

## Estructura del Computador - 66.70

### Compilación - Ensamblado - Enlace - Carga

Lenguaje de alto nivel -> Lenguaje ensamblador (Assembler)

COMPILACIÓN

Lenguaje ensamblador -> Código de máquina

ENSAMBLADO

Combinación de módulos ensamblados de forma separada

ENLACE

Carga del programa en memoria y preparación para la ejecución

CARGA

La **compilación** se encarga exclusivamente de trasladar un programa escrito por un programador en un lenguaje de alto nivel (C, Python, Go, etc) a un programa en lenguaje ensamblador.

Se encarga de analizar el programa de la siguiente manera:

- Análisis lexicográfico: reconoce los elementos básicos del programa.
- Análisis sintáctico: a partir de los elementos básicos reconoce la estructura del programa.
- Análisis semántico:
  - Análisis de nombres: se le asigna a cada nombre una variable particular y una dirección de memoria.
  - Análisis de tipos: reconoce todos los tipos de variables que el programa requiera.

Luego de estos análisis se produce la asignación de acciones y se escribe para cada línea del programa en lenguaje de alto nivel 2 o más líneas de lenguaje ensamblador.

El **ensamblado** escribe cada línea del programa en Assembler a código de máquina (1 y 0). La relación es 1 a 1.

El ensamblador de ARC es un ensamblador de 2 pasadas, es decir que leerá 2 veces el código, ofreciendo como ventaja el uso de símbolos del programa antes de su definición. Además permite la ubicación de palabras en memoria, el uso de macros y la posibilidad de definir etiquetas para direcciones de memoria constantes.

El ensamblado comienza con un preproceso donde se expanden las macros y se registran y reemplazan definiciones.

En la primera pasada se genera la tabla de símbolos, cuya utilidad será muy importante en el enlace y carga. La tabla de símbolos

contiene para cada símbolo su dirección en memoria, si es reubicable o no y si es global o extern.

En la segunda pasada se genera el código de máquina.

Del ensamblado sale el código de máquina, la tabla de símbolos y además la dirección de memoria de la primera instrucción a ejecutar.

El **enlace** realizado por el linker se encarga de combinar 2 o más módulos que hayan sido ensamblados de forma separada. Para eso puede reubicar las direcciones internas cuando hay direcciones que entre módulos se pisan. Además resuelve las referencias externas y globales. Por estos últimos 2 procesos necesita la tabla de símbolos.

El **loader** carga el programa en memoria y por esta razón puede reubicar direcciones para que coexistan varios programas al mismo tiempo. Además inicializa ciertos registros como el %sp y el %pc.