Entorno Integral de desarrollo para lenguaje en ensamblador basado en los servicios de Linux

Omar Mar Cornelio
Dpto. de Programación, Facultad 6, Universidad de las Ciencias Informáticas, Carretera a
San Antonio de los Baños, km 2 ½, Boyeros, La Habana, Cuba.
omarmar@uci.cu

Yadián Pérez Fernández
Dpto. de Programación, Facultad 6, Universidad de las Ciencias Informáticas, Carretera a
San Antonio de los Baños, km 2 ½, Boyeros, La Habana, Cuba.
yfernandezp@uci.cu

Juan Carlos Fiorenzano González
Dpto. de Programación, Facultad 6, Universidad de las Ciencias Informáticas, Carretera a
San Antonio de los Baños, km 2 ½, Boyeros, La Habana, Cuba.
jcfiorenzano@uci.cu

Darwis Rodríguez Licea
Dpto. de Programación, Facultad 6, Universidad de las Ciencias Informáticas, Carretera a
San Antonio de los Baños, km 2 ½, Boyeros, La Habana, Cuba.
drlicea@uci.cu

RESUMEN

Producto a la estrategia de migración de la Universidad de las Ciencias Informática UCI, se rediseña la asignatura de Arquitectura de Computadoras, para la cual no se contaba con una herramienta que permitiera la codificación de programas ensambladores sobre plataformas libres. El presente trabajo describe la solución a dicha problemática codificando de un compilado como Entorno Integral de Desarrollo (IDE) donde se utilizó Java como lenguaje de programación, Fireworks CS4 para la creación de imágenes e iconos, NetBeans como entorno de Desarrollo. El sistema cuenta con varias herramientas que facilitan la reutilización de código y la detección de errores lo que permite agilizar el proceso a los desarrolladores.

Palabras claves: Codificación, entorno de desarrollo integrado, lenguaje ensamblador, sistema.

INTRODUCCIÓN

Los lenguajes ensambladores fueron desarrollados en los años 1960. Su importancia radica principalmente que se trabaja directamente con el microprocesador; Dándole al

programador la capacidad de realizar tareas muy técnicas que serían difíciles de implementar en un lenguaje de alto nivel. (1)

De ahí que la carrera de informática en las universidades cubanas asuma la enseñanza del lenguaje ensamblador. La Universidad de las Ciencias Informáticas (UCI) la implantó desde el año 2003 mediante la asignatura de Máquinas Computadoras, enfocada a la implementación de aplicaciones que hagan uso de los servicios del BIOS y el DOS. En septiembre del 2011 como parte de la estrategia de migración asume un nuevo programa para dicha asignatura rediseñándola con el nombre de Arquitectura enfocada a los servicios de sistema operativo Linux. (2)

Actualmente en la universidad las clases prácticas de Arquitectura de Computadoras se imparten apoyándose en herramientas de trabajo que de cierta manera atentan con la buena asimilación del contenido por parte de los estudiantes. Por lo general, los proyectos se codifican en un editor de texto convencional dígase Leafpad o Gedit, por solo mencionar algunos ejemplos, mientras que el proceso de ensamblaje, enlazado y ejecución se realiza a través de la Terminal de Linux, ejecutando las herramientas Netwide Assembler (NASM) y GNU Compiler Collection (GCC). Después de analizar la situación se detectaron los siguientes inconvenientes:

Dentro de las dificultades detectadas a los estudiantes a la hora de trabajar conjuntamente con las herramientas antes mencionadas se encuentran: Insuficiencia para encontrar errores de código. El proceso de codificación es muy angustioso.Como se utiliza un editor de texto convencional, realizar los proyectos toman mucho tiempo y añade complicaciones que un entorno de desarrollo eliminaría.

En la universidad actualmente no existe una herramienta que elimine estas deficiencias, tras la migración. Como resultado del análisis de la situación se decide: Desarrollar un Entorno de Desarrollo Integrado para la codificación de programas en lenguaje Ensamblador que utilice los servicios de los sistemas operativos Linux.

MATERIALES Y MÉTODOS

El Entorno de Desarrollo Integral para la compilación de programas en ensamblador, en su versión 1.0 está orientado a soportar la implementación de aplicaciones en lenguaje ensamblador utilizando los servicios de Linux. El mismo permite la creación y modificación del código fuente, así como compilación, construcción y ejecución de programas.
El entorno está provisto de una serie de funcionalidades, entre ellas podemos encontrar las siguientes:

- Editor de código inteligente, con resaltado de sintaxis y completamiento de código.
- Biblioteca de macros y procedimientos personalizable, donde puede crear sus propias macros e insertarlas en el código.
- Capacidad para interactuar con más de un proyecto.
- Facilidad de manejo y una interfaz agradable e intuitiva para el usuario.

Para el desarrollo del IDE se utilizaron las siguientes herramientas y tecnologías:

- Como lenguaje de programación se utilizó Java, por ser un lenguaje simple, orientado a objetos, distribuido, interpretado, robusto, seguro, de arquitectura neutra, portable, de altas prestaciones, multitarea y dinámico.
- Como Entorno de Desarrollo se escoge a NetBeans IDE 7.1, el cual ofrece mejoras en el constructor de interfaces Swing. Es el primer entorno de desarrollo Java con soporte para JavaFX 2.0 con lo que es posible llevar a cabo los pro-

cesos completos de compilación y depuración de las aplicaciones desarrolladas en esta plataforma.

- El generador de analizadores lexicográficos JFlex, el cual está desarrollado en Java. Permite actuar sobre aquellas cadenas de un fichero de texto que encajan en una expresión regular.(3)
- Para desarrollar la ayuda se utilizó JavaHelp, el cual es una expansión de Java que facilita la programación de las ventanas de ayuda.
- El Lenguaje Unificado de Modelado (UML) fue el lenguaje utilizado para modelar la solución en términos ingenieriles según la metodología propuesta. UML ha mejorado el desarrollo de software no solo al establecer un estándar común que simplifica la comunicación entre los desarrolladores de software. Sus principios fundamentales son fáciles de entender y de aprender. Hoy día, es el lenguaje que complementa la ingeniería de Software.
- Para el diseño de diagramas se utilizó VisualParadigm 8 para UML, pues es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software.
- Para el diseño de logos e imágenes se utilizó Fireworks CS4 pues es una aplicación enormemente accesible, con más potencia para crear elementos gráficos.

La estructura de codificación utilizada, es la programación modular. Permite una estrategia efectiva para resolver problemas complejos. Bajo “el paraguas” del programa principal, distintos subprogramas serán llamados para realizar su función en un orden establecido. Dentro de las características que posee el entorno organizado modularmente se destacan: (4), (5)

1. La existencia de una estructura superior que engloba a todos los módulos. Cada módulo contribuye a un fin más amplio mediante el cumplimiento de una tarea concreta.

2. La independencia de funcionamiento entre módulos, lo cual no significa que no haya comunicación entre ellos o con la estructura superior.

El hecho de que exista independencia de funcionamiento unido a que un módulo cumpla una tarea concreta, permiten su traslado o copia de organización en otra estructura, pero necesitada del proceso concreto que realiza el módulo. Ejemplo de ello es el editor de código este objeto es el centro de la edición del código fuente en la interfaz principal. Pero a su vez es necesario para las ventanas auxiliares, como la personalización de macros y procedimientos. Tras el análisis del caso, podría decidirse trasladar este objeto con todo lo que al respecta, resaltado de sintaxis, completamiento de código, analizador lexicográfico, sintáctico y semántico. El traslado o copia de un módulo requerirán normalmente una pequeña adaptación para encajar adecuadamente en la nueva estructura.

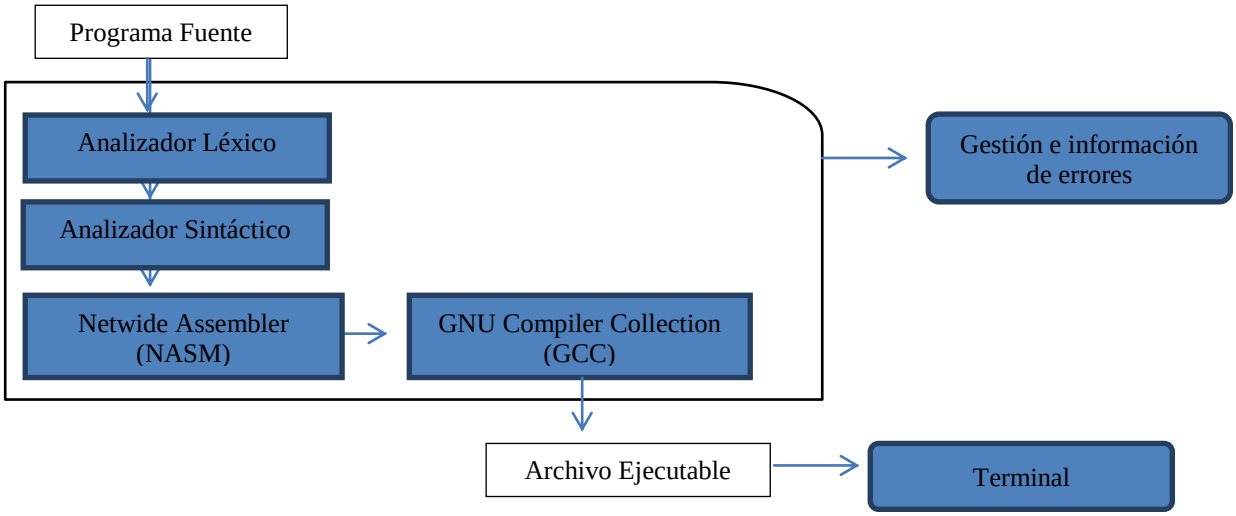
3. La complejidad del módulo es siempre inferior a la del conjunto de módulos. Esto es importante en relación a la estructura mental de los humanos y está relacionado con el “trabajo por objetivos”. Dividir un gran problema en pequeños problemas se comprueba que lleva a:

- Aumentar la probabilidad de éxito al concentrar la atención del programador y evitar su dispersión.
- Aumentar la tranquilidad y confianza del programador, que ante problemas de magnitud excesiva tenderá a sentirse abrumado.

La complejidad intrínseca de un problema será la misma independientemente de cómo se aborde. En cambio, la probabilidad de éxito para un humano varía sustancialmente según sea la estrategia de resolución empleada.

4. La división por módulos facilita la organización y comprensión desde el punto de vista humano. Con esta organización resulta más fácil detectar errores, corregirlos y tener una visión de conjunto de la estructura y funcionamiento de los programas.

Para garantizar la ejecución de los programas a codificar se hace llamado a la consola. La Figura 1 muestra un diagrama con el proceso de ejecución. El sistema reconoce la consola del sistema y a través de las clases brindadas por java *Runtime y Process* se puede controlar la misma, de esa forma se puede mandar a ejecutar el programa recién compilado. (6)



La Figura 1: Diagrama para el proceso de ejecución

La Figura 2 representa una vista de la estructura de la aplicación, con todos los paquetes, las relaciones de dependencias que se establecen entre los componentes para el correcto funcionamiento del entorno.

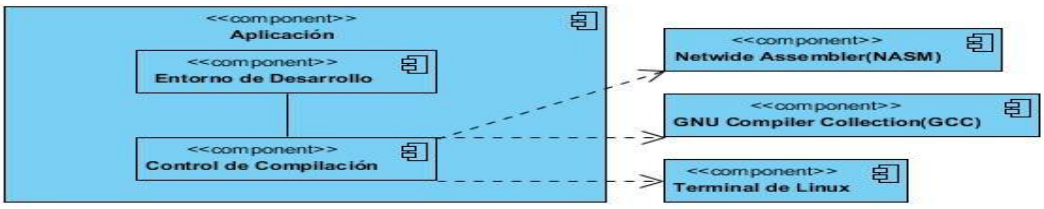


Figura.2: Diagrama de Componentes de la Aplicación

Dentro de los requerimientos de Hardware definidos para garantizar el correcto funcionamiento, es necesario como requisitos mínimos debe tener un microprocesador Pentium IV o superior, una memoria operativa de 256 Mb o superior.

RESULTADOS Y DISCUSIÓN

La Figura 3. Visualiza la ventana principal del Entorno de Desarrollo Integrado a través de la cual se puede acceder a las diferentes áreas y funcionalidades como son la barra de herramienta, explorador de proyecto, navegador de etiquetas, biblioteca de macros, panel de salida y el área de edición.

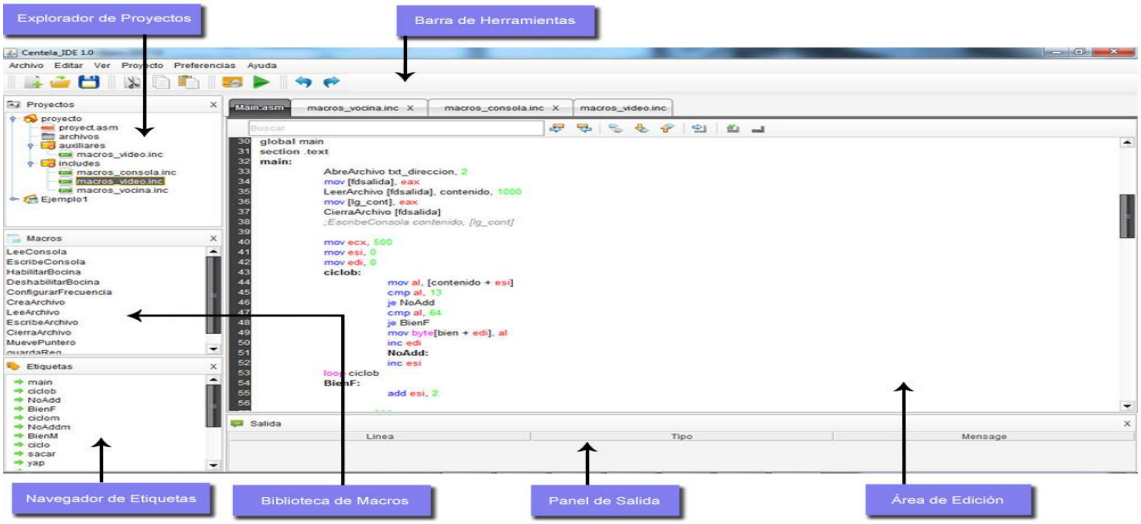


Figura.3: Ventana principal del Entorno de Desarrollo Integrado.

La Barra de Herramientas*:* contiene las herramientas estándares para abrir y guardar archivos, cortar, copiar y pegar. Además, ofrece las opciones de construcción y ejecución del proyecto, de vital importancia para el desarrollo y prueba del mismo.

El Explorador de Proyectos: muestra los proyectos en un árbol del proyecto. Contiene una lista de todos los proyectos abiertos en la sesión actual. Los archivos de cada proyecto se agrupan de acuerdo a su tipo de archivo representando un panel clave para navegar a través de los diversos proyectos en desarrollo.

El Área de Edición: Es donde se desplegarán las ventanas necesarias para editar el código fuente. Cuenta con un Editor de código inteligente, con resaltado de sintaxis y de errores en tiempo real, así como completamiento de código, esta utilidad le permite al usuario agilizar el proceso de codificación.

La Biblioteca de Macros: Proporciona una lista de macros personalizables, posibilitando la reutilización de código.

El Área de Navegador de Etiquetas: Proporciona una lista de las etiquetas contenidas en el código fuente del proyecto en edición.

El Panel de Salida: El Panel de Salida proporciona una lista de errores y avisos encontrados durante la compilación.

En la Figura 4. Muestra la ventana que nos permite crear un nuevo proyecto. En ella se evidencia la capacidad de poder incluir macros predeterminadas por el sistema, creando así un nuevo archivo llamado “macros.inc” que automáticamente se añade al proyecto, lo que facilita a los desarrolladores aumentar su eficiencia reutilizando rutinas previamente definidas.

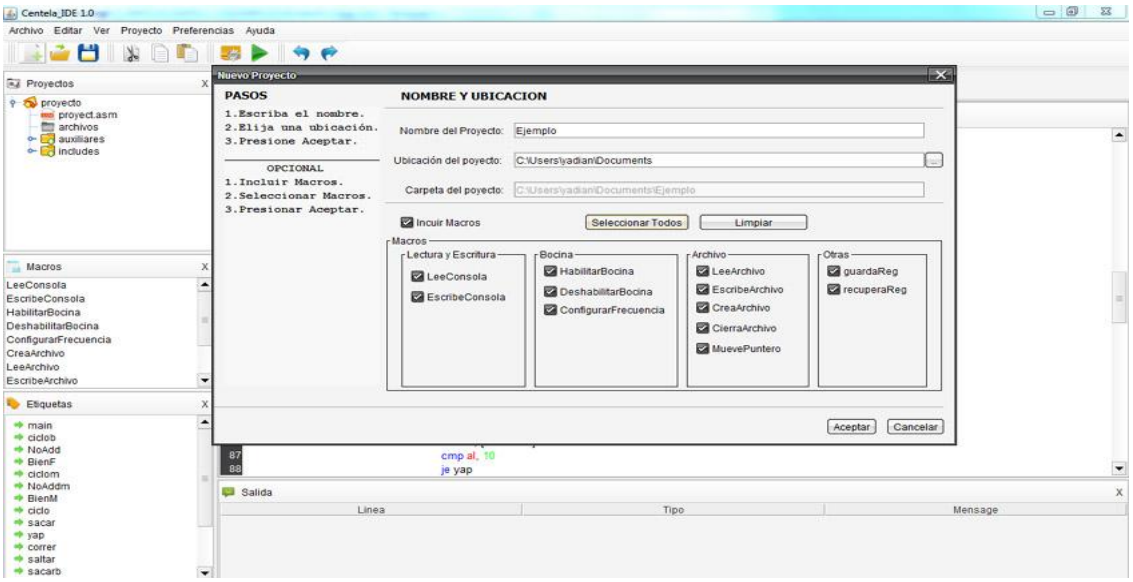


Figura.4: Menú creación de proyectos.

En la Figura 5. Muestra un proyecto compilado, de existir un error este sería mostrado en la sección de mensajes. En este caso, en la sección de mensajes se muestra "Construcción Correcta" ya que no se encontró ningún error, ejecutando así automáticamente el programa.

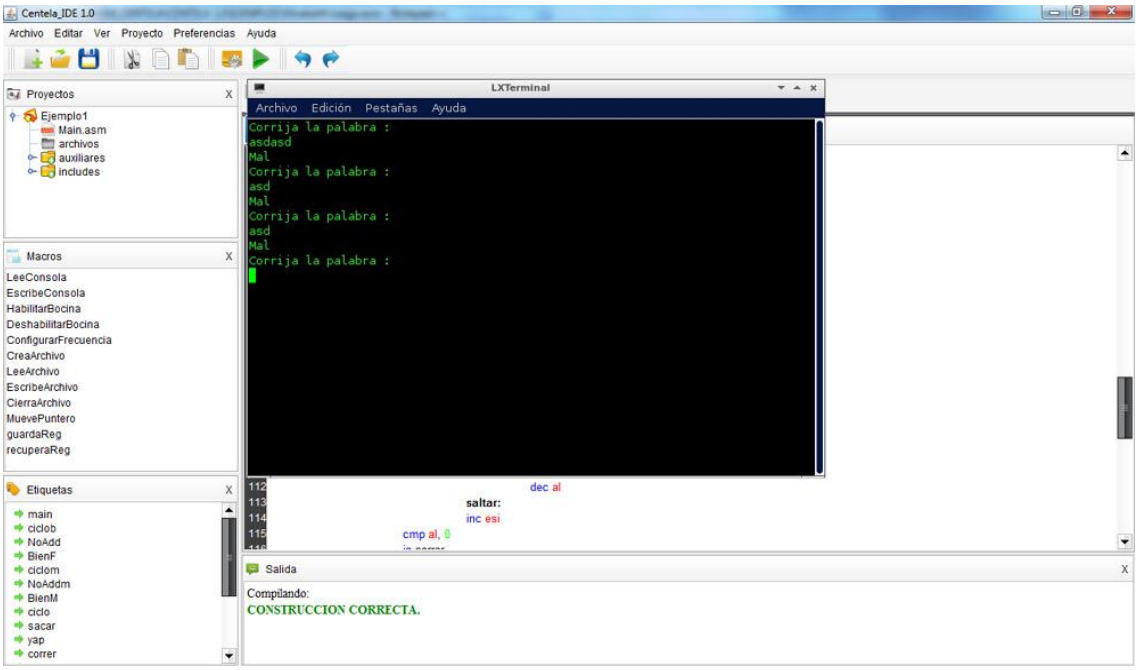


Figura.5: Menú compilar.

Aporte y novedad

La herramienta desarrollada, mantiene la apariencia de los demás entornos de desarrollos utilizados en la universidad, lo que garantiza que el usuario se sienta cómodo al programar.
Con la utilización del registro de macros es posible reutilizar los procedimientos reduciendo tiempo y esfuerzo.

Utilizando la herramienta propuesta en los laboratorios de Arquitectura de Computadoras, se evidencias una fácil asimilación por parte de los estudiantes ya que las clases se centran más en las dificultades de aprender un nuevo lenguaje y no se pierde tiempo explicando herramientas con bajas prestaciones y diseños diferentes.

CONCLUSIONES

Con la implementación del Entorno de Desarrollo Integral es posible la codificación de programas ensambladores que utilicen los servicios de Linux para la asignatura de Arquitectura de Computadoras en la Universidad de las Ciencias Informáticas.

Con la propuesta presentada es posible la reutilización de códigos, lo que determina una mayor velocidad y la facilidad de uso para los programadores no habituados con el lenguaje ensamblador.

Bibliografías

(1) Dart, S., Ellison, R."Software Development Environments". IEEE Computer, Vol.20 No.11 pp.18-28. *2008*

(2) Herrera, A. (2011) Programa analítico Arquitectura de Computadoras. Consultado 15 de noviembre, 2012, disponible http://eva.uci.cu/course/

(3) URQUIZA J."AnalizadorLexico-JFlex", Consultado 8 de octubre 2012, disponible en: http://www.escet.urjc.es/~procesal/transp/Tema2-AnalizadorLexico-JFlex.pdf. 2009

(4) Rancel Mario, "Didáctica y divulgación de la programación", Consultado 8 de octubre 2012, disponible en: http://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=113:introduccion-a-la-programacion-por-modulos-estrategia-de-divide-y-venceras-cu00203a&catid=36:curso-qbases-de-la-programacion-nivel-iiq&Itemid=60. 2012

(5) Holzner, Steven, "La Biblia del Java 2", Consultado 8 de octubre, 2012, Disponible en: http://www.bibliotheka.org/?/ver/39707. 2008

(6) Charte, Francisco, "Ensamblador para Dos, Linux y Windows", Madrid, Anaya Multimedia, 701 páginas. ISBN 9788441514829. 2009

Edita:

Director: José Ángel Ruiz Felipe
Jefe de publicaciones: Antero Soria Luján
D.L.: AB 293-2001
ISSN: 1578-326x