

Para Universidad Autónoma de Baja California  
Facultad de Ciencias Químicas e Ingeniería



Alumno: Eduardo Marcelo Gutiérrez Soto  
Profesora: Evangelina Lara Camacho  
Practica 12 Lenguaje C y Ensamblador Manipulación de cadenas

**Teoría:**

Describe por medio de dos ejemplos cómo están implementadas las funciones **strcmp** y **strchr** en la biblioteca de funciones **string.h** de C.

**STRCMP:**

**BIBLIOTECA:** string.h

**PARÁMETROS:** int strcmp(const char \*cad1, const char \*cad2);

**VALOR DEVUELTO:** Un numero entero

lo que realiza es comparar dos cadenas si son iguales o si es mayor una que la otra y funciona de la siguiente manera, Si la primer cadena es mayor que la segunda retorna un valor positivo 1, Si es mayor la segunda cadena devuelve un valor negativo -1, y si las cadenas son iguales nos retorna un valor entero 0. y todo eso lo realiza dependiendo la suma de sus caracteres en valores ascii.

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

int main()
{
    char * cad1 = {"abcd"};
    char * cad2 = {"efgh"};

    if(strcmp(cad1,cad2) == 0)
    {
        printf("Son iguales 0");
    } else if(strcmp(cad1,cad2) == 1)
    {
        printf("la primer cadena es mayor 1");
    } else if(strcmp(cad1,cad2) == -1)
    {
        printf("la segunda cadena es mayor -1");
    }
}
```

```
C:\ocle\C>a.exe
la segunda cadena es mayor -1
C:\ocle\C>
```

**STRCHR:**

**BIBLIOTECA:** string.h

**DECLARACIÓN:** char \*strchr(const char \*cadena, int caracter);

**PARÁMETROS:** la cadena a analizar, la letra que se busca.

**VALOR DEVUELTO:** La posición (PUNTERO) o NULL

**DETALLE:** devuelve NULL si no la contiene, o un puntero a la posición en que se encuentra.

```
if (strchr(texto,letra) == NULL) puts("No la contiene");
```

## Desarrollo 1.

Agregue al programa ManipulacionCadenas.asm que desarrolló en la Práctica 10 los siguientes procedimientos:

a) my\_strcmp: compara dos cadenas lexicográfica-mente. Recibe como parámetros dos apuntadores a cadenas de caracteres, str1 y str2, y retorna en AX un 0 si las cadenas son iguales, un valor  $< 0$  si str1 es menor que str2, o un valor  $> 0$  de lo contrario. La comparación es acorde al orden de los caracteres en la tabla ASCII.

## CÓDIGO ASM:

este es el código que realiza lo que es el STRCMP pero también se hizo de un procedimiento extra que hace las sumas de carácter por carácter de la cadena para poder así comparar las dos cadenas correspondientes

```
;compara que dos cadenas lexicograficamente

_my_strcmp proc
    push bp
    mov bp,sp
    mov si,[bp+4] ;str1
    mov di,[bp+6] ;str2
    mov bx,0
    mov cx,0 ;inicializamos registros en 0

    ;primero recibe el valor de la cadena 1
    call suma_ASCII
    mov cx,ax ;movemos el valor de la cadena 1 en cx

    mov si,di ;movemos la cadena que tenemos en DI a SI
    call suma_ASCII
    ;recibimos el valor en el registro AX

    cmp cx,ax ;comparamos las cadenas 1 con la 2
    jz iguales ;si el valor es cero es que las cadenas son iguales
    cmp cx,ax
    ja cad1_mayor ;salta si la cadena 1 es mayor
    cmp cx,ax
    jb cad2_mayor

cad2_mayor:
    mov ax,-1 ;si ninguna de las condiciones de salto se cumplen
    jmp @@fin
cad1_mayor:
    mov ax,1
    jmp @@fin
iguales:
    mov ax,0

    @@fin:
    pop bp
    ret
_my_strcmp endp
```

## PROCEDIMIENTO EXTRA:

```
                                ;recibe una cadena en SI y retorna el valor en AX
suma_ASCII proc ;procedimineto para sacar el valor ascii total de una cad
push si
push bx
push cx ;salvamos registros

mov bx,0
mov cx,0
mov ax,0
nxt:  cmp byte ptr [si+bx],00h
      je salir
      mov ax,[si+bx] ;movemos el valor apuntado por [si+bx]
      add cx,ax      ;sumamos los valores ascii
      inc bx
      jmp nxt

      salir:
      mov ax,cx      ;retornamos el valor en AX

      pop cx ;recuperamos registros
      pop bx
      pop si
      ret
      endp
```

lo que realizamos con este procedimiento es agarrar el carácter actualmente apuntado a la cadena y lo mandamos al registro AX y empezamos a sumar para poder así sacar el valor de la cadena y lo retornamos por AX, para poder usar ese valor en el main par apoder hacer las respectivas comparaciones y así determinar el valor de la cadena que estamos utilizando

## EJECUCIÓN:

```
C:\C>P12-A.EXE
Ingresa una cadena: abcd
Ingresa otra cadena: efgh
La primer cadena es menor
C:\C>_
```

```
C:\C>P12-A.EXE
Ingresa una cadena: abg
Ingresa otra cadena: abg
Las cadenas son iguales
C:\C>_
```

```
C:\C>P12-A.EXE
Ingresa una cadena: enola
Ingresa otra cadena: eduardo
La primer cadena es menor
C:\C>_
```

b) my\_strchr: retorna la posición de un carácter en una cadena. Recibe como parámetros un apuntador a una cadena y un carácter. La función retorna en AX un apuntador a la primera aparición del carácter dentro de la cadena, o regresa un 0 si no se encuentra.

### CÓDIGO:

```
_my_strchr proc
    push bp
    mov bp,sp
    mov si,[bp+4]    ;str1
    mov ax,[bp+6]    ;caracter
    mov bx,0         ;inicializamos bx en 0

    buscar: cmp byte ptr [si+bx],00h    ;verificamos si llegamos al final
            je no_esta                 ;si es NULL no esta la letra
            cmp byte ptr [si+bx],al    ;movemos un caracter a al
            je retornar                ;si lo encuentra retorna
            inc bx                     ;incrementamos la siguiente pos
            jmp buscar

    no_esta:
        mov ax,0                     ;retornamos un 0
        jmp bye

    retornar:
        add si,bx                    ;sumamos el offset de BX
        mov ax,si;retornamos la direccion donde se encontro el valor
    bye:
        pop bp
        ret
_my_strchr endp
end
```

lo que se realizo fue recibir una cadena y un carácter y lo que hacíamos era ir buscando ese carácter en la cadena y si el carácter era encontrado retornamos la posición exacta donde se encontró, caso contrario de que no se encontrara retornamos un valor 0.

### EJECUCIÓN:

```
C:\C>P12-B.EXE
Ingresa una cadena: enola
Ingresa un caracter: n
Buscando el caracter n en enola...

se encontro a partir de nola
fin de la busqueda
C:\C>_
```

```
C:\C>P12-B.EXE
Ingresa una cadena: PROFESORA_EUANGELINA
Ingresa un caracter: E
Buscando el caracter E en PROFESORA_EUANGELINA...

se encontro a partir de ESORA_EUANGELINA
se encontro a partir de EUANGELINA
se encontro a partir de ELINA
fin de la busqueda
C:\C>_
```

**Conclusiones y comentarios.**

Al realizar esta practica me recordó mucho cuando estaba en programación estructurada ya que utilizaba mucho ese tipo de funciones y programarlas en ensamblador se me hizo divertido y muy fácil ya que con el hecho de saber como funcionan en C, se me facilita mas el como realizarlas en ensamblador ya que tengo la idea de como hacerlo.

**Dificultades.**

Ninguna