Introducción:

Este manual tiene como fin ayudar a comprender los aspectos técnicos del programa para poder facilitar el uso y funcionamiento del mismo, a continuación se explica, el ensamblador utilizado para esta práctica, los requisitos del ensamblador y la forma de poder ejecutar programas utilizando este ensamblador

Ensamblador Utilizado: Nasm

Descripción del ensamblador:

El Netwide Assembler o NASM, es un ensamblador libre para la plataforma Intel x86. Puede ser usado para escribir programas tanto de 16 bits como de 32 bits (IA-32). En el NASM, si se usan las bibliotecas correctas, los programas de 32 bits se pueden escribir de una manera tal para que sean portables entre cualquier sistema operativo x86 de 32 bits. El paquete también incluye un desensamblador, el NDISASM.

Historia:

El NASM fue escrito originalmente por Simon Tatham con ayuda de Julian Hall, y actualmente es desarrollado por un pequeño equipo en SourceForge que le hace mantenimiento. Fue lanzado originalmente bajo su propia licencia, pero más adelante fue cambiada por la licencia GNU Lesser General Public License, seguido de un número de problemas políticos causado por la selección de la licencia. La próxima versión del NASM, la 2.00, actualmente está siendo desarrollada bajo la bifurcación 0.99, e incluirá soporte para el x86-64 (x64/AMD64/Intel 64), junto con la respectiva salida de archivo objeto de 64 bits.

Características:

- El NASM puede generar varios formatos binarios en cualquier máquina, incluyendo COFF (y el ligeramente diferente formato Portable Executable usado porMicrosoft Windows), el a.out, ELF, Mach-O, y el formato binario nativo Minix. El NASM incluso define su propio formato binario, RDOFF, que es usado actualmente solamente por el proyecto del sistema operativo RadiOS).
- La variedad de formatos de la salida permite a uno portar los programas a virtualmente cualquier sistema operativo x86. Además, el NASM puede crear archivos binarios planos, usables para escribir boot loaders (cargadores de arranque), imágenes ROM, y varias facetas del desarrollo sistemas operativos. El NASM incluso puede correr en plataformas diferentes del x86, como SPARC y PowerPC, aunque no puede producir programas usables por esas máquinas.
- El NASM usa la tradicional sintaxis de Intel para el lenguaje ensamblador x86, mientras que otros ensambladores libres, como el ensamblador del GNU (GAS), utilizan la sintaxis de AT&T. También evita características como la generación automática de sobreescritura (override) de segmentos y la relacionada directiva *ASSUME* usada por el MASM y los ensambladores compatibles, pues estas pueden ser a menudo confusas -- los programadores deben seguir por sí mismos el contenido de los registros de segmento y la localización de variables a los que éstos se refieren

Requisitos para poder utilizar Nasm:

1. Tener instalado gcc (Compilador de C)
2. Tener instalado dosbox (Segunda forma de poder compilar el programa)
3. Una computadora de 64 bits
4. Un sistema operativo base linux de 64 bits

Pasos para poder compilar y ejecutar un programa:

1. Generar el archivo .asm
2. Compilar el archivo .asm con el siguiente código: **nams -f elf64 NombreArchivo.asm**, este comando traducirá el código en el archivo .asm a un código binario que el procesador pueda entender, como resultado de la ejecución de este código se tendrá un archivo .o donde estará el código binario que el procesador entiende.
3. Generar el ejecutable en base al archivo .o generado en el paso anterior, para esto se introduce lo siguiente en consola: **ld NombreEjecutable NombreArchivo.o**
4. Ejecutar el ejecutable generador el paso anterior se hará de la siguiente forma: **./NombreEjecutable**
5. En consola se podrá ver el resultado del programa escrito

Pasos para poder comilar y ejecutar un programa con dosbox:

1. Generar el archivo .asm
2. Compilar el .asm con el siguiente código: **nasm NombreArchivo.asm -fbin -o NombreCom.com,** este comando traducirá el .asm a código binario entendible por dosbox, como resultado de la ejecución se genera un archivo .com que será el archivo que se compilara en dosbox
3. Ejecutar el siguiente comando en la ruta donde se haya generado el archivo .com: **dosbox ./NombreCom.com -exit**, esto hará que dosbox compile el .com y ejecute el programa, el -exit es para indicarle a dosbox que cuando termine la ejecución se cierre