6. Pointers and Strings

In C, strings are arrays of characters, so pointers work naturally with strings.

6.1 String Representation

```
char str1[] = "Hello";     // Array notation
char *str2 = "Hello";      // Pointer notation

// Both represent the same thing in memory:
// 'H' 'e' 'l' 'l' 'o' '\0'
```

Memory Layout:

```
str1: char array (modifiable)
┌───┬───┬───┬───┬───┬────┐
| H | e | l | l | o | \0 |
└───┴───┴───┴───┴───┴────┘


str2: pointer to string literal (read-only)
     ┌────────┐
str2 | addr   |─────> "Hello\0" (in read-only memory)
     └────────┘
```

6.2 String Traversal Using Pointers

```
#include <stdio.h>

void printString(char *str) {
    while (*str != '\0') {  // Until null terminator
        printf("%c", *str);
        str++;  // Move to next character
    }
    printf("\n");
}
```

```
int main() {
    char message[] = "Hello, World!";
    printString(message);
    return 0;
}
```

6.3 String Length Using Pointers

```
#include <stdio.h>

int stringLength(char *str) {
    int length = 0;
    while (*str != '\0') {
        length++;
        str++;
    }
    return length;
}

// Alternative using pointer arithmetic
int stringLength2(char *str) {
    char *start = str;
    while (*str != '\0') {
        str++;
    }
    return str - start;  // Pointer subtraction
}

int main() {
    char text[] = "Programming";
    printf("Length: %d\n", stringLength(text));
    return 0;
}
```

6.4 String Copy Using Pointers

```
#include <stdio.h>

void stringCopy(char *dest, char *src) {
```

```
    while (*src != '\0') {
        *dest = *src;
        dest++;
        src++;
    }
    *dest = '\0';  // Don't forget null terminator!
}

// More elegant version
void stringCopy2(char *dest, char *src) {
    while ((*dest++ = *src++));  // Copy until '\0' (which is 0/false)
}

int main() {
    char source[] = "Hello";
    char destination[50];

    stringCopy(destination, source);
    printf("Copied string: %s\n", destination);

    return 0;
}
```

Revision #1
Created 2025-10-01 03:28:01 UTC by DS
Updated 2025-10-01 03:28:23 UTC by DS