

x86: Segunda Práctica

Estructuras de control

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 realización de esta práctica satisfactoriamente, debería haber leído y comprendido al menos hasta el **Capítulo 3** de la documentación.

Esta práctica está compuesta por un único ejercicio que consistirá en crear un reproductor de partituras muy básico. La idea consiste en que dados un vector de enteros que representan las notas de la partitura y un vector de enteros que represente para cada nota del vector anterior su longitud expresada en milisegundos, podamos reproducir dicha partitura. En el vector de notas, consideraremos la siguiente representación:

- Valores negativos - Fin de la partitura.
- 0 - Silencio.
- 1 - Do.
- 2 - Re.
- 3 - Mi.
- 4 - Fa.
- 5 - Sol.
- 6 - La.
- 7 - Si.
- 8 - Do alto.
- 9 - Re alto.
- 10 - Mi alto.
- 11 - Fa alto.

Existe en el **API de Win32** una función llamada **beep** que recibe, en este orden, la *frecuencia* y la *duración* de un tono y lo reproduce usando los altavoces del ordenador o el altavoz interno (según si tenemos o no tarjeta de audio). Ambos parámetros tienen un tamaño **Doubleword**. Abra **Problema1.ecp**. Encontrará en la sección de datos tres vectores:

- El vector «frecuencias»: que tiene en cada posición la frecuencia asociada a cada nota (siguiendo la notación anteriormente descrita, frecuencia[0] es la frecuencia del silencio, frecuencia[1] es la frecuencia de Do, frecuencia[2] es la frecuencia de Re, etc.)
- El vector «notas»: que contiene la sucesión de notas de una melodía que, si consigue reproducir, seguro que le sonará.
- El vector «tempos»: de igual longitud que el anterior y que tiene en cada posición el tiempo asociado a cada nota del vector notas.

Entonces, el programa recorrerá el vector notas y en cada iteración llamará a la función **beep** con los argumentos frecuencia[nota_actual] y tempos[nota_actual]. Por ejemplo, en la primera iteración, deberemos invocar **beep** (frecuencias[notas[0]], tempos[0]), en la siguiente **beep** (frecuencias[notas[1]], tempos[1]) y así hasta llegar al final de la partitura, que representamos con un valor negativo en el vector «notas». Su misión consiste en:

1. Completar el bucle que llamará en cada iteración a **beep** pasándole los parámetros en el **orden y tamaño adecuado**. El bucle deberá comprobar si el valor del siguiente elemento del vector «notas» es negativo (en cuyo caso el bucle terminará pues habremos llegado al fin de la partitura) o bien si el valor es positivo o 0 (en cuyo caso el programa realizará una iteración más). Tenga en cuenta que los vectores notas y tempos **son de tamaño Word (2 bytes)**. Para invocar a la función **beep**, utilice las siguientes líneas de código:

```
1 pusha
2 invoke beep, frecuencia, duracion
3 popa
```

El funcionamiento de estas instrucciones se estudiará en la siguiente práctica, pero si siente curiosidad por ellas puede leer el **Capítulo 4** de la documentación.

Una vez terminados los ejercicios, experimente libremente, si es su deseo, con el programa que ha creado. Puede cambiar las notas de la partitura (cambiando el vector «notas»), la duración de las mismas (cambiando el vector «tempos») e incluso añadir más notas, ya sean más agudas, más graves o incluso añadir las notas sostenidas/bemoles (deberá hallar la frecuencia de dichas notas y cambiar la representación de las mismas).