

11. Strings in C

- **THEORY EXERCISE:** Explain string handling functions like `strlen()`, `strcpy()`, `strcat()`, `strcmp()`, and `strchr()`. Provide examples of when these functions are useful.

1. `strlen()`

Description: The `strlen()` function calculates the length of a string (excluding the null terminator).

Syntax:

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

Example:

```
#include <stdio.h>

#include <string.h>

int main() {

    const char *str = "Hello, World!";

    size_t length = strlen(str);

    printf("Length of the string: %zu\n", length); // Output: 13

}
```

2. `strcpy()`

Description: The `strcpy()` function copies a string from the source to the destination.

Syntax:

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

Example:

```
#include <stdio.h>

#include <string.h>

int main() {

    char source[] = "Hello, World!";

    char destination[50]; // Ensure destination is large enough

    strcpy(destination, source);

    printf("Copied string: %s\n", destination); // Output: Hello, World!

}
```

3. `Strcat()`

Description: The **strcat()** function concatenates (appends) one string to the end of another.

Syntax:

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

Example:

```
#include <stdio.h>

#include <string.h>

int main() {

    char str1[50] = "Hello, ";

    char str2[] = "World!";

    strcat(str1, str2);

    printf("Concatenated string: %s\n", str1); // Output: Hello, World!

}
```

4.strcmp()

Description: The **strcmp()** function compares two strings lexicographically.

Syntax:

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

Example:

```
#include <stdio.h>

#include <string.h>

int main() {

    const char *str1 = "Hello";

    const char *str2 = "World";

    int result = strcmp(str1, str2);

    if (result < 0) {

        printf("%s is less than %s\n", str1, str2);

    } else if (result > 0) {

        printf("%s is greater than %s\n", str1, str2);

    } else {

        printf("%s is equal to %s\n", str1, str2);

    }

}
```

```
}
```

Description: The **strchr()** function locates the first occurrence of a character in a string.

Syntax:

```
char *strchr(const char *str, int character);
```

Example:

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

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

```
int main() {
```

```
    const char *str = "Hello, World!";
```

```
    char ch = 'o';
```

```
    char *ptr = strchr(str, ch);
```

```
    if (ptr != NULL) {
```

```
        printf("First occurrence of '%c': %s\n", ch, ptr); // Output: o, World!
```

```
    } else {
```

```
        printf("Character not found.\n");
```

```
    }
```

```
}
```

Use Case: **strchr()** is useful for parsing strings, such as finding delimiters in a CSV file or searching for specific characters in user input.