

Universidad de Costa Rica

Escuela de Ciencias de la Computación e Informática

CI-0118 Lenguaje Ensamblador

Tema: GUI Turbo Assembler versus WinAsm

Integrantes:

Madrigal Barrantes Braunny (C24436)

Picado Leiva Marcelo (C15950)

Rojas Carrillo Henoc (C26764)

Villafuerte Beita Kendall (C28387)

I Semestre

2024

Tabla de contenidos

Portada.....	p. 01
Tabla de contenidos.....	p. 02
Introducción.....	p. 03
Antecedentes.....	p. 03
Objetivos.....	p. 03
Alcances y limitaciones.....	p. 03
Marco teórico.....	p. 04
Metodología.....	p. 04
Tipo de investigación.....	p. 04
Enfoque de la investigación.....	p. 04
Técnicas de investigación.....	p. 04
Capítulo I: Similitudes entre ambos entornos de desarrollo integrado.....	p. 04
Capítulo II: Diferencias entre ambos entornos de desarrollo integrado.....	p. 05
Análisis de resultados.....	p. 06
Conclusiones.....	p. 08
Bibliografía.....	p. 09

Introducción

A la hora de desarrollar programas en lenguaje ensamblador, la elección del entorno de desarrollo integrado (IDE) tiene un impacto significativo en la eficiencia de la persona programador o de incluso un equipo de programadores. Dos de los IDEs más utilizados para este propósito son GUI Turbo Assembler (GUI TASM) y WinAsm, cada uno popular en su terreno; Turbo Assembler en el caso de GUI TASM y Microsoft Macro Assembler en el caso de WinAsm.

GUI TASM es conocido por su simplicidad y enfoque directo en la programación en ensamblador, ofreciendo herramientas básicas pero efectivas para la gestión y compilación de código. Por otro lado, WinAsm proporciona un conjunto más amplio de características.

Antecedentes

GUI Turbo Assembler es un IDE integral que incluye Borland Turbo Assembler (TASM), Turbo Linker (TLINK), Turbo Debugger (TD) y DOSBox. Este entorno soporta modos de 16 y 32 bits. Además, GUI TASM soporta la localización en siete idiomas y permite la creación de archivos localizados, con la opción de contribuir a su desarrollo en GitHub. El paquete incluye versiones específicas de TASM, TLINK y TD para ambos modos, y utiliza DOSBox para emular el modo de 16 bits en sistemas x64 (Nath, 2023).

Por otra parte, WinAsm es un IDE optimizado para el desarrollo de programas de 32 bits para Windows y 16 bits para DOS utilizando ensamblador. Originalmente creado como el hogar del IDE WinAsm y HiEditor (Kyprianou, s.f.).

Objetivos

Objetivo General:

- Analizar los entornos de desarrollo integrado GUI Turbo Assembler y WinAsm.

Objetivos Específicos:

- Destacar las semejanzas entre los entornos de desarrollo integrado propuestos.
- Listar las diferencias claves entre ambos entornos de desarrollo.

Alcances y limitaciones

El alcance de esta investigación se limita al análisis comparativo de las características de los entornos de desarrollo integrado GUI Turbo Assembler y WinAsm según la información suministrada por el equipo de desarrollo de cada uno. Se tiene como limitación principal no poder realizar un proyecto completo en ambos entornos; que pueda servir para una prueba de campo a los mismos.

Marco Teórico

Turbo Assembler

Es un ensamblador desarrollado por Borland y publicado en 1989. Diseñado para ejecutarse en entornos compatibles con MS-DOS, así como en Windows. TASM genera código para procesadores x86 de 16 o 32 bits. Este ensamblador se integra con productos como Turbo Pascal, Turbo Basic, Turbo C, Turbo C++, Turbo Linker y Turbo Debugger (Swan, 1989).

Microsoft Macro Assembler

Es un ensamblador desarrollado por Microsoft y publicado en 1981. Utiliza la sintaxis de Intel y está diseñado para MS-DOS y Microsoft Windows. MASM genera código para procesadores x86 de 16, 32 o 64 bits. Se suele incluir en varios SDK de Microsoft y compiladores C (Detmer, 2007).

Metodología

Tipo de Investigación

La investigación será de carácter descriptivo, utilizando enfoques cualitativos para examinar las características de los entornos de desarrollo integrado.

Enfoque de la Investigación

Se utilizarán métodos cualitativos para entender las descripciones técnicas de profesionales de informática.

Técnicas de Investigación

Se revisarán estudios previos, documentaciones técnicas y otros recursos relevantes. Además, se examinarán manuales y guías de instalación de ambos entornos de desarrollo integrado.

Capítulo I: Similitudes entre ambos entornos de desarrollo integrado

GUI Turbo Assembler y WinAsm comparten similitudes fundamentales que los posicionan como opciones robustas para desarrollo en lenguaje ensamblador. Ambos entornos permiten ajustar el tamaño de la letra, lo que mejora la personalización y la comodidad durante el desarrollo. Además, ofrecen ejemplos y plantillas de código predefinidos, acelerando la creación de proyectos al proporcionar estructuras y soluciones comunes.

Una característica destacada en ambos IDEs es su capacidad avanzada de búsqueda y reemplazo de texto a lo largo del código, facilitando la navegación y la edición eficiente de programas complejos. El menú de “View” en ambas plataformas permite a los usuarios personalizar la interfaz gráfica según sus necesidades, gestionando qué elementos como la barra de herramientas o la salida del programa se muestran o se ocultan con facilidad.

Además de su funcionalidad básica, GUI TASM y WinAsm ofrecen herramientas sólidas para la gestión de proyectos, incluyendo opciones intuitivas para la creación, guardado y manipulación de archivos. Esta capacidad promueve una organización eficiente del trabajo y facilita la colaboración en proyectos complejos. Ambos entornos son software gratuito y de código abierto, fomentando la accesibilidad y la participación activa de la comunidad de desarrolladores en su mejora continua.

Capítulo II: Diferencias entre ambos entornos de desarrollo integrado

GUI Turbo Assembler y WinAsm presentan diferencias significativas. GUI Turbo Assembler, por ejemplo, carece de la capacidad de modificar el tipo de letra dentro de su entorno, limitando las opciones de personalización visual en comparación con WinAsm, que permite ajustes en el tipo de letra para una mejor legibilidad. Además, WinAsm ofrece un tema oscuro y claro, proporcionando a los usuarios opciones estéticas que se adaptan a sus preferencias de visualización.

Una diferencia clave radica en la variedad de archivos que cada IDE puede manejar. GUI Turbo Assembler admite archivos .asm y .inc, mientras que WinAsm amplía su soporte incluyendo también archivos .rc (de recursos). Esta capacidad adicional en WinAsm es crucial

para proyectos que requieren la integración de recursos adicionales dentro del código ensamblador.

Otra distinción importante es la automatización de tareas en el proceso de desarrollo. GUI Turbo Assembler realiza el ensamblado y construcción del código automáticamente, simplificando el flujo de trabajo del programador. En contraste, WinAsm requiere que el usuario indique explícitamente el nombre y la ubicación para guardar el archivo ensamblado, ofreciendo un mayor control sobre el proceso de compilación.

Estas diferencias se extienden a las funcionalidades de edición y depuración. GUI Turbo Assembler carece de opciones integradas para indentar o comentar automáticamente el código, lo que requiere que el usuario realice estas tareas manualmente. En contraste, WinAsm proporciona herramientas para indentar, desindentar y comentar automáticamente el código seleccionado, mejorando la eficiencia y la legibilidad del programa. Además, mientras GUI Turbo Assembler tiene un depurador que se considera eficaz pero poco eficiente, WinAsm no cuenta con un depurador integrado de forma nativa, lo que puede afectar la capacidad del programador para encontrar y corregir errores de manera rápida y efectiva.

Análisis de Resultados

GUI Turbo Assembler y WinAsm
<ol style="list-style-type: none">1. Capacidad de aumentar o disminuir el tamaño de letra.2. Incluyen ejemplos o plantillas con código hecho previamente.3. Menú para buscar palabras a lo largo del código, con opción de reemplazar.4. Menú de “View” que permite seleccionar qué partes de la interfaz mostrar y qué partes ocultar, como la barra de herramientas o la salida.5. Opciones de creación y guardado de archivos/proyectos.6. Opciones de manejo de texto como copiar, cortar y pegar.7. Ambos manejan atajos de teclado para la mayoría de sus opciones, para facilitar su uso.8. Son gratuitos y de código libre.

GUI Turbo Assembler	WinAsm
<ol style="list-style-type: none"> 1. No permite modificar el tipo de letra. 2. Tiene un tema oscuro y un tema claro. 3. Maneja archivos .asm y .inc. 4. Ensambla y construye el código automáticamente. 5. Posee una opción para convertir a mayúscula o a minúscula todo el texto seleccionado. 6. No posee opciones para indentar o comentar código automáticamente, se debe hacer a mano. 7. Cuenta con un depurador poco eficiente pero eficaz. 8. De forma nativa sólo cuenta con soporte para TASM. 	<ol style="list-style-type: none"> 1. Permite modificar el tipo de letra. 2. Solo tiene un tema claro. 3. Maneja archivos .asm, .inc y .rc. 4. Requiere indicar nombre y ubicación en la que guardar el archivo ensamblado. 5. Si se desea cambiar a mayúscula o a minúscula una porción del texto, se deberá hacer manualmente. 6. Tiene opciones para indentar o desindentar el código seleccionado automáticamente, así como para comentar o descomentar. 7. No tiene depurador integrado. 8. De forma nativa sólo cuenta con soporte para MASM.

Conclusiones

Al elegir entre GUI Turbo Assemble y WinAsm como IDEs para programación en lenguaje ensamblador, la decisión depende de las prioridades del desarrollador y las necesidades del proyecto. GUI TASM se destaca por su automatización en el proceso de ensamblado y construcción del código, lo cual es ideal para proyectos que requieren una compilación rápida y eficiente sin necesidad de extensas personalizaciones visuales o soporte para múltiples tipos de archivos.

Por otro lado, WinAsm ofrece una fuerte ventaja en personalización visual y soporte amplio para diferentes tipos de archivos, incluyendo archivos de recursos (.rc). Es ideal para desarrolladores que valoran la flexibilidad en la interfaz de usuario y necesitan manejar proyectos complejos que involucran diversos tipos de recursos y archivos.

En conclusión, GUI Turbo Assembler es la mejor opción cuando se prioriza la automatización del proceso de construcción y la eficiencia en la compilación. Sin embargo, WinAsm destaca en entornos donde la personalización visual y el manejo flexible de múltiples tipos de archivos son cruciales para el desarrollo de proyectos complejos.

Bibliografía

Detmer, R. C. (2007). Essentials of 80x86 Assembly Language. Jones & Bartlett Learning.

Kyprianou, A. (s.f.). WinAsm Studio. WinAsm. Recuperado de <https://winasm.org/index.html>

Nath, L. J. (2023). GUI Turbo Assembler (Versión última actualización: 23 de noviembre de 2023). GitHub. <https://github.com/ljnath/GUI-Turbo-Assembler>

Swan, T. (1989). Mastering Turbo Assembler (2nd ed.). Carmel, IN: Howard W. Sams & Company, Hayden Books division of Macmillan Computer Publishing.