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)

August 29, 2023

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.

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:

```c
#include <stdio.h>
#include <string.h>

int main() {
  char str[] = "Hello, World!";
  int len = strlen(str);
  printf("Length of the string: %d\n", len);
  return 0;
}
```

String Copy: strcpy

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

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

Example:

```c
#include <stdio.h>
#include <string.h>

int main() {
  char src[] = "Hello, World!";
  char dest[20];
  strcpy(dest, src);
  printf("Copied string: %s\n", dest);
  return 0;
}
```

String Concatenation: strcat

The strcat function is used to concatenate two strings.

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

Example:

```c
    #include <stdio.h>
    #include <string.h>

    int main() {
      char str1[20] = "Hello";
      char str2[] = ", World!";
      strcat(str1, str2);
      printf("Concatenated string: %s\n", str1);
      return 0;
    }
```

String Comparison: strcmp

The strcmp function is used to compare two strings.

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

Example:

```c
#include <stdio.h>
#include <string.h>

int main() {
  char str1[] = "Hello";
  char str2[] = "Hello";
  if (strcmp(str1, str2) == 0) {
    printf("Strings are equal.\n");
  } else {
    printf("Strings are not equal.\n");
  }
  return 0;
}
```

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:

```c
    #include <stdio.h>
    #include <string.h>

    int main() {
      char str[] = "Hello, World!";
      char *pos = strstr(str, "World");
      if (pos != NULL) {
        printf("Substring found at position: %ld\n", pos - str);
      } else {
        printf("Substring not found.\n");
      }
      return 0;
    }
```

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.