

Lab 7: Functions and string manipulation

Goals

Understand how to build simple C language functions and use them to process null-terminated text strings.

Background

Strings

- Character strings (or simply strings) are arrays of *char* variables.
- They are used in a C language program to store words or sentences in the English (or some other) language.
- Some programming languages have an actual variable type called *string*, but C does not.
- The logical end of a C language string is denoted by the presence of a byte consisting of all 0 bits which is sometimes called a *NULL byte*.
- The program must explicitly provide storage for the NULL byte.
- Thus, a character array that can hold the string "Hello" must be at least 6 bytes in size: `char hello[6];`

String functions

The C standard library contains a collection of functions that can be used to manipulate NULL terminated strings.

- To use them `#include <string.h>`
- `strlen(char str[]);` returns the length of a string
- `strcat(char str1[], char str2[]);` catenates `str2` onto the end of `str1`.
- `strcpy(char str1[], char str2[]);` copies `str2` over the top of `str1`.
- `strcmp(char str1[], char str2[]);` compares `str1` and `str2` and returns 0 if they are the same length and identical in all character positions.

Assignment:

Write a program called lab7.c that contains three functions as follows:

1. Create a function `int my_strlen (char s1[])` that will compute and return the length of a string s1. A single while loop should be used to count the characters one at a time. Processing should terminate when the NULL terminator is encountered (`s1[ndx] == 0`)
2. Create a `main()` function to test `my_strlen()`.
 - a. Define three character arrays v1 and v2 of *length 16* and two integer variables l1 and l2.
 - b. Read in two strings (maximum length 7) into v1 and v2 from the standard input. Invoke `my_strlen()` to set l1 and l2 to the lengths of v1 and v2 respectively. Use the format string `"len: %3d - string: %s \n"` to print l1, v1 and then l2, v2 to the standard output.
3. Create a function `void my_strcat (char s1[], char s2[])` that will concatenate s2 onto the end of s1. Two while loops are required here. The first should identify the end of s1. The second should be used to copy the characters of s2 one at a time. Copying should end when `s2[ndx] == 0`, but *care should be taken to ensure that the concatenated string is properly terminated with a NULL.*
4. Augment your `main()` function so that it will concatenate string v2 onto v1, and print v1 using the format string "%s".

Turn In Work

Show your TA that you completed the assignment. Then turn in your lab6.c program using the command:

```
sendlab.101.section_number 7 lab7.c
```