x86: Tercera Práctica
Llamada a subprogramas

Departamento de Ingeniería en Automática, Electrónica,

Arquitectura y Redes de Computadores

Universidad de Cádiz

Autor: Jesús Relinque Madroñal
Supervisora: Mercedes Rodríguez García

En esta práctica se utilizará el Entorno Integrado de Desarrollo (IDE) **Easy Code** creado por Ramón Sala, que se puede encontrar en la web: `http://www.easycode.cat/Spanish/Download.htm`. Se utilizará la versión para **GoAsm**.

- En el entorno se integra un editor junto con el ensamblador **GoAsm** y el enlazador **GoLink**, creados por Jeremy Gordon. Ambos pueden hallarse, junto con la correspondiente documentación, en la web:
  `http://www.godevtool.com/`. También pueden ser descargadas del campus. Aparte puede usarse el depurador **OllyDbg**, creado por Oleh Yuschuk. Puede descargarse en la web: `http://www.ollydbg.de/`.
- Versión: 1.06
- Ejecutable en todas las versiones de 32 y 64 bits de Windows desde XP hasta Windows 8.

1. Ejercicios

Para la correcta finalización de la práctica, debería haber leído y comprendido previamente los **Capítulos 3 y 4** de la documentación de x86. Esta práctica consiste en construir una calculadora con las operaciones aritméticas *sumar*, *restar*, *multiplicar* y *dividir* (estas dos últimas para números con signo). Nuestra calculadora pedirá por la entrada estándar los dos operandos y el operador y, según el operando seleccionado, llamará a un subprograma que será el encargado de realizar la operación. Deberemos crear por lo tanto una estructura similar a la estructura "*switch*"de C en el programa principal y cuatro subprogramas que realicen las operaciones pedidas. Abra **Problema1.ecp**. Su misión consiste en:

1. Utilizando el **API de C**, pida por la entrada estándar y almacene en memoria, *operando1*, *operador* y *operando2* (deberá crear en memoria dichas posiciones).
2. Cree una estructura *switch* que, según el operador introducido, llame al correspondiente subprograma. Pista: Piense cómo funciona internamente una estructura *switch* en C.
3. Cree un subprograma llamado "*suma*"que reciba los dos operandos y devuelva la suma total en **EAX**.
4. Cree un subprograma llamado "*resta*"que reciba los dos operandos y devuelva la diferencia en **EAX**.
5. Cree un subprograma llamado "*mult*"que reciba los dos operandos y devuelva el producto en **EAX**.
6. Cree un subprograma llamado "*division*"que reciba los dos operandos y devuelva el cociente en **EAX** y el resto en **EDX**. Tenga cuidado, pues deberá controlar que no se realicen divisiones cuyo divisor sea 0. En ese caso mostrará por pantalla el mensaje "División por cero".
7. Muestre por la salida estándar el resultado de la operación.