

PRÁCTICAS DE ENSAMBLADOR

José Flich¹, Jorge Real¹, Juan Carlos Cano¹ y Julio Sahuquillo¹

¹*Departamento de Informática de Sistemas y Computadores
E. U. I.–Universidad Politécnica de Valencia–
e_mail: {jfflich, jorge, jucano, jsahuqui}@disca.upv.es*

RESUMEN: En la asignatura de microprocesadores y Periféricos (MYP) se imparten prácticas de ensamblador. Estas prácticas estaban diseñadas para que el alumno solamente aprendiese el lenguaje. Para ello el alumno implementaba programas bajo un entorno MS-DOS. A su vez, el alumno, en la parte teórica, aprendía conceptos de microprocesadores utilizándose el 8086 de Intel como referencia.

En el curso actual (99-00), se ha modificado el temario y las prácticas de la asignatura con el fin de motivar al alumno ofreciendo ejemplos prácticos en los que el lenguaje ensamblador es de utilidad. Además, la parte teórica se ha basado en el procesador Pentium.

En este artículo se describen los cambios realizados en la asignatura tanto en la parte teórica como en la práctica.

1.- INTRODUCCIÓN.

En la asignatura de Microprocesadores y Periféricos (MYP) de la Escuela Universitaria de Informática de la Universidad Politécnica de Valencia se han desarrollado en los últimos años prácticas del lenguaje ensamblador. Esta asignatura es optativa y se enmarca en el tercer curso de la titulación de Ingeniería Técnica en Informática de Gestión (ITIG).

En esta asignatura se imparte docencia de dos temas claramente diferenciados. En una primera parte se imparten conceptos de los microprocesadores mientras que en una segunda parte se imparten conceptos de periféricos del computador. Ambas partes tienen la misma duración de 7 semanas, tanto en las clases teóricas como en las clases prácticas. La asignatura tiene asignados 3 créditos de teoría y 3 de prácticas.

En los cursos anteriores, en la parte de microprocesadores se utilizaba como base el microprocesador 8086. Se profundizaba en conceptos de organización y arquitectura del juego de instrucciones del procesador 8086.

En las sesiones de prácticas correspondientes a la parte de microprocesadores, se profundizaba en el conocimiento del lenguaje ensamblador. Las prácticas ejercitaban la declaración de variables en ensamblador, los modos de direccionamiento, las instrucciones de salto, las instrucciones de cadena, etc. Estas prácticas se realizaban sobre el sistema operativo MS-DOS.

El código fuente era introducido por los alumnos con el editor *edit* de MS-DOS y se utilizaba el *TASM*, *TLINK* y *TD* de Borland para ensamblar, enlazar y depurar el código.

Las prácticas en ensamblador son importantes para el alumno, pues ayudan a comprender el funcionamiento interno del computador, así como a conocer mejor la arquitectura y organización del microprocesador en cuestión. Además, el lenguaje ensamblador es fundamental en el diseño de compiladores. Igualmente, el ensamblador es de gran valor en aquellas aplicaciones con requerimientos de tiempo críticos. En estas aplicaciones, el código implementado en ensamblador puede facilitar el cumplimiento de requisitos temporales de la aplicación. Ejemplos de tales aplicaciones son los de toma de datos, captura y procesamiento de imágenes, control de procesos, etc. Por último, el conocimiento del lenguaje ensamblador es un buen medio para conocer mejor el funcionamiento de un lenguaje de alto nivel.

No obstante, para el alumno, el aprendizaje de dicho lenguaje siempre supone un pequeño rechazo, pues piensa que el lenguaje ensamblador no es útil en la actualidad, ya que casi cualquier programa puede ser implementado íntegramente con un lenguaje de alto nivel. Además, el alumno estaba doblemente desmotivado por el entorno de trabajo utilizado en las prácticas, ya que se trabajaba con el sistema operativo MS-DOS, estando más familiarizado con sistemas operativos con un entorno gráfico.

En el curso actual (99-00), se ha cambiado la orientación de la docencia (teoría y prácticas) de la parte de microprocesadores de la asignatura MYP para darle un enfoque más atractivo y actual al alumno. En primer lugar, se ha optado por el cambio de la materia impartida dándole un enfoque más novedoso. Por ello, se ha modificado el temario de la asignatura utilizando el microprocesador Pentium como ejemplo. En lo referente a las prácticas, se han desarrollado nuevas prácticas que motiven más al alumno y que de verdad muestren los beneficios del lenguaje ensamblador al mismo tiempo de que sirvan para que el alumno aprenda el lenguaje ensamblador. Para ello, las prácticas se han desarrollado bajo el entorno MS-Windows (el alumno trabaja con aplicaciones Windows desarrolladas con el lenguaje ensamblador) y se han diseñado para que demuestren tales ventajas.

En el presente artículo, se detalla la organización del nuevo material tanto teórico como práctico de la parte de microprocesadores, haciendo hincapié en la experiencia personal de tales cambios.

2. TEMARIO DE LA ASIGNATURA Y BIBLIOGRAFÍA.

En la Tabla 1 se muestra el temario actual de la asignatura y la duración de cada tema. Los tres primeros temas corresponden a microprocesadores.

Tema	Título	Semanas
1	Introducción a los Microprocesadores	2
2	Arquitectura del Juego de Instrucciones del Pentium	3
3	Estructura/Organización del Pentium	2
4	Dispositivos de Entrada	2
5	Interfaces Paralelo y Serie	3
6	Dispositivos de Salida	2
-	Presentación de trabajos	1

Tabla 1 Temario de la asignatura MYP.

En un primer tema se introducen los microprocesadores y se describen de forma general las características comunes y particulares de los microprocesadores actuales. Se tratan temas como unidades funcionales, segmentación, prebúsqueda de instrucciones, ejecución dinámica, etc. También se describe la arquitectura del juego de instrucciones típica en los microprocesadores actuales. En este primer tema se ofrece un enfoque global, para que el alumno tenga conciencia de que además del microprocesador que se detalla en los siguientes temas (Pentium) también existe una gran variedad de microprocesadores.

En los temas 2 y 3 se describe con todo detalle un ejemplo práctico: el Pentium. Se describe tanto la arquitectura de su juego de instrucciones como su organización. Hay que resaltar que estos dos temas están diseñados para que el alumno conozca la evolución de los microprocesadores 80x86 de la familia Intel. Para ello, los temas hacen hincapié en cada apartado en su ámbito de aplicación (en qué microprocesador está implementado).

En estos temas se describen conceptos como: modos de direccionamiento, modos de operación, registros y juego de instrucciones. En la parte de ensamblador también se describen: etiquetas y tipos de datos, operadores, atributos, definición de segmentos, etc. Los conceptos más avanzados de direccionamiento a memoria (modos de operación) se describen de forma más generalista. La bibliografía asociada a estos temas se muestra en la Tabla 2.

Tema	Bibliografía
Tema 1	<i>Advanced Microprocessors</i> , Daniel Tabak. McGraw-Hill. 1991. Computer Organization And Architecture, William Stallings. Prentice Hall
Tema 2	The Art of Assembly Language Programming, Randall Hyde, 1996. Manual del Turbo Assembler, Borland
Tema 3	<i>Pentium System Architecture</i> . Tom Shanley. Pc System Architecture Series. Ed. Addison Wesley. (2nd edition). 1998 <i>Pentium Pro and Pentium II System Architecture</i> . Tom Shanley. Pc System Architecture Series. Ed. Addison Wesley. (2nd edition). 1997.

Tabla 2 Bibliografía Asociada

3. PRÁCTICAS DE LABORATORIO.

a) Entorno de trabajo.

Las prácticas de laboratorio se han desarrollado en un entorno gráfico (MS-Windows). El alumno utiliza/modifica el código en ensamblador de una aplicación Windows, sirviéndose para ello del entorno de desarrollo integrado de Borland C++ 4.52. Todo el código a modificar y/o examinar en las prácticas está integrado en un proyecto con diferentes módulos. Para realizar una buena depuración de la aplicación, el alumno utiliza el TurboDebugger (TD) de Borland, integrado en el propio BC++ 4.52. En la Ilustración 1 podemos ver el entorno de trabajo con el Borland C++ 4.52 y el proyecto de la aplicación.

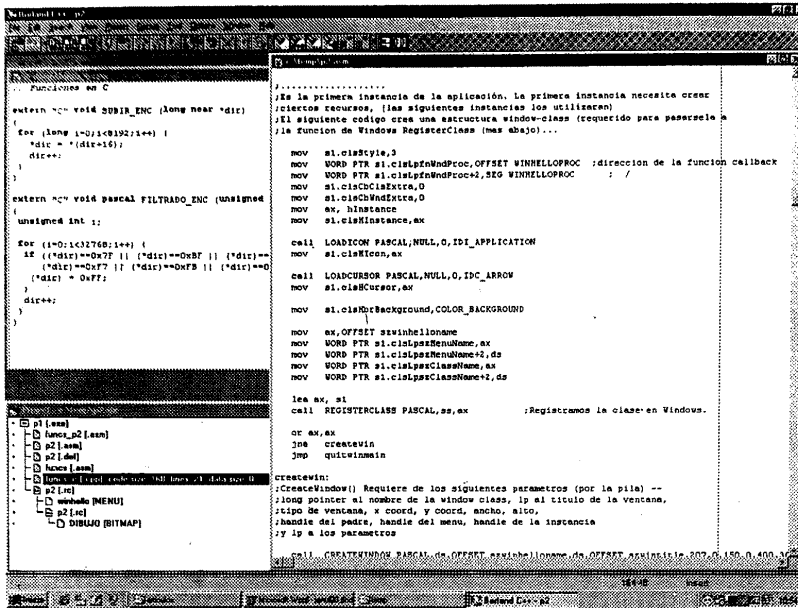


Ilustración 1 Entorno de trabajo y proyecto de la aplicación.

La estructura principal de la aplicación se facilita al alumno, dejando ciertos procedimientos para que el alumno codifique el código requerido en cada práctica.

b) Las prácticas de ensamblador.

Las prácticas de ensamblador son las siguientes:

Práctica	Semanas
Introducción a una aplicación Windows en ensamblador	2
Modos de direccionamiento	1
Instrucciones de cadena	1
Instrucciones varias	2
Recuperación	1

Tabla 2 Prácticas de microprocesadores (1,5 créditos).

En la primera práctica el alumno se familiariza con el entorno de trabajo y con la aplicación Windows en ensamblador (que será la aplicación que utilizará en el resto de prácticas). Para

ello, el alumno repasa todo el código de la aplicación. Dicho código está agrupado y comentado por bloques.

A su vez, el alumno empieza a relacionarse con los proyectos del Borland C++ 4.52 y aún más importante, se relaciona con el Turbo Debugger de Borland. Para ello, se proponen ejercicios de ejecución paso a paso, viendo cómo cambia el estado del procesador (registros, bits de estado, variables, etc.). Además también se ejercita en la búsqueda automática de código (prólogo, epílogo, llamadas a procedimientos) con el Turbo Debugger. Por último, y debido a la naturaleza del API de Windows, aprenden en la práctica el mecanismo de paso de parámetros y variables locales utilizando el convenio PASCAL. Debido a la relativa complejidad del código ensamblador y a que es la primera toma de contacto, esta práctica se realiza en dos sesiones (4 horas en total). En la

Ilustración 1 Entorno de trabajo y proyecto de la aplicación.

se puede ver un fragmento del código de la aplicación.

En la segunda práctica se hace hincapié en los modos de direccionamiento del procesador. Esta práctica es imprescindible debido a la dificultad que los alumnos tienen en ejercitar y entender los modos de direccionamiento de los procesadores Intel. El buen conocimiento de los modos de direccionamiento es imprescindible para diseñar eficientemente código en ensamblador. Por lo tanto, esta práctica va a servir para que en las siguientes prácticas el alumno pueda utilizar los modos de direccionamiento de forma correcta. En la práctica acceden a estructuras de datos anidadas (vectores de registros, registros anidados, etc) utilizando diversos modos de direccionamiento.

En la práctica 3 ya se ejercita con un ejemplo concreto de mejora de prestaciones utilizando el lenguaje ensamblador. En concreto se pide realizar diferentes operaciones sobre una imagen en blanco y negro de 512x512 puntos. Se les facilita una función en C que desplaza una fila hacia arriba toda la imagen (*scroll*). Con la instrucción *cpuid* del juego de instrucciones del Pentium el alumno obtiene en ciclos de procesador el tiempo de ejecución de la función en C, para posteriormente implementar la función equivalente en lenguaje ensamblador. Esto lo realiza primeramente con instrucciones de transferencia (*mov*) con diferentes tamaños de operandos (*byte*, *word* y *double word*) y tomando los tiempos de ejecución con la instrucción *cpuid*. El alumno, también implementa la función con instrucciones de cadena y prefijos de repetición, tomando de nuevo tiempos, y viendo las mejoras que se obtienen en cada caso.

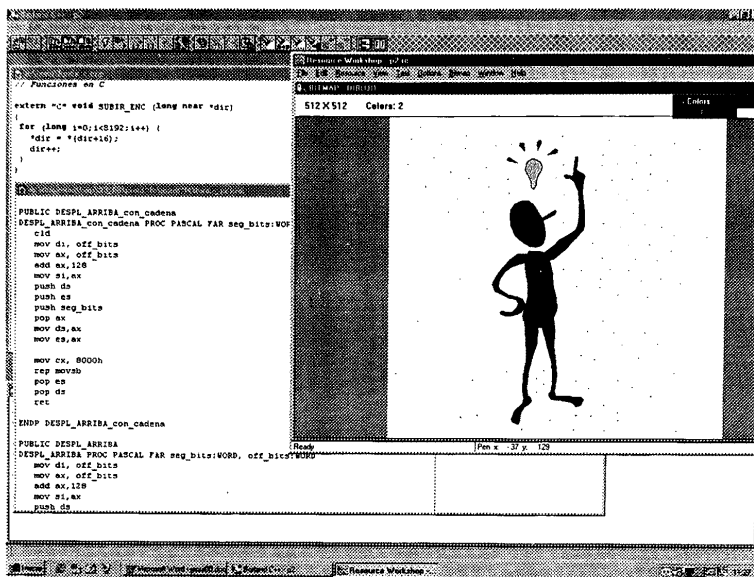


Ilustración 2 Funciones de scroll e imagen tratada.

En la misma práctica, el alumno debe identificar por qué la función en C resulta más lenta. Esto lo hace viendo directamente el código en ensamblador generado a partir de la función en C. El alumno identifica posibles deficiencias del código. En la Ilustración 2 vemos la imagen tratada y el código en C y ensamblador que implementan la función de *scroll*.

En la misma práctica se motiva al alumno para que ejercite otras instrucciones de cadena (*stos* y *lods*) y que identifique posibles pérdidas de prestaciones al utilizar estas instrucciones sin prefijos de repetición. Esto se realiza implementando una función de filtrado de la imagen (*xor* a cada byte del mapa de bits).

Por último en la práctica 4 se conciencia al alumno de que un código en ensamblador mal diseñado puede repercutir en un mayor tiempo de ejecución con respecto a un código equivalente en C. Para ello, el alumno debe realizar un filtrado a la imagen eliminando los *pixels* que aparezcan aislados en un byte del mapa de bits (un byte posee 8 *pixels*). El alumno incurre en la implementación del algoritmo con dos bucles anidados (el más externo para tomar bytes del mapa de bits y el más interno para contar el número de *pixels* activos). El alumno aprende a mejorar su código hasta que su versión mejora el tiempo de ejecución del equivalente en C.

Para la realización de estas prácticas en MS-Windows se ha utilizado la siguiente referencia bibliográfica:

Windows Assembly Language and Systems Programming,
Barry Kauler, R&D Books.

4. CONCLUSIONES.

Como conclusiones, la parte teórica y práctica de microprocesadores de la asignatura MYP se ha actualizado. Se imparte el tema de los microprocesadores utilizando un microprocesador actual como es el Pentium, cuando antes se impartían conceptos del 8086. El resultado ha sido una mayor motivación en los alumnos en las clases de teoría, debido a que se estudia un microprocesador actual.

Las prácticas se han actualizado a un entorno gráfico y se han diseñado nuevas prácticas que muestren los beneficios del lenguaje ensamblador.

Por tanto, utilizar un entorno más atractivo y un procesador más actual, ha ayudado a estimular al alumno en el estudio de un lenguaje tan importante como el ensamblador.