

1º Curso del Grado en Ing. de Tecnologías de Telecomunicación Fundamentos de Computadores – Curso 2010/11 Práctica 1

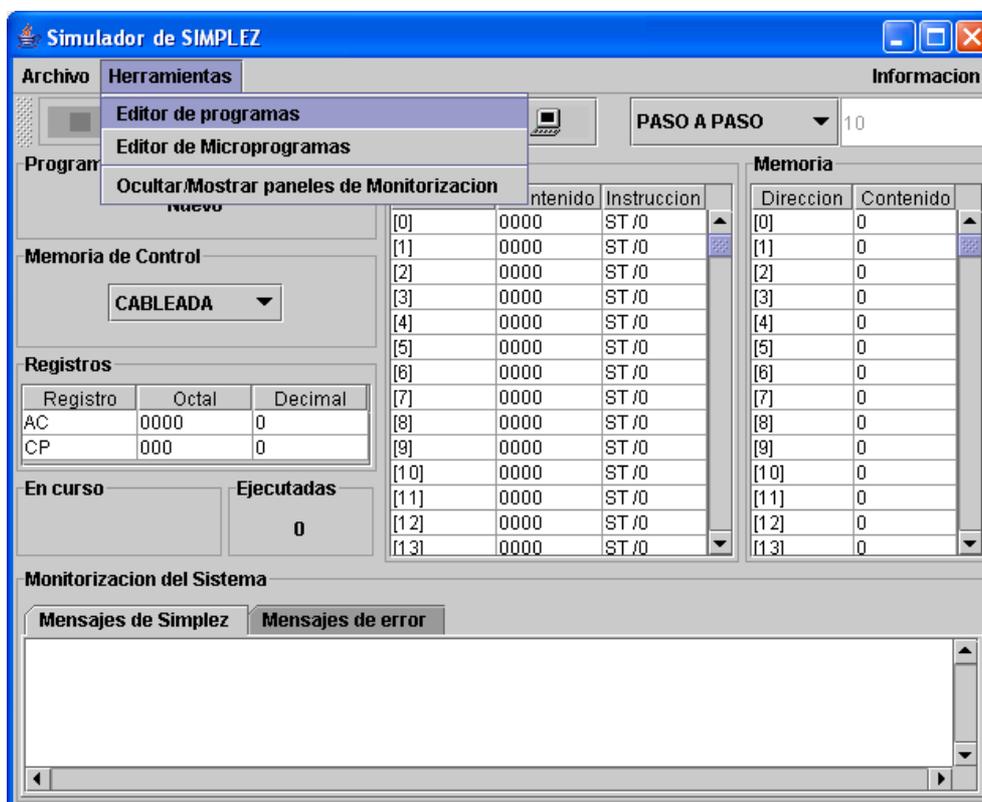
Objetivo:

Familiarizarse con un simulador de Símplez, de forma que se puedan codificar programas y ver el resultado de su ejecución en el ordenador ficticio Símplez. El entorno de simulación empleado ha sido creado por Iván Peña y Francisco Javier Rodríguez, y puede encontrarse originalmente en la web de la Escuela Superior de Ingenieros de Telecomunicaciones de la Universidad Politécnica de Madrid. Para ello se va a editar un programa que permite multiplicar dos números y se va a analizar la ejecución del mismo.

Contenido:

A) Edición / ensamblado / montaje de un programa en lenguaje en Ensamblador.

1. Abrir el entorno de simulación, ejecutando el fichero *Simplez.bat*, y seleccionar el menú *Herramientas->Editor de programas*.



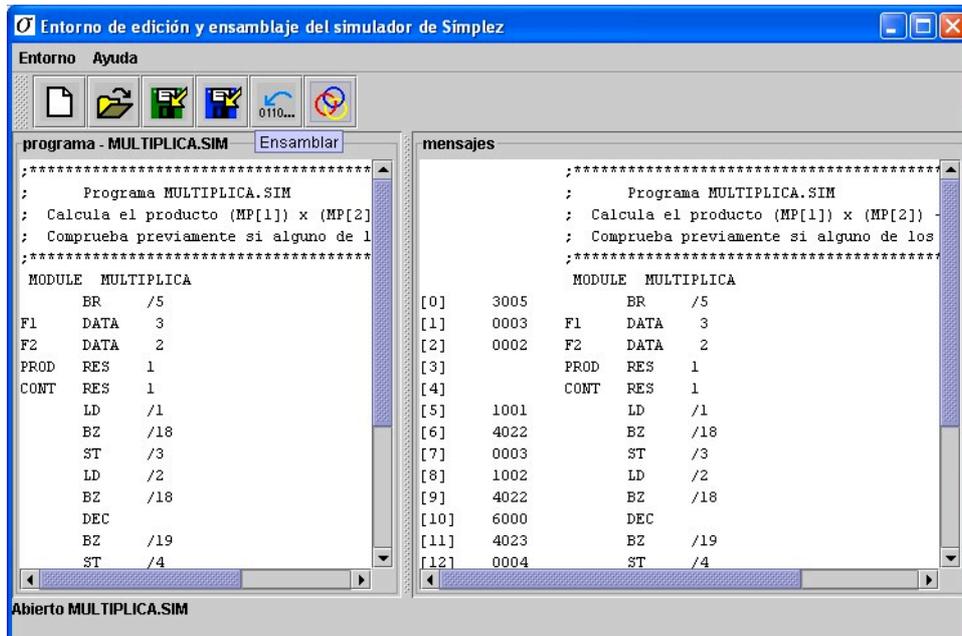
2. Editar el programa que se indica a continuación en la ventana de *programa* y guardar el programa editado con el nombre MULTIPLICA.SIM.

```

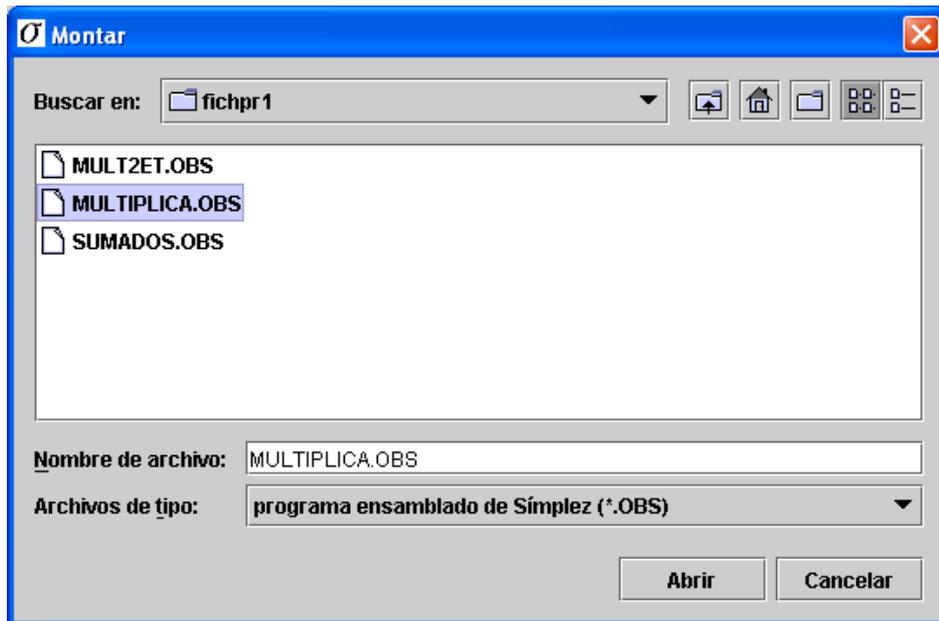
;*****
;
; Programa MULTIPLICA.SIM
; Calcula el producto (MP[1]) x (MP[2]) -> MP[3]
; Comprueba previamente si alguno de los factores es cero
;*****
MODULE MULTIPLICA

BR /5
F1 DATA 3
F2 DATA 2
PROD RES 1
CONT RES 1
LD /1
BZ /18
ST /3
LD /2
BZ /18
DEC
BZ /19
ST /4
LD /3
ADD /1
ST /3
LD /4
BR /10
ST /3
HALT
END
    
```

3. Ensamblar el programa editado, pulsando el botón correspondiente. En la ventana de *mensajes* del entorno de edición aparecerá el código binario (en octal) junto a la línea de lenguaje ensamblador que lo ha generado. También aparecerían mensajes de error si los hubiera. Al realizar el ensamblado, se generan dos ficheros con el mismo nombre pero con distinta extensión: MULTIPLICA.ENS y MULTIPLICA.OBS. Éste último contiene el código objeto del programa.



4. Por último, es necesario montar el programa seleccionando el botón correspondiente y el fichero MULTIPLICA.OBS. Se genera el fichero MULTIPLICA.LNS, que será utilizado posteriormente en el entorno de ejecución.



5. Cerrar el entorno de Edición de programas.

B) Ejecución de un programa en el simulador de Símplez.

1. El programa que se ha propuesto toma dos números almacenados en las posiciones 1 y 2 de memoria (el 3 y el 2 respectivamente) y los multiplica, almacenando el resultado en la posición 3 de memoria. Para ello va sumando el primero de ellos tantas veces como indique el segundo. La posición de memoria 4 se utiliza como contador para indicar el número de sumas a realizar.

Para comenzar la ejecución del programa primero se carga en memoria, mediante la opción *Archivo->Abrir fichero de código*, y seleccionado el fichero MULTIPLICA.LNS.



2. Ejecutar el programa Paso a paso, viendo como se va modificando el contenido de las posiciones de memoria 3 y 4, y como se abandona el programa cuando el número de iteraciones se ha alcanzado.

El contenido de la memoria del ordenador se muestra desde dos puntos de vista. En la ventana de *Memoria* se muestra la posición de memoria y su contenido en decimal. En la ventana *Programa* también se muestra el contenido de la memoria del computador, pero en formato octal en la columna *Contenido* y su interpretación en ensamblador en la columna *Instrucción*.

C) Ejercicios propuestos.

1. Modificar directamente el contenido de las posiciones de memoria 1 y 2 para que los números a multiplicar sean el 5 y el 7. Realizar esta ejecución de forma *Continua*, no *Paso a paso*. Para volver a ejecutar el programa sin necesidad de cargarlo de nuevo, sólo tiene que reiniciar el Contador de Programa poniendo un cero en el mismo.
2. Editar y ejecutar un programa que realice la resta de dos números situados en la posición 1 y 2 de memoria. La operación a realizar es, por tanto, $MP[1]-MP[2]$ y el resultado se almacena en la posición 3 de memoria. Puede suponer que el número $MP[1]$ es siempre mayor que $MP[2]$.
3. Editar y ejecutar un programa que compare dos números situados en las posiciones 1 y 2 de memoria. Si $MP[1]$ es mayor que $MP[2]$, en la posición 3 de memoria se debe guardar un 1. Si $MP[1]$ es menor que $MP[2]$, en la posición 3 de memoria se debe guardar un 2. Si ambos son iguales se debe almacenar un 0 en la posición 3.