Práctica 11

Interfaz entre Lenguaje C y Lenguaje Ensamblador

Objetivo
El alumno se familiarizará con el desarrollo de programas donde se interface el lenguaje C y lenguaje ensamblador lo cual le permitirá optimizar sus aplicaciones.

Equipo
Computadora personal con el software TCC, TASM y TLINK.

Teoría
Responda las siguientes preguntas:

¿Qué significa la palabra reservada *extern* en lenguaje C?

¿Qué significa la palabra reservada *public* en lenguaje ensamblador?

Al realizar una interfaz entre código en lenguaje C y en lenguaje ensamblador:

- ¿De qué manera se pasan los parámetros a una función?
- Cuando se pasa más de un parámetro a una función, ¿en qué orden se envían?
- ¿De qué manera se retorna un valor menor o igual a 16 bits?
- ¿De qué manera se retorna un valor mayor a 16 bits, pero menor o igual a 32 bits?

Desarrollo
1. Cree los programas **myputc.asm** y **P11.c** que contengan el código del Listado 1.
2. Compile, ensamble y encadene mediante la línea de comando:
```
C:\OCLE>tcc -ms -f- P11.c myputc.asm
```
3. Ejecute el archivo generado P12.exe el cual desplegará el mensaje "Hola Mundo".

4. Cree el programa **Pal.asm** que contiene la rutina pública **Palindromo** implementada en lenguaje Ensamblador. Esta rutina recibe como parámetro una **cadena** y retorna en **AX** un **1** si ésta es palíndroma, de lo contrario retorna **0**. Cree el programa **P11-1.c** en lenguaje C el cual invoca la función externa **Palindromo**, enviándole como parámetro una cadena de caracteres declarada en C. Después de invocar la función, despliega en pantalla si ésta es o no palíndroma.

5. Basándose en las rutinas que desarrolló de tarea sobre manipulación de cadenas: **erase**, **substr**, **strcmp**, **strchr**, **strstr**, cree el programa **ManipulacionCadenas.asm** que contiene las siguientes rutinas públicas:
   a) **my_erase:** borra una porción de una cadena. Recibe como parámetros un apuntador a una **cadena**, un entero que representa la **posición** del primer carácter a

borrar y un entero con la **cantidad** de caracteres a borrar. Si la cadena es más corta que los caracteres solicitados, el procedimiento borra todos los posibles. Si la posición inicial es mayor que la longitud de la cadena, el procedimiento retorna un **-1** en **AX**, caso contrario retorna la cantidad de caracteres que pudo borrar.

b) **my_substr:** almacena en una cadena una copia de una porción de otra cadena. Recibe como parámetros una **cadena fuente**, una **cadena destino**, la **posición inicial** a copiar y la **cantidad** de caracteres. Si la cadena fuente es más corta que los caracteres solicitados, el procedimiento copia todos los posibles. Si la posición inicial es mayor que la longitud de la cadena, el procedimiento retorna un **-1** en **AX**, caso contrario retorna la cantidad de caracteres que pudo copiar.

c) **my_strcmp:** compara dos cadenas lexicográficamente. Recibe como parámetros **dos cadenas** y retorna en **AX** un **0** si son iguales, un valor **< 0** si la primera cadena es menor que la segunda, o un valor **> 0** de lo contrario.

d) **my_strchr:** retorna la posición de un carácter en una cadena. Recibe como parámetros una **cadena** y un **carácter** a buscar. Retorna en **AX** la posición del carácter en la cadena (la primera que encontró) o un **0** (NULL) si no se encuentra en la cadena.

e) **my_strstr**: retorna la posición de una cadena en otra cadena. Recibe como parámetros **dos cadenas**. Retorna en **AX** la posición en que está la segunda cadena en la primera, o un **0** (NULL) si no se encuentra en la cadena.

Cree el programa **P11-2.c** en lenguaje C el cual ejemplifica el uso de todas las funciones externas descritas anteriormente. Envíe como parámetros cadenas declaradas en C. Despliegue en pantalla el valor que retornó cada función y la cadena resultante en el caso de **my_erase, my_substr, my_strchr** (si el apuntador es diferente a NULL) y **my_strstr** (si el apuntador es diferente a NULL).

6. Cree el programa **ConjCollatz.asm** que contiene la rutina pública **ConjeturaDeCollatz** implementada en lenguaje Ensamblador. Esta rutina recibe como parámetro un número de 8 bits y retorna en **AX** el número de iteraciones necesarias para llegar al valor 1 en base a:

$$a_n = \begin{cases} \frac{1}{2}a_{n-1}, & si\ a_{n-1}\ es\ par \\ 3a_{n-1}+1, & si\ a_{n-1}\ es\ impar \end{cases}$$

Ejemplo:
Sea $\boldsymbol{a_0 = 17}$, la secuencia obtenida es 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1.
Que corresponde a **12** iteraciones.

Cree el programa **P11-3.c** en lenguaje C el cual invoca la función externa **ConjeturaDeCollatz**, enviándole como parámetro un entero de 8 bits positivo. Después de invocar la función despliega en pantalla el valor retornado.

Listado 1.

myputc.asm

```
dosseg
.model small
.code
    public  _myputchar

    _myputchar  PROC
                push bp
                mov bp,sp

                mov dl,[bp+4]
                mov ah,2
                int 21h

                pop bp
                ret
    _myputchar ENDP

END
```

P11.c

```
extern void myputchar( char x );

char * str ={"Hola Mundo!!\n"};

void main ( void )
{
    while(*str) {
        myputchar(*str++);
    }

    getchar();
}
```

Conclusiones y comentarios

Dificultades en el desarrollo

Referencias