Práctica 10

Interface 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 interface 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 **P10-1.c** que contengan el código del Listado 1. Compile, ensamble y encadene mediante la línea de comando:
   ```
   C:\OCLE>tcc -ms -f- P10-1.c myputc.asm
   ```
   Ejecute el archivo generado **P10-1.exe**, el cual desplegará el mensaje "Hola Mundo".

2. Cree el programa **ManipulacionCadenas.asm**, el cual contiene las siguientes rutinas públicas implementadas en lenguaje Ensamblador, y que van a ser invocadas desde lenguaje C:

   a) **my_erase:** borra una porción de una cadena. Recibe como parámetros un apuntador a una **cadena**, un apuntador a la **posición inicial** a borrar y un entero con la **cantidad** de caracteres a borrar a partir de esa posición. Si la cadena es más corta que los caracteres solicitados, el procedimiento borra todos los posibles. Si la posición inicial está fuera del límite de la cadena, el procedimiento retorna un **-1** en **AX**, en caso contrario retorna la cantidad de caracteres que pudo borrar.

Su declaración externa en C es: ***int my_erase(char * cadena, char * posicion, int cantidad)***.

Ejemplo:

```
char * str = "Hola mundo!!";
int res;
res = my_erase(str, str+2, 4);
printf("Se borraron %d caracteres y la cadena resultante es: %s",
res, str);
```

En pantalla se muestra:

```
Se borraron 4 caracteres y la cadena resultante es: Houndo!!
```

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

Su declaración externa en C es: ***int my_substr(char * fuente, char * destino, char * posicion, int cantidad)***.

Ejemplo:

```
char * str = "Hola mundo!!";
char destino[32];
int res;
res = my_substr(str, destino, str,2);
printf("Se copiaron %d caracteres y la cadena resultante es: %s",
res, destino);
```

En pantalla se muestra:

```
Se copiaron 2 caracteres y la cadena resultante es: Ho
```

Cree el programa **P10-2.c** en lenguaje C, donde ejemplifique el uso de las funciones externas descritas anteriormente. Codifique diferentes invocaciones para cada función, incluya también los casos en que la función retorna error (-1). Para cada invocación, despliegue en pantalla el valor que retornó la función y la cadena resultante.

3. Cree el programa **ConjCollatz.asm**, el cual contiene la rutina pública **ConjeturaDeCollatz** implementada en lenguaje Ensamblador, y que va a ser invocada desde lenguaje C. Esta rutina recibe como parámetro un número de 8 bits,

denominado **a**, y retorna en **AX** el número de iteraciones necesarias para llegar al valor 1 en base a la fórmula:

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

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 **P10-3.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, se despliega en pantalla el valor retornado.

Ejemplo:
```
int a = 17, iteraciones;
iteraciones = ConjeturaDeCollatz(a);
printf("Se hicieron %d iteraciones para el numero %d", iteraciones,
a);
```

En pantalla se muestra:
```
Se hicieron 12 iteraciones para el numero 17
```

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
```

P10-1.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