

Especificación del lenguaje ensamblador usado durante el curso

Características generales de la arquitectura	
Propiedad	Valor
Unidad mínima direccionable (byte)	8 bits
Tamaño de una palabra	16 bits
Tamaño de una dirección	16 bits
Codificación de datos	Little endian
Codificación de enteros	Complemento a dos
Alineación de datos en memoria	Opcional

Registros disponibles	
Nombre del registro	Código máquina
RL0 (byte)	001
RH0 (byte)	010
RX0 (palabra)	011
RL1 (byte)	101
RH1 (byte)	110
RX1 (palabra)	111

Conjunto de instrucciones		
Sintaxis	Código máquina	Descripción
NOP	00000000	Instrucción vacía
TRAP	11111111	Interrupción del programa
INC <i>ra</i>	01000 aaa	Incremento de registro
DEC <i>ra</i>	01001 aaa	Decremento de registro
TEST <i>ra</i>	01010 aaa	Examina el contenido del registro. Si es cero, inhabilita la ejecución de la siguiente instrucción.

MOV <i>ra</i> , <i>entero</i>	01011 aaaa eeeeeeeeeeeeeeeeeeee	Asignación a registro mediante direccionamiento inmediato. Si el registro es de un byte, se almacena el byte menos significativo.
MOV <i>ra</i> , [<i>entero</i>]	10000 aaaa eeeeeeeeeeeeeeeeeeee	Lectura de un byte o palabra en memoria (dependiendo del ancho del registro destino) mediante direccionamiento directo. El valor del entero se interpreta como una dirección sin signo.
MOV [<i>entero</i>], <i>ra</i>	10001 aaaa eeeeeeeeeeeeeeeeeeee	Escritura de un byte o palabra en memoria (dependiendo del ancho del registro origen) mediante direccionamiento directo. El valor del entero se interpreta como una dirección sin signo.
MOV <i>ra</i> , [<i>rb</i>]	1001000000 aa abbb	Lectura de un byte o palabra en memoria (dependiendo del ancho del registro destino) mediante direccionamiento indirecto. El contenido del registro se interpreta como una dirección sin signo.
MOV [<i>ra</i>], <i>rb</i>	1001100000 aa abbb	Escritura de un byte o palabra en memoria (dependiendo del ancho del registro origen) mediante direccionamiento indirecto. El contenido del registro se interpreta como una dirección sin signo.
JMP <i>entero</i>	11000000 eeee eeeeeeeeeeeeeeee	Salto mediante direccionamiento directo. El valor del entero se interpreta como una dirección sin signo.
JMP <i>ra</i>	11001 aaa	Salto mediante direccionamiento indirecto. El contenido del registro se interpreta como una dirección sin signo.
JMP OFFSET <i>entero</i>	11100000 eeee eeeeeeeeeeeeeeee	Salto mediante direccionamiento relativo. El valor del entero se interpreta como un desplazamiento sobre la dirección de la siguiente instrucción.

JMP OFFSET <i>ra</i>	11101aaa	Salto mediante direccionamiento relativo. El contenido del registro se interpreta como un desplazamiento sobre la dirección de la siguiente instrucción.
----------------------	----------	--

Directivas del ensamblador	
Sintaxis	Descripción
ALIGN <i>entero</i>	Posicionamiento en la siguiente dirección de memoria múltiplo de <i>entero</i>
ALLOC BYTE <i>identificador</i>	Apartar un byte bajo la etiqueta <i>identificador</i>
ALLOC BYTE <i>identificador, entero</i>	Apartar un byte bajo la etiqueta <i>identificador</i> , inicializado con <i>entero</i>
ALLOC BYTE ARRAY <i>identificador, entero</i>	Apartar <i>entero</i> bytes bajo la etiqueta <i>identificador</i>
ALLOC WORD <i>identificador</i>	Apartar una palabra bajo la etiqueta <i>identificador</i>
ALLOC WORD <i>identificador, entero</i>	Apartar una palabra bajo la etiqueta <i>identificador</i> , inicializado con <i>entero</i>
ALLOC WORD ARRAY <i>identificador, entero</i>	Apartar <i>entero</i> palabras bajo la etiqueta <i>identificador</i>
SECTION <i>identificador</i>	Inicio de sección
IMPORT <i>identificador</i>	Importación de etiqueta estática
IMPORT DYNAMIC <i>identificador</i>	Importación de etiqueta dinámica
EXPORT <i>identificador</i>	Exportación de etiqueta
BEGIN MACRO <i>identificador</i>	Inicio de macro
END MACRO	Fin de macro
EXPAND <i>identificador</i>	Expansión de macro

Un identificador consiste en una secuencia de entre 1 y 15 caracteres ASCII alfanuméricos y guiones bajos, donde el primer carácter no puede ser un dígito. Una instrucción puede tener asociada una etiqueta, la cual es un identificador que aparece antes de la instrucción y con dos puntos (:) como separador. No pueden existir etiquetas repetidas en el mismo código fuente.

Las directivas IMPORT, EXPORT y definiciones de macros deben aparecer antes de cualquier sección. El código, declaraciones de variables y expansiones de macros deben ir dentro de una sección. Una macro sólo puede incluir código. No pueden existir macros con nombres repetidos en el mismo código fuente.

Toda sección debe ser "text", "data", "rdata" o "bss". Todos los datos en las secciones "text", "data" y "rdata" deben estar inicializados. Ningún dato en la sección "bss" puede estar inicializado. El nombre de una sección definida por el usuario empieza con uno de los nombres anteriores, un punto y luego un identificador (por ejemplo, "text.usuario"). Una sección puede reabrirse y es equivalente a si el contenido se escribiera inmediatamente después del contenido previo de la misma sección.

Especificación del código objeto y código ejecutable	
Código máquina	Descripción
eeeeeeee	Tipo de archivo: 00000000 para objeto o 00000001 para ejecutable
eeeeeeeeeeeeeeee (sólo en ejecutables)	Dirección del punto de entrada
eeeeeeeeeeeeeeee	Número de secciones
...	La información de cada sección
eeeeeeeeeeeeeeee	Número de entradas en la tabla de símbolos
...	La información de cada entrada de la tabla de símbolos

Especificación de una sección	
Sintaxis	Descripción
<i>byte₀ byte₁ ... byte_{n-1} \0</i>	Nombre de sección con terminador nulo (máximo n=15 caracteres no nulos)
eeeeeeee	Banderas de sección: 00000001 (con permisos de ejecución), 00000010 (con permisos de escritura), 00000100 (con estado inicializado en blanco "bss").
eeeeeeeeeeeeeeee (sólo en ejecutables)	Dirección inicial de la sección
eeeeeeeeeeeeeeee	Longitud en bytes de la sección
... (excepto "bss")	Código máquina de la sección
eeeeeeeeeeeeeeee (excepto "bss")	Número de entradas de relocalización (dependencias)
... (excepto "bss")	La información de cada entrada de relocalización

Especificación de una entrada de relocalización	
Sintaxis	Descripción
eeeeeeeeeeeeeeee	Dirección del primer byte de la dependencia (relativo al inicio de la sección)
eeeeeeee	Tipo de la dependencia: 00000000 (dependencia de sección), 00000001

	(dependencia de etiqueta importada)
eeeeeeeeeeeeeeee	Número de sección o etiqueta (según la numeración de secciones o según la tabla de símbolos) con la cual hay dependencia

Especificación de una entrada de la tabla de símbolos (sólo etiquetas)	
Sintaxis	Descripción
<i>byte₀ byte₁ ... byte_{n-1} \0</i>	Nombre del identificador con terminador nulo (máximo n=15 caracteres no nulos)
eeeeeeee	Tipo de símbolo: 00000000 (etiqueta local), 00000001 (etiqueta importada estáticamente), 00000010 (etiqueta importada dinámicamente), 00000011 (etiqueta exportada), 00000100 (archivo externo)
eeeeeeeeeeeeeeee (sólo etiquetas importadas dinámicamente, sólo ejecutables)	Número de símbolo (según la tabla de símbolos) del archivo que provee la etiqueta importada dinámicamente.
eeeeeeeeeeeeeeee (sólo etiquetas locales y exportadas)	Número de la sección que contiene a la etiqueta
eeeeeeeeeeeeeeee (sólo etiquetas locales y exportadas)	Dirección de la etiqueta. En códigos objeto: relativa al inicio de la sección que la contiene; en códigos ejecutables: dirección virtual final.

Ejemplo de sintaxis
EXPORT fun SECTION data ALLOC BYTE b, 5 ALIGN 2 ALLOC WORD var, 2 SECTION data.otros ALLOC WORD raro, 5

```
SECTION bss
```

```
    ALLOC WORD ARRAY arreglo, 10
```

```
SECTION text
```

```
fun: NOP
```

```
eti: MOV [var], RX0      # comentario, poniendo un 10 en var
```