

Strings in C

Course: Introduction to Programming and Data Structures

Laltu Sardar

Institute for Advancing Intelligence (IAI),
TCG Centres for Research and Education in Science and Technology (TCG Crest)

tcg crest

Inventing Harmonious Future

Table of Contents

1 Strings

- String Declaration and Initialization
- Some common Operations on Strings
 - String Length: `strlen`
 - String Copy: `strcpy`
 - String Concatenation: `strcat`

2 Array of Strings

Strings

Introduction

- Strings are a fundamental concept in C programming.
- In C, strings are represented as arrays of characters.
- Strings can be accessed using pointers. A pointer to a string is a variable that stores the address of the first character in the string.
- C-style strings are null-terminated, meaning they are terminated by a null character (`\0`).

String Declaration and Initialization

- Strings can be declared and initialized in various ways:
 - `char str[] = "Hello";`
 - `char str[10] = "Hello";`
 - `char *str = "Hello";`
- The size of the array should accommodate the string length plus one for the null character.

Differences

- `char str[] = "Hello";`: Creates a modifiable array in stack memory.
- `char *str = "Hello";`: Creates a pointer to a read-only string literal, stored in static memory.

Some common Operations on Strings

String Length: strlen

The `strlen` function is used to find the length of a string.

```
size_t strlen(const char *str);
```

Example:

```
1  #include <stdio.h>
2  #include <string.h>
3
4  int main() {
5      char str[] = "Hello , World!";
6      int len = strlen(str);
7      printf("Length of the string: %d\n", len);
8      return 0;
9  }
```

String Copy: strcpy

The strcpy function is used to copy one string to another.

```
char *strcpy(char *dest, const char *src);
```

Example:

```
1  #include <stdio.h>
2  #include <string.h>
3
4  int main() {
5      char src [] = "Hello , World!";
6      char dest [20];
7      strcpy(dest , src);
8      printf("Copied string: %s\n", dest);
9      return 0;
10 }
```


String Concatenation: strcat

The `strcat` function is used to concatenate two strings.

```
char *strcat(char *dest, const char *src);
```

Example:

```
1  #include <stdio.h>
2  #include <string.h>
3
4  int main() {
5      char str1[20] = "Hello";
6      char str2[] = ", World!";
7      strcat(str1, str2);
8      printf("Concatenated string: %s\n", str1);
9      return 0;
10 }
```

String Comparison: strcmp

The strcmp function is used to compare two strings.

```
int strcmp(const char *str1, const char *str2);
```

Example:

```
1  #include <stdio.h>
2  #include <string.h>
3
4  int main() {
5      char str1 [] = "Hello";
6      char str2 [] = "Hello";
7      if (strcmp(str1, str2) == 0) {
8          printf("Strings are equal.\n");
9      } else {
10         printf("Strings are not equal.\n");
11     }
12     return 0;
13 }
```

String Search: strstr

The `strstr` function is used to find the first occurrence of a substring in a string.

```
char *strstr(const char *haystack, const char *needle);
```

Example:

```
1  #include <stdio.h>
2  #include <string.h>
3
4  int main() {
5      char str[] = "Hello , World!";
6      char *pos = strstr(str, "World");
7      if (pos != NULL) {
8          printf("Substring found at position: %ld\n", pos - str);
9      } else {
10         printf("Substring not found.\n");
11     }
12     return 0;
13 }
```

String Functions

- C provides a set of functions in the `<string.h>` library for string manipulation:
 - `strlen()`
 - `strcpy()` and `strncpy()`
 - `strcat()` and `strncat()`
 - `strcmp()` and `strncmp()`
 - `strstr()` and `strchr()`
 - `sprintf()` and `sscanf()`

Some Other common Operations on Strings

There are many operations that can be performed on strings in C. Some of the most common operations include:

- Searching for a substring in a string: This operation returns the index of the first occurrence of a substring in a string.
- Replacing a substring in a string: This operation replaces all occurrences of a substring in a string with another substring.
- Sorting the characters in a string: This operation sorts the characters in a string in alphabetical order.

Array of Strings

Declaration and Initialization

- Declaring an array of strings:
 - `char names[5][20];`
 - `char cities[3][15];`
- Initializing the array of strings:
 - `char fruits[][10] = {"apple", "banana", "cherry"};`

Accessing and Modifying Elements

- Accessing individual strings: `names[2]`
- Modifying strings: `strcpy(names[1], "John");`
- Using loops for batch operations:
 - `for (int i = 0; i < 3; i++) { strcpy(cities[i], "Unknown"); }`

Multidimensional Arrays vs. Array of Strings

- Multidimensional arrays: Elements are of the same data type (e.g., `int`).
- Array of strings: Elements are arrays themselves (`char` arrays).
- Array of strings allows flexibility in handling variable-length text.

Thank You

for your attention.

Questions?

tcg crest

Inventing Harmonious Future

Laltu Sardar

laltu {dot} sardar [at]tcgcrest(.)org
<https://laltu-sardar.github.io>