

String

Yung-Hsiang Lu

C String = Array of char

- examples of strings:
 - “Hello”
 - “Good Morning”
 - “I am learning C.”
- static string, created when a program is written
 - `char str1[] = “This is a string.”;`
 - `char * str2 = “Learning C Programming”;`
 - `char str3[] = {‘a’,‘b’,‘c’,‘d’,‘e’, ‘\0’}; /* equivalent to string “abcde” */`

Every string must end with a special character ‘\0’.
- array of char, the string can be changed
 - `char str5[80];`

Functions for Strings

`strlen(str)` \Rightarrow length of a string (not counting `'\0'`)

`strcmp(str1, str2)` \Rightarrow compare two strings using dictionary order (lexicographical order, alphabetic order)

0 if the two strings are the same

1 if `str1 > str2` (i.e. `str1` would appear later in a dictionary)

-1 if `str1 < str2` (i.e. `str1` would appear earlier)

`strcpy(dest, src)` \Rightarrow copy string `src` to `dest`

make sure dest has enough space

`strcat(dest, src)` \Rightarrow append `src` to `dest`

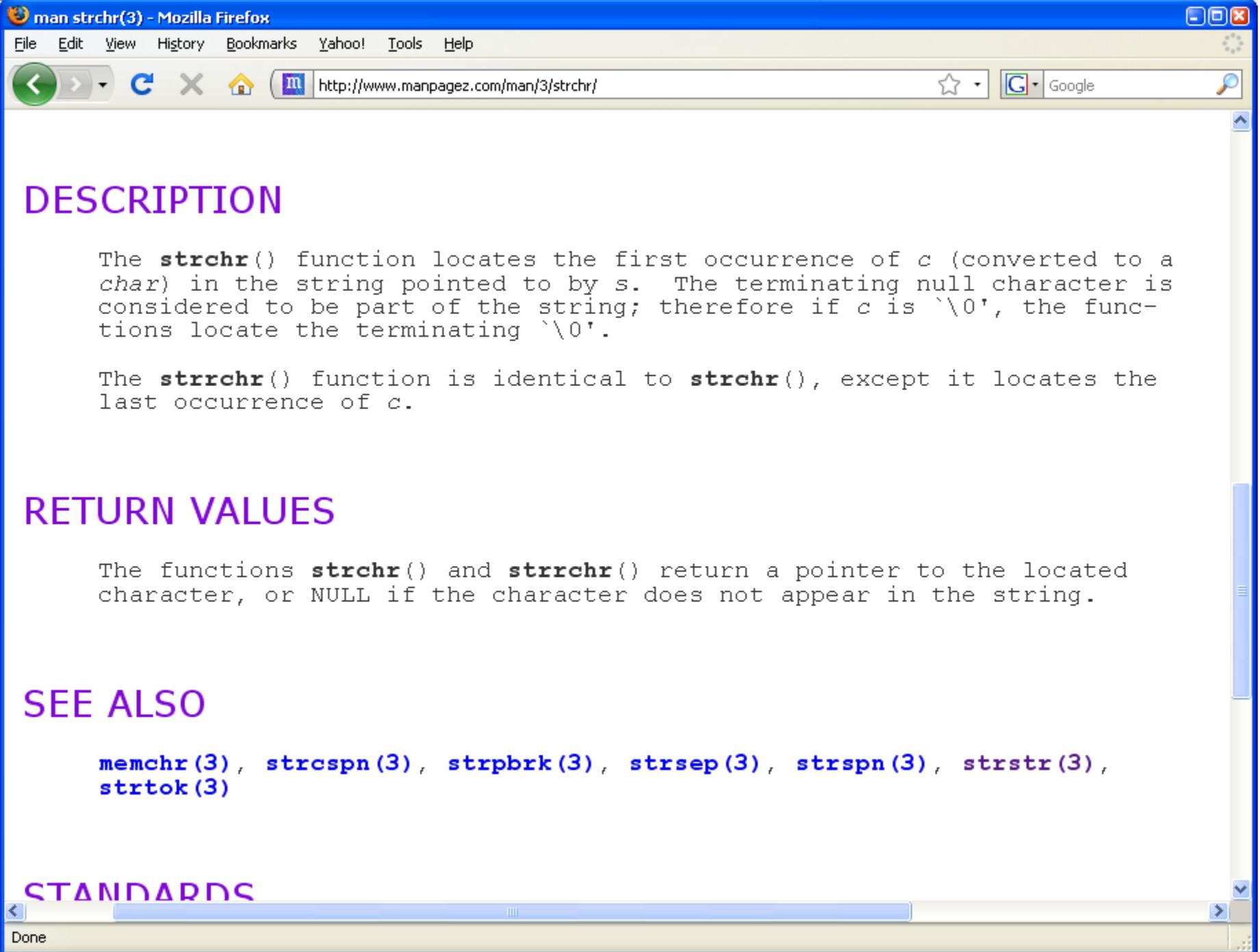
make sure `dest` has enough space

String and Character

`strchr(str, ch)` \Rightarrow return a pointer to the first occurrence of the character. If the string does not contain this character, return `NULL`

`strrchr(str, ch)` \Rightarrow return a pointer to the last occurrence of the character. If the string does not contain this character, return `NULL`

`strstr(str1, str2)` \Rightarrow return a pointer to the first occurrence of the second string. If the string does not contain this string, return `NULL`



DESCRIPTION

The **strchr()** function locates the first occurrence of *c* (converted to a *char*) in the string pointed to by *s*. The terminating null character is considered to be part of the string; therefore if *c* is `\0`, the functions locate the terminating `\0`.

The **strrchr()** function is identical to **strchr()**, except it locates the last occurrence of *c*.

RETURN VALUES

The functions **strchr()** and **strrchr()** return a pointer to the located character, or `NULL` if the character does not appear in the string.

SEE ALSO

[memchr\(3\)](#), [strcspn\(3\)](#), [strpbrk\(3\)](#), [strsep\(3\)](#), [strspn\(3\)](#), [strstr\(3\)](#), [strtok\(3\)](#)

STANDARDS

man strstr(3) - Mozilla Firefox

File Edit View History Bookmarks Yahoo! Tools Help

http://www.manpagez.com/man/3/strstr/ Google

```
#include <string.h>
#include <xlocale.h>

char *
strcasestr_1(const char *s1, const char *s2, locale_t loc);
```

DESCRIPTION

The **strstr()** function locates the first occurrence of the null-terminated string *s2* in the null-terminated string *s1*.

The **strcasestr()** function is similar to **strstr()**, but ignores the case of both strings.

The **strnstr()** function locates the first occurrence of the null-terminated string *s2* in the string *s1*, where not more than *n* characters are searched. Characters that appear after a `\0` character are not searched. Since the **strnstr()** function is a FreeBSD specific API, it should only be used when portability is not a concern.

While the **strcasestr()** function uses the current locale, the **strcasestr_1()** function may be passed a locale directly. See [xlocale\(3\)](#) for more information.

RETURN VALUES

Done

1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2
L e a r n i n g C P r o g r a m m i n g

string.c

```
#include <stdio.h>
#include <string.h>
int main(int argc, char * argv[])
{
    char * str1 = "Good Morning";
    char str2[] = "Learning C Programming";
    char str3[] = {'a','b','c','d','\0'};
    printf("%s\n", str1);
    printf("strlen(str2) = %d\n", strlen(str2));
    printf("strchr(str3, 'c') = %s\n", strchr(str3, 'c'));
    char str4[80];
    strcpy(str4, str1);
    strcat(str4, str2);
    printf("str4 = %s\n", str4);
    printf("strstr(str4, \"ing\") = %s\n", strstr(str4, "ing"));
    return 0;
}
```

Console

<terminated> Function (3) [C/C++ Local Application] /home/yunglu/workspace/String/Debug/Function (09/08/2009 11:23 PM)

```
Good Morning
strlen(str2) = 22
strchr(str3, 'c') = cd
str4 = Good MorningLearning C Programming
strstr(str4, "ing") = ingLearning C Programming
```

Writable

Smart Insert

18 : 1

The screenshot shows the Eclipse IDE interface. The main editor window displays the source code for a C program named `strlen.c`. The code defines a function `strlen2` that counts the length of a string by iterating through characters until a null terminator is reached. The `main` function tests this function with two strings: "Good" and "Learning C Programming".

```
#include <stdio.h>
#include <string.h>
int strlen2(char * s)
{
    int len = 0;
    while ((*s) != '\0')
    { len ++; s++; }
    return len;
}
int main(int argc, char * argv[])
{
    char * str1 = "Good";
    char str2[] = "Learning C Programming";
    printf("strlen2(%s) = %d\n", str1, strlen2(str1));
    printf("strlen2(%s) = %d\n", str2, strlen2(str2));
    return 0;
}
```

The console window below the editor shows the output of the program:

```
<terminated> Function (3) [C/C++ Local Application] /home/yunglu/workspace/String/Debug/Function (09/09/2009 6:
strlen2(Good) = 4
strlen2(Learning C Programming) = 22
```

The status bar at the bottom of the IDE indicates the current file is writable, has smart insert enabled, and the cursor is at line 18, column 1.

Increment a Pointer

	str1	sp																	
index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14				
char	H	e	l	l	o		C		P	r	o	g	r	a	m				

The screenshot shows the Eclipse IDE with a C program named 'strlen.c'. The code defines a character array 'str1' containing 'Hello C Program' and a pointer 'sp' pointing to the first element of 'str1'. The main function prints the character at the address pointed to by 'sp' three times, incrementing 'sp' each time. The first print outputs 'H', the second outputs 'e', and the third outputs 'r'. A green arrow in the code points to the 'sp++' line, and another green arrow in the output window points to the 'e' character.

```
#include <stdio.h>
#include <string.h>
int main(int argc, char * argv[])
{
    char str1[] = "Hello C Program";
    char * sp = & str1[0];
    printf("%c\n", (* sp));
    sp ++;
    printf("%c\n", (* sp));
    sp += 8;
    printf("%c\n", (* sp));
    return 0;
}
```

Writable | Smart Insert | 14 : 1

Increment a Pointer

	C/C++ - strlen.c - Eclipse SDK														
	str1									sp					
index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
char	H	e	l	l	o		C		P	r	o	g	r	a	m

```
#include <stdio.h>
#include <string.h>
int main(int argc, char * argv[])
{
    char str1[] = "Hello C Program";
    char * sp = & str1[0];
    printf("%c\n", (* sp));
    sp ++;
    printf("%c\n", (* sp));
    sp += 8;
    printf("%c\n", (* sp));
    return 0;
}
```

Writable | Smart Insert | 14 : 1

<terminated> Function (3) [I
H
e
r

What is the output?

```
char str1[] = "Nice Day."  
char * sp = str1;  
sp += 2;  
printf("%c\n", *(sp));
```

The output is .

Correct - Click anywhere to
continue

Incorrect - Click anywhere to
continue

Your answer:

You did not answer this question

You must answer the question
before continuing

Submit

Clear

String

Your Score	{score}
Max Score	{max-score}
Number of Quiz Attempts	{total-attempts}

Question Feedback/Review Information Will Appear Here

Continue

Review Quiz