

Guía práctica lenguaje ensamblador

(c) Francisco Charte Ojeda

Introducción

- ¿Qué es el lenguaje ensamblador?
- Aplicaciones de la programación en ensamblador
- El lenguaje ensamblador de los x86
 - 8, 16, 32 y 64 bits
- A quién va dirigido este libro

Cómo usar este libro

- Estructura de la guía
- Ejercicios resueltos y propuestos
- Convenciones tipográficas

1. Conceptos fundamentales

- 1.1. Introducción
- 1.2. Arquitectura de un microprocesador
 - 1.2.1. La unidad de control
 - 1.2.2. La unidad aritmético lógica
 - 1.2.3. El banco de registros
 - 1.2.4. Buses
 - 1.2.5. Diseños clásicos
- 1.3. Arquitectura de los x86
 - 1.3.1. Modos de funcionamiento
 - 1.3.2. Banco de registros
 - 1.3.3. El registro de estado
- 1.4. Sistemas de numeración
 - 1.4.1. El sistema binario
 - 1.4.2. El sistema hexadecimal
 - 1.4.3. Notación al expresar datos numéricos
- 1.5. Representación de datos en memoria
 - 1.5.1. Caracteres y cadenas de caracteres
 - 1.5.2. Números enteros
 - 1.5.3. Números enteros con signo
 - 1.5.4. Otros formatos de representación
- 1.6. Resumen

2. Cómo escribir y ejecutar código ensamblador

- 2.1. Introducción
- 2.2. Editar código ensamblador
 - 2.2.1. Editores genéricos
 - 2.2.2. Editores específicos
- 2.3. Ensamblado o compilación
 - 2.3.1. MASM
 - 2.3.2. NASM
 - 2.3.3. Visual C++ y GCC
- 2.4. Depuración de programas en ensamblador

- 2.5. Entornos integrados
- 2.6. Cómo se trabajará en este libro
 - 2.6.1. Edición, ensamblado y depuración con Visual C++
 - 2.6.2. Código ensamblador embebido en GCC
 - 2.6.3. Comprobación de los ejemplos con MASM o NASM
- 2.7. Resumen

- 3. Transferencia de datos
 - 3.1. Introducción
 - 3.2. Instrucciones de transferencia de datos
 - 3.3. Modos de direccionamiento
 - 3.3.1. Direccionamiento por registro
 - 3.3.2. Direccionamiento inmediato
 - 3.3.3. Direccionamiento directo
 - 3.3.4. Direccionamiento indirecto
 - 3.3.5. Direccionamiento indexado
 - 3.3.6. Direccionamiento escalado
 - 3.4. Modos de direccionamiento y registros de segmento
 - 3.5. Acceso a memoria mediante punteros
 - 3.5.1. Etiquetas de datos en ensamblador
 - 3.5.2. Variables C/C++ como etiquetas de datos
 - 3.5.3. Obtener la dirección de una etiqueta
 - 3.5.4. Punteros compuestos de segmento y desplazamiento
 - 3.6. Trabajar con la pila
 - 3.6.1. Inicialización de la pila
 - 3.6.2. La pila en el ensamblador embebido
 - 3.7. Otras operaciones de transferencia de datos
 - 3.7.1. Intercambio de datos
 - 3.7.2. Búsqueda indexada
 - 3.7.3. Transferencia condicional
 - 3.8. Resumen

- 4. Estructuras de control
 - 4.1. Introducción
 - 4.2. Saltos condicionales y bits de estado
 - 4.2.1. Instrucciones que afectan al registro de estado
 - 4.2.2. Inspección de los bits de estado durante la depuración
 - 4.2.3. Instrucciones de salto condicional
 - Igualdades y desigualdades
 - Relación entre números sin signo
 - Relación entre números con signo
 - Otras instrucciones de salto
 - 4.3. Implementación de estructuras condicionales
 - 4.3.1. Condicionales simples
 - 4.3.2. Condicionales con puerta falsa
 - 4.3.3. Condicionales compuestos
 - 4.3.4. Condicionales con operando común
 - 4.4. Implementación de bucles
 - 4.4.1. Bucles controlados por condición
 - 4.4.2. Bucles controlados por contador

- 4.4.3. Bucles anidados
- 4.5. Resumen

5. Cálculos aritméticos

- 5.1. Introducción
- 5.2. Operaciones aritméticas básicas
 - 5.2.1. Sumas
 - Conversiones de tamaño
 - Sumas con acarreo
 - 5.2.2. Restas
 - 5.2.3. Incrementos y decrementos
 - 5.2.4. Productos
 - 5.2.5. Divisiones
- 5.3. Formatos BCD y ASCII
- 5.4. Operaciones aritméticas en punto flotante
 - 5.4.1. Representación de números en punto flotante
 - Normalización de la mantisa
 - Codificación del exponente
 - Codificación del signo
 - 5.4.2. Registros de la FPU
 - 5.4.3. Tipos de datos
 - 5.4.4. Introducción de datos en la FPU
 - 5.4.5. Ejecución de operaciones
 - 5.4.6. Recuperación de datos de la FPU
 - 5.4.7. Ejemplo práctica de uso de la FPU
- 5.5. Resumen

6. Trabajar al nivel de bits

- 6.1. Introducción
- 6.2. Operaciones booleanas
 - 6.2.1. La operación lógica OR
 - 6.2.2. La operación lógica AND
 - 6.2.3. La operación lógica XOR
 - 6.2.4. La operación lógica NOT
- 6.3. Comprobación de bits individuales
- 6.4. Rotaciones y desplazamientos
 - 6.4.1. Rotación de bits
 - 6.4.2. Desplazamiento de bits
- 6.5. Instrucciones avanzadas de trabajo con bits
- 6.6. Resumen

7. Estructuración del código

- 7.1. Introducción
- 7.2. Llamadas y retornos de subrutinas
 - 7.2.1. Llamada a una subrutina
 - Llamadas inter-privilegio
 - Llamadas inter-tarea
 - 7.2.2. Cuerpo de una subrutina
 - 7.2.3. Retorno de una subrutina
 - 7.2.4. Una rutina de retardo

- 7.3. Transferencia de parámetros y resultados
 - 7.3.1. A través de registros
 - 7.3.2. A través de la pila
 - 7.3.3. A través de memoria compartida
 - 7.3.4. Devolución de resultados y de errores
- 7.4. Procedimientos externos en ensamblador
 - 7.4.1. Prólogo y epílogo
 - 7.4.2. Acceso a los parámetros de entrada y devolución de resultados
 - 7.4.3. Compilación, ensamblado y enlace
- 7.5. Resumen

- 8. Procesar cadenas de bytes
 - 8.1. Introducción
 - 8.2. Aspectos genéricos
 - 8.2.1. Punteros de origen y destino
 - 8.2.2. Sentido del recorrido
 - 8.3. Transferencia de datos individuales
 - 8.3.1. Repetición automática
 - 8.4. Transferencia de bloques de memoria
 - 8.5. Búsquedas y comparaciones en cadenas de datos
 - 8.5.1. Búsqueda de un dato en una cadena
 - 8.5.2. Comparación de cadenas
 - 8.6. Resumen

- 9. Ensamblador en entornos de 16 bits
 - 9.1. Introducción
 - 9.2. Segmentación de la memoria
 - 9.2.1. Registros de segmento
 - 9.2.2. Aritmética de generación de direcciones
 - 9.2.3. Acceso a datos alojados en la memoria
 - 9.3. Acceso a los servicios del DOS
 - 9.3.1. Entrada y salida por la consola
 - 9.3.2. Comunicación serie y paralelo
 - 9.3.3. Fecha y hora
 - 9.3.4. Gestión de memoria
 - 9.4. Acceso a los servicios de la BIOS
 - 9.4.1. Acceso al adaptador de vídeo
 - 9.4.2. Lectura del teclado
 - 9.4.3. Configuración del sistema
 - 9.4.4. Acceso a unidades de disco
 - 9.4.5. Puertos serie y paralelo
 - 9.4.6. Fecha y hora
 - 9.5. Resumen

- 10. Ensamblador en entornos de 32 bits
 - 10.1. Introducción
 - 10.2. El modo protegido
 - 10.2.1. Registros de control del procesador
 - 10.2.2. Modificación de los registros de control
 - 10.2.3. Segmentos y selectores

- 10.2.4. Descriptores de segmentos
 - Tipos de segmentos
- 10.2.5. Tablas de descriptores
- 10.2.6. De vuelta a los selectores de segmento
- 10.2.7. Direccionamiento en modo protegido
- 10.2.8. Entrada y salida del modo protegido
 - Preparación de la GDT
 - Cálculo de direcciones físicas
 - Núcleo del programa
- 10.2.9. Interrupciones en modo protegido
- 10.3. DPMI
 - 10.3.1. Anfitriones DPMI
 - 10.3.2. Clientes DPMI
 - 10.3.3. Detectar la presencia de un anfitrión DPMI
 - 10.3.4. Activación del modo protegido
 - 10.3.5. Servicios DPMI
 - 10.3.6. Un ejemplo
- 10.4. Extensores DOS
- 10.5. Resumen

A. Referencia de instrucciones

- A.1. Instrucciones aritméticas
- A.2. Instrucciones lógicas y de rotación/desplazamiento
- A.3. Instrucciones de conversión
- A.4. Instrucciones de cadena
- A.5. Instrucciones de transferencia de datos
- A.6. Instrucciones de control de flujo
- A.7. Instrucciones de entrada/salida
- A.8. Instrucciones de control
- A.9. Otras instrucciones

B. Soluciones a los ejercicios

Índice alfabético