

Desarrollo de un Entorno Integrado para un Computador Didáctico Elemental, para la Asignatura de Fundamentos de Informática del Nuevo Grado en Ingeniería de Tecnologías de Telecomunicación

H.Pomares, I.Rojas, A.Guillén, J.González, O.Valenzuela, J.P.Florido, J.Urquiza,
A.B.Cara, L.López-Mansilla, S.Egea-Navarro

Departamento de Arquitectura y Tecnología de Computadores. E.T.S. Ing. Informática y de
Telecomunicación, Universidad de Granada
Granada, España
hpomares@atc.ugr.es

Resumen. El objetivo principal de este trabajo es la potenciación y mejora de la metodología docente en la asignatura de Fundamentos de Informática, asignatura obligatoria correspondiente al nuevo Grado en Ingeniería de Tecnologías de Telecomunicación de la Universidad de Granada. Para ello, se ha desarrollado un entorno integrado didáctico que engloba, en una sola aplicación, tanto el simulador como el ensamblador de un Computador Didáctico Elemental desarrollado en nuestro departamento, CODE-2, y que, además, incorpora un programa editor y un ensamblador en línea con el que el alumno puede visualizar directamente el código máquina generado por el código ensamblador que está escribiendo en tiempo real. De esta forma, conseguimos facilitar que los alumnos puedan adquirir los conocimientos necesarios para comprender el funcionamiento interno de un computador.

Palabras Clave: Fundamentos de Informática; Ingeniería de Tecnologías de Telecomunicación; Computador Didáctico Elemental; CODE-2.

1 Introducción

En octubre de 2010 ha comenzado la implantación de los estudios del nuevo Grado en Ingeniería de Tecnologías de Telecomunicación por la Universidad de Granada. Como consecuencia de la visión de las necesidades del mercado laboral, y teniendo en cuenta el Libro Blanco de la titulación y los recursos humanos y materiales disponibles, se decidió proponer un único título de Graduado en Ingeniería de Tecnologías de Telecomunicación, con tres especialidades profesionales diferenciadas y elevado atractivo en el entorno socioeconómico, regional, nacional y europeo: 1. Sistemas de Telecomunicación, 2. Telemática, 3. Sistemas Electrónicos. Dentro de la materia básica *Fundamentos Tecnológicos y Empresariales*, común a todas las especialidades, se encuentra la asignatura *Fundamentos de Informática*.

Según la memoria de verificación de grado (disponible en <http://etsiit.ugr.es>), el objetivo principal de la asignatura es que el alumno adquiriera conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.

2 Temario de la asignatura

Teniendo en cuenta todo esto, los responsables de la asignatura han decidido el siguiente temario para la misma:

- Tema 1.- Introducción a la Informática: Conceptos básicos y definiciones. Soporte lógico y soporte físico. Tipos de ordenadores. Tipos de software: herramientas informáticas de aplicación en Ingeniería.
- Tema 2.- Representación de la Información: Codificación. Representación de texto, datos numéricos, sonido e imágenes.
- Tema 3.- Estructura funcional de los ordenadores: Niveles conceptuales de descripción de un computador. Esquema de funcionamiento de un computador. Estructuras básicas de interconexión.
- Tema 4.- Elementos de programación: Tipos de lenguajes. Elementos básicos de los lenguajes de programación. El proceso de traducción y ejecución de programas. Fundamentos de algorítmica.
- Tema 5.- Fundamentos de Sistemas Operativos: Definición. Gestión del procesador, de la memoria y E/S. Seguridad.
- Tema 6.- Bases de Datos: Conceptos básicos. Bases de Datos Relacionales. Diseño de Bases de Datos. Sistemas Gestores de Bases de Datos.

Igualmente, se han propuesto, como complemento docente, los siguientes seminarios:

- Seminario 1: Estructura y montaje de un PC.
- Seminario 2: Instalación de un Sistema Operativo.
- Seminario 3: Búsqueda de información en Internet y en Bibliotecas Digitales.
- Seminario 4: Software Libre y Software Propietario Sistemas operativos y software específico para servidores.

Finalmente, la parte práctica de la asignatura está compuesta por las siguientes cuatro prácticas:

- Práctica 1: Uso del Sistema Operativo.
- Práctica 2: Funcionamiento a bajo nivel de un ordenador.
- Práctica 3: Herramientas informáticas con aplicación en Ingeniería.
- Práctica 4: Uso básico de un Sistema Gestor de Bases de Datos.

En las siguientes secciones, se presenta el entorno integrado que se ha desarrollado para la segunda de estas prácticas: *Funcionamiento a bajo nivel de un ordenador*.

3 El Computador Didáctico Elemental CODE-2

CODE-2 es una máquina de tipo von Neumann que contiene todas las unidades típicas de este modelo: entradas, salidas, unidad de control, unidad de procesamiento y memoria de datos e instrucciones. CODE-2 es de 16 bits de longitud de palabra y tiene tan sólo 16 instrucciones máquina.

A través del lenguaje máquina de CODE-2 tenemos acceso a los siguientes elementos:

- Banco de registros: compuesto por 16 registros (r0 .. rF). Todos son de propósito general, aunque el registro rE se suele utilizar como puntero de pila y rD como registro de dirección.
- Unidad aritmético lógica (ALU): podremos hacer sumas y restas en C2, la operación NAND y desplazamientos.
- Bistables indicadores: cero (Z), signo (S), acarreo (C) y desbordamiento (V), según la operación realizada en la ALU.
- Memoria principal: compuesta por 64Kpalabras de 16 bits (128KB).
- Puertos de entrada: se pueden usar hasta 256 puertos de entrada (desde el IP00 a IPFF).
- Puertos de salida: se pueden usar hasta 256 puertos de salida (desde el OP00 a OPFF).

En la siguiente tabla se describe el repertorio completo de instrucciones de este computador didáctico elemental.

Nombre	Nemónico	Parámetros	Explicación
Cargar	LD	rx,[v]	$rx \leftarrow M(rD+v)$
Almacenar	ST	[v],rx	$M(rD+v) \leftarrow rx$
Carga inmediata baja	LLI	rx,v	$rx(15:8) \leftarrow H'00;$ $rx(7:0) \leftarrow v$
Carga inmediata alta	LHI	rx,v	$rx(15:8) \leftarrow v$
Entrada	IN	rx,IPv	$rx \leftarrow IPv$
Salida	OUT	OPv,rx	$OPv \leftarrow rx$
Suma	ADDS	rx,rs,ra	$rx \leftarrow rs+ra$
Resta	SUBS	rx,rs,ra	$rx \leftarrow rs-ra$
NAND	NAND	rx,rs,ra	$rx \leftarrow (rs \cdot ra)'$
Desplaz. izqda	SHL	rx	$C \leftarrow rx(15), rx(i) \leftarrow rx(i-1), i=15, \dots, 1; rx(0) \leftarrow 0$
Desplaz. dcha	SHR	rx	$C \leftarrow rx(0), rx(i) \leftarrow rx(i+1), i=0, \dots, 14;$ $rx(15) \leftarrow 0$
Desplaz. aritmétic. dcha	SHRA	rx	$C \leftarrow rx(0), rx(i) \leftarrow rx(i+1), i=0, \dots, 14$
Salto	B-	cnd	Si cnd se cumple, $PC \leftarrow rD$
Subrutina	CALL-	cnd	Si cnd se cumple, $rE \leftarrow rE-1, M(rE) \leftarrow PC,$ $PC \leftarrow rD$
Retorno	RET	-	$PC \leftarrow M(rE); rE \leftarrow rE+1$
Parar	HALT	-	Parar

4. Descripción del entorno desarrollado

Antes del desarrollo del proyecto actual, se disponía del siguiente conjunto de herramientas para CODE-2:

- Un simulador básico de CODE-2.
- Un programa ensamblador que traduce un programa escrito en lenguaje ensamblador de CODE-2 a código máquina.

Para mejorar el potencial pedagógico de estas herramientas, nuestro equipo ha desarrollado, a partir de estas dos aplicaciones independientes, un entorno integrado didáctico que engloba, en una sola aplicación tanto el simulador como el ensamblador y que, además, incorpora un programa editor y un ensamblador en línea con el que el alumno pueda visualizar directamente el código máquina que está generando el código ensamblador que está escribiendo en tiempo real. De esta forma, el alumno sólo tendrá que utilizar una única aplicación en sus prácticas de la asignaturas que hagan uso de dicho Computador Didáctico Elemental.

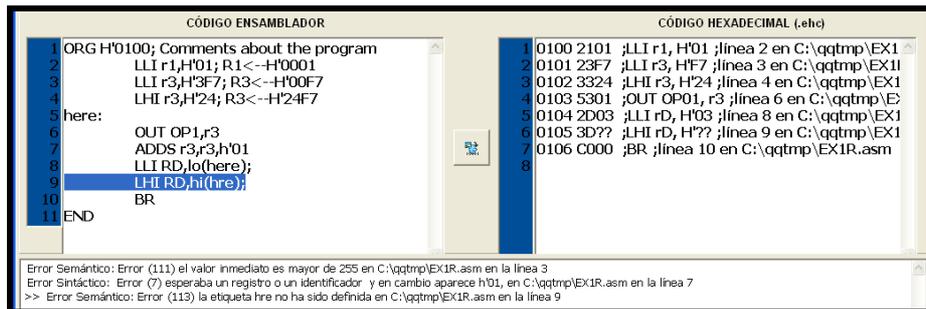


Figura 1. Ensamblador en línea dentro del entorno integrado desarrollado.

En la Figura 1 se muestra el aspecto del ensamblador en línea que se ha desarrollado como primer elemento del entorno integrado. Esta herramienta está compuesta por:

- a) Un editor (panel izquierdo) en el que el estudiante escribe sus programas mediante lenguaje ensamblador de CODE-2.
- b) Una ventana de código máquina (panel derecho) donde se traduce cada instrucción en lenguaje ensamblador a código máquina. Esta traducción se realiza en tiempo real cada vez que el alumno escribe una nueva instrucción. De esta forma, el estudiante puede comprobar cómo se va generando el programa final que debe cargarse en la memoria de CODE-2.
- c) Una ventana de mensajes de error (panel inferior). Conforme el alumno va escribiendo su programa, en esta ventana van apareciendo los diversos mensajes de error (si los hubiere) existentes en el mismo. Con un simple doble clic, el cursor se posiciona automáticamente en la línea de código que ha generado cada error.

Junto con el ensamblador en tiempo real, hemos desarrollado un multi-editor (Figura 2) con el que los estudiantes pueden abrir, modificar y guardar diversos ficheros tanto en formato ensamblador como en código máquina. El editor está equipado con las siguientes herramientas:

- Un conjunto de pestañas para poder navegar entre los diferentes ficheros abiertos.
- Una barra de estado con el nombre del fichero actualmente seleccionado, la línea y columna del cursor, y el número total de líneas del fichero.
- Cada fichero puede ser abierto, editado, imprimido y cerrado.
- Se pueden modificar las fuentes y su tamaño.
- Se puede copiar, cortar, pegar, seleccionar todo, deshacer, rehacer, buscar y reemplazar.
- Se pueden ensamblar cada fichero directamente desde el editor. En ese caso, aparece una ventana adicional (ver Figura 2) donde se indican los mensajes de error de igual forma que en el caso del ensamblador en línea.

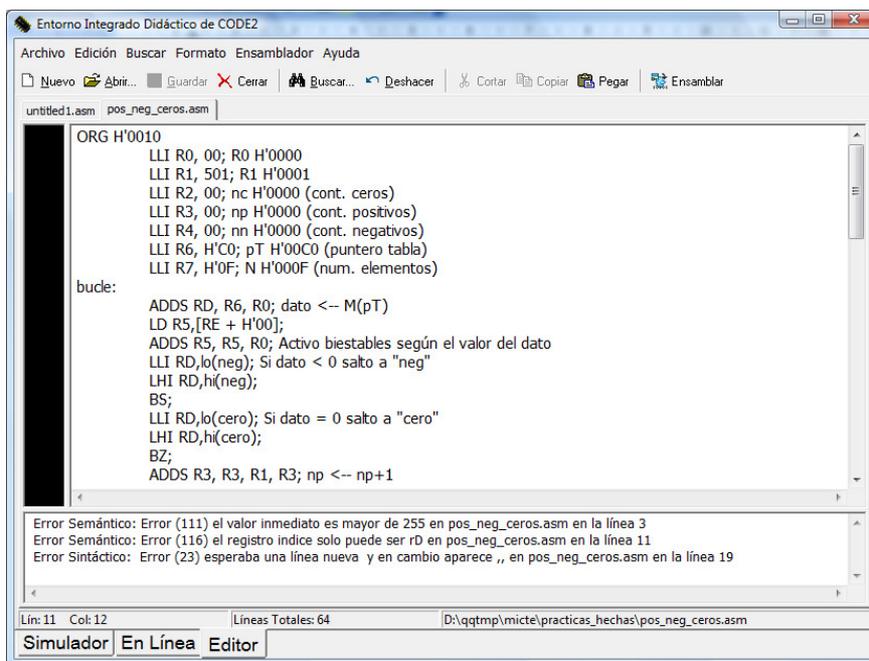


Figura 2. Multi-editor para la edición de programas para CODE-2

Finalmente, en la Figura 3 se muestra el aspecto del simulador de CODE-2 ya integrado en el entorno desarrollado. A través del simulador, los alumnos pueden cargar los programas ensamblados mediante cualesquiera de los editores anteriores y ejecutarlos paso a paso, comprobando en cada momento el efecto de cada instrucción ejecutada.

Por lo tanto, podemos resumir la labor realizada diciendo que hemos desarrollado una única herramienta didáctica que consta de:

- Un programa simulador completo de CODE-2, que permite visualizar fácilmente los contenidos de todos los registros, posiciones de memoria y puertos de E/S.
- Un editor de programas en ensamblador para CODE-2.
- Un ensamblador de CODE-2 integrado en la aplicación, de modo que se permite, con un simple clic del ratón editar la línea que ha generado cada uno de los códigos de error del ensamblador.
- Un programa intérprete que ensambla en tiempo real el programa que va escribiendo el alumno, de tal forma que le permite observar el código máquina generado cada vez que se escribe una línea de código.

De entre todas las características de la aplicación informática desarrollada, nos gustaría destacar su fiabilidad, robustez, facilidad de uso, y apariencia estética. Atributos que se han considerado indispensables para que el alumno pueda trabajar cómodamente con la herramienta, le sirva de ayuda, y no la deje de lado por su complejidad, por contener errores o cualquier otro motivo.

Finalmente, consideramos que la herramienta desarrollada constituye una mejora significativa en el desarrollo tanto de la parte teórica como de las prácticas de laboratorio, ya que con ella los alumnos podrán corroborar la correcta realización de los programas que deban escribir para dicho computador didáctico y, en caso de haber cometido un fallo, podrán de forma muy sencilla encontrar la fuente del mismo.

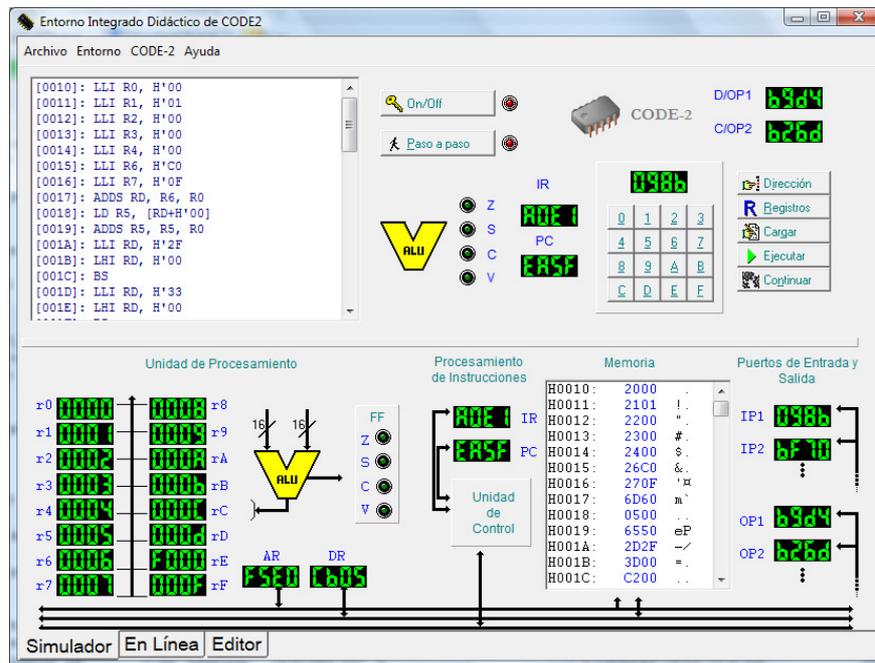


Figura 3. Simulador de CODE-2 dentro del entorno integrado desarrollado

4. Conclusiones

En este trabajo se ha presentado una herramienta para facilitar el aprendizaje del funcionamiento de un computador, dirigida a alumnos de la asignatura de Fundamentos de la Informática, asignatura obligatoria correspondiente al nuevo Grado en Ingeniería de Tecnologías de Telecomunicación de la Universidad de Granada. Dicha herramienta está compuesta por un ensamblador en línea para CODE-2, un multi-editor y un simulador de CODE-2, todo ello integrado en un mismo entorno. Esta herramienta está libremente disponible a través de Internet para todos los alumnos de la Universidad de Granada y de otras universidades. Consideramos que la herramienta diseñada constituye una mejora significativa en el desarrollo tanto de la parte teórica como de las prácticas de laboratorio, ya que con ella los alumnos podrán comprobar la correcta realización de los programas que deban escribir para dicho computador didáctico y, en caso de haber cometido un fallo, podrán de forma muy sencilla encontrar la fuente del mismo.

Agradecimientos. Este trabajo ha sido subvencionado parcialmente por el proyecto de innovación docente 10-22 de la Universidad de Granada.

Referencias

1. A. Prieto and A. Lloris, Introducción a la Informática, 3th edition, McGraw Hill, 2004.
2. J.Díaz, Prototipo hardware de CODE-2, proyecto fin de carrera, Universidad de Granada, 2002.
3. C.Hamacher, Z.Vranesic and S.Zaky, Organización de Computadores, McGraw Hill, 3th edition, 2000.
4. W.Stallings, Computer Organization & Architecture: Designing for Performance, Prentice-Hall, 6th edition, 2003.
5. H.Pomares et al, Novel Assembler to Facilitate the Understanding of How Processors Work, Current Developments in Technology-Assisted Education, Seville, Spain, 2006.

