int argc, char * argv[])
6 {
7     char * str;
8     char ch;
9
10    if(argc != 3) {
11        fprintf(stderr, "Usage: %s string
12        character.\n", argv[0]);
13        return 1;
14    }
```



```
15 str = argv[1];
16 ch = argv[2][0];
17
18 if (is_within(str, ch))
19     printf("Find the character %c\n", ch);
20 else
21     printf("Cannot find the character %c!\n", ch);
22
23 return 0;
24 }
25
26 bool is_within(const char * str, int c)
27 {
28     while(*str && *str != c)
```

```
29     str++;  
30     if (*str == c)  
31         return true;  
32     return false;  
33 }
```

例

函数 `strncpy(s1, s2, n)` 从 `s2` 复制 `n` 个字符给 `s1`。

- 若 `s2` 的长度大于或等于 `n`，则目标字符串没有标识结束的空字符。
- 函数返回 `s1`。

请自行编写该函数，取名为 `my_strncpy()`。

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 char * my_strncpy (char * target, const
  char * source, int n);
5
6 int main(int argc, char * argv[])
7 {
8     char * target, * source;
9     int n;
10
11     if(argc != 4) {
12         fprintf(stderr, "Usage: %s target
  source n", argv[0]);
13         return 1;
```

```
14     }  
15  
16     target = argv[1];  
17     source = argv[2];  
18     n = atoi(argv[3]);  
19  
20     my_strncpy(target, source, n);  
21  
22     printf("%s\n", target);  
23  
24     return 0;  
25 }  
26  
27 char * my_strncpy (char * target, const  
    char * source, int n)
```

```
28 {  
29     char *start = target;  
30  
31     while (n && (*target++ = *source++))  
32         n--;  
33  
34     if (n)  
35         while (--n)  
36             *target++ = '\0';  
37  
38     return (start);  
39 }
```

例

编写函数 `my_strstr()`，它接受两个字符串指针参数。

- 若第二个字符串被包含在第一个字符串中，则返回被包含的字符串开始的地址。
- 如，`my_strstr("hats", "at")`返回 `"hats"` 中 `'a'` 的地址，否则返回 `NULL`。

```
1 #include<stdio.h>
2 char * my_strstr(const char * str1, const
  char * str2);
3
4 int main(int argc, char * argv[])
5 {
6     char * str1, * str2;
7     char * p;
8
9     if(argc != 3) {
10         fprintf(stderr, "Usage: %s string
           character.\n", argv[0]);
11         return 1;
12     }
13
```



```
14 str1 = argv[1];  
15 str2 = argv[2];  
16  
17 p = my_strstr(str1, str2);  
18 if (p)  
19     printf("%s\n", p);  
20 else  
21     printf("%s is not contained in %s\n",  
22           str2, str1);  
23  
24 return 0;  
25  
26 }
```

```
27 char * my_strstr(const char * str1, const
    char * str2)
28 {
29     char * start = (char *) str1;
30     char *s1, *s2;
31
32     if (!*str2) // if str2 is empty
33         return((char *) str1);
34
35     while (*start) {
36         s1 = start;
37         s2 = (char *) str2;
38
39         while (*s1 && *s2 && !(*s1 - *s2))
40             s1++, s2++;
```

```
41  
42     if (!*s2)  
43         return start;  
44  
45     start++;  
46 }  
47 return NULL;  
48 }
```