

Ensambladores

- Código de ensamblaje: Se utilizan nombres para las instrucciones y también se utilizan nombres para las direcciones de memoria.

```
MOV a, R1
ADD #2, R1
MOV R1, b
```

- Ensamble de dos pasos:

- Primer paso: todos los identificadores se asignan a direcciones de memoria (desplazamiento 0) por ejemplo, sustituya 0 por a, y 4 por b
- Segunda pasada: producir código máquina reubicable:

0001 01 00 00000000 *	→	0001 01 00 00001111
0011 01 10 00000010		0011 01 10 00000010
0010 01 00 00000100 *		0010 01 00 00010011

* Bit de reubicación

25

Ensambladores

- Todavía no conocemos la dirección, por lo que el código fuente debe leerse dos veces

Código Fuente (ensamblador)			Primera pasada		Segunda pasada	
			Dirección relativa	Instrucción mnemónica	Dirección relativa	Instrucción mnemónica
JOHN	START	0				
	USING	*, 15				
	LOAD	1, FIVE	0	L 1, - (0,15)	0	L 1, 16 (0,15)
	ADD	1, FOUR	4	A 1, - (0,15)	4	A 1, 12 (0,15)
	STO	1, TEMP	8	ST 1, - (0,15)	8	ST 1, 20 (0,15)
FOUR	DC	F'4'	12	4	12	4
FIVE	DC	F'5'	16	5	16	5
TEMP	DS	1F	20	-	20	-
	END					

26

Cargadores y editores de enlaces

- Cargador: tomar código máquina reubicable, alterar las direcciones y colocar las instrucciones alteradas en la memoria.
- Editor de enlaces: tomar muchos programas (reubicables) de código máquina (con referencias cruzadas) y producir un solo archivo.
 - Necesita realizar un seguimiento de la correspondencia entre los nombres de las variables y las direcciones correspondientes en cada fragmento de código.

27

La agrupación de fases

Front End : Análisis + Generación de Código
intermedio vs.

Back End : Generación de código + Optimización

Número de pasadas:

Una pasada: requiere lectura/escritura archivos intermedios

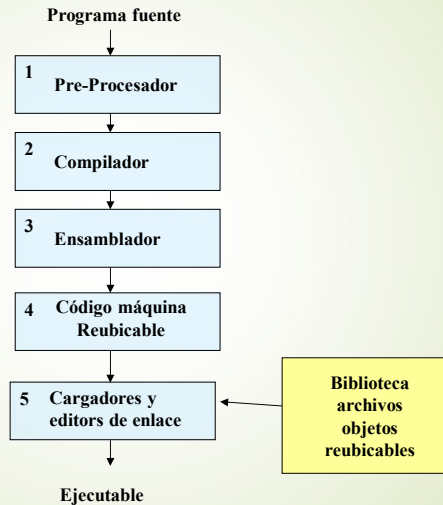
Menos pasadas: más eficiencia.

Sin embargo: menos pasadas requieren una administración de memoria más sofisticada y una interacción de fase del compilador.

Equilibrio...

28

Sistema de procesamiento del lenguaje



29

Herramientas de construcción de compiladores

- Generadores de analizadores : Producen analizadores de sintaxis
- Generadores de escáner : Producen analizadores léxicos
- Motores de traducción dirigidos por sintaxis : Generan código intermedio
- Generadores automáticos de código : Generan código real
- Motores de flujo de datos : Apoyan Optimización

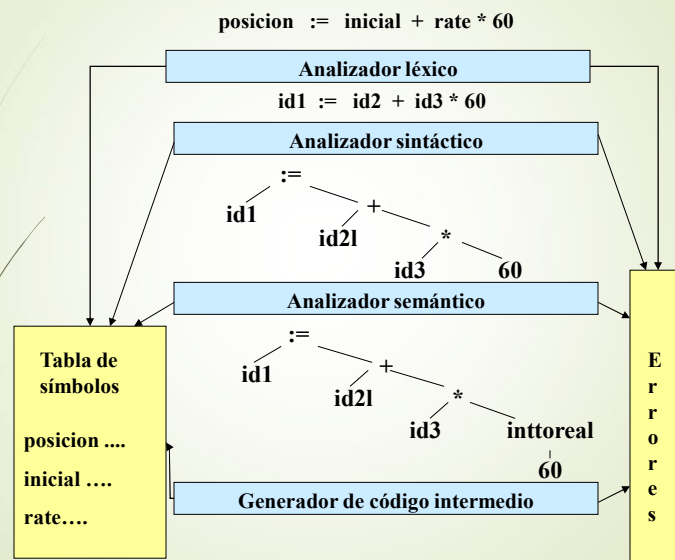
30

Herramientas

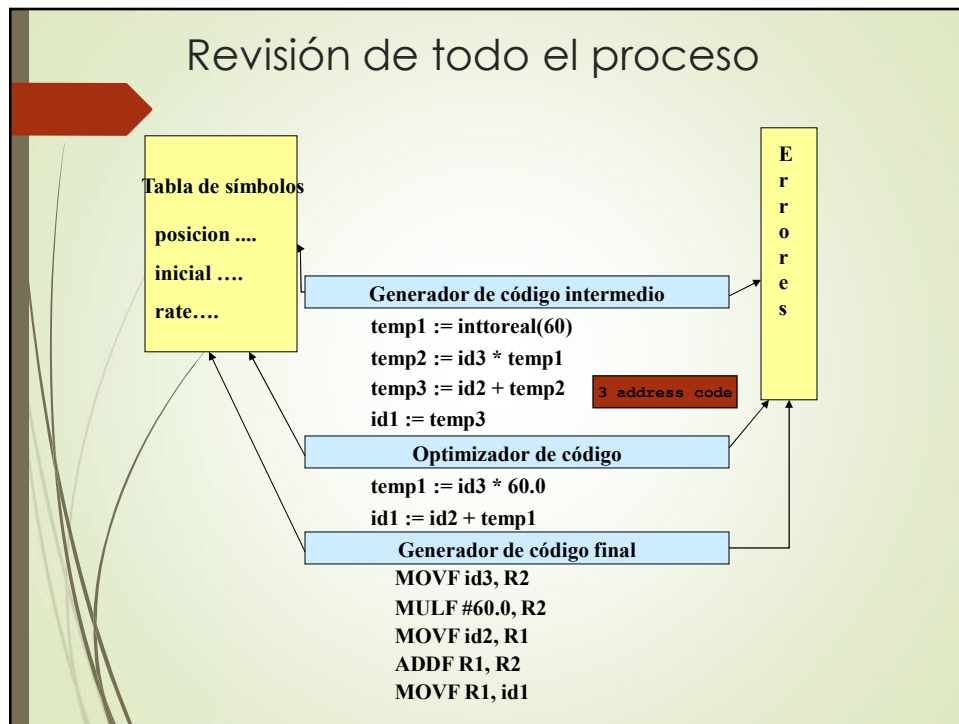
- Existen herramientas para ayudar en el desarrollo de algunas etapas del compilador
- Lex (Flex) - Generador de analizador léxico
- Yacc (Bison) - Generador de analizador sintáctico.

31

Revisión de todo el proceso



32



33