



6. – Lenguajes y Ciclo de Desarrollo

Introducción a los microprocesadores
2015

Lenguajes y Ciclo de Desarrollo

- Lenguajes
 - Cód. Máquina
 - Lenguajes alto nivel
 - Lenguaje ensamblador
- Ciclo de desarrollo

Lenguajes: Código Máquina

- Nivel de abstracción bajo
 - ceros y unos.
- Es lo único que entiende el procesador.
- Objetivo:
 - código de máquina → memoria.
- Imposible entender, menos aún corregir o modificar por un humano.
 - Necesidad de mayor nivel de abstracción

Código Máquina		
0011	1010	3A
0000	0001	01
1011	0010	B2
0100	0111	47
0011	1010	3A
0000	0010	02
1011	0010	B2
1000	0000	80
0011	0010	32
0000	0000	00
1011	0010	B2

Lenguajes: Lenguaje de Alto Nivel

- Pascal, C, Java

`Suma := Oper1 + Oper2`

- Mayor nivel de abstracción
- No lo comprende el procesador
 - Necesita **Traducción** : Se le llama “Compilación”
 - Traductor = **Compilador** (programa automático)
- Es independiente del computador elegido (**Portable**).



Lenguajes:

Lenguaje ensamblador

- Nivel de abstracción intermedio
- Específico del **procesador** usado
- Traducción más simple
 - Una línea corresponde a una instrucción
 - Traducción = Ensamblado
 - Compilador = Ensamblador (igual que el lenguaje)

Código Máquina	Ensamblador	Alto Nivel
3A 01 B2	LD A, (OPER1)	SUMA:=OPER1 + OPER2
47	LD B, A	
3A 02 B2	LD A, (OPER2)	
80	ADD A, B	
32 00 B2	LD (SUMA), A	

Lenguajes:

Lenguaje ensamblador

¿Cuándo usar Alto Nivel y cuándo Ensamblador ?

- Alto nivel siempre que sea posible
- A veces Ensamblador (assembler) por:
 - Acceso a todos los recursos del procesador
 - Eficiencia
 - en tiempo de ejecución
 - en tamaño de código
 - No disponibilidad de compilador
- En general mixto
 - Secciones críticas en assembler
 - Resto en alto nivel

Lenguajes y Ciclo de Desarrollo

- ✓ Lenguajes
- Ciclo de desarrollo
 - Diseño
 - Codificación
 - Traducción
 - Prueba y depuración
 - Documentación y mantenimiento

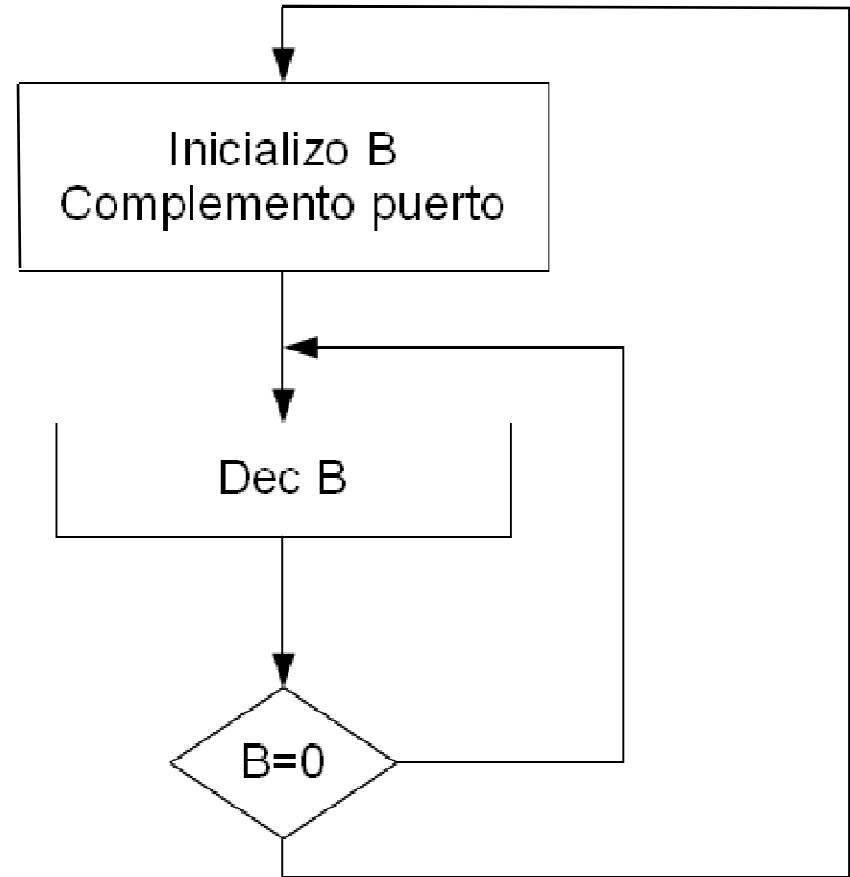
Ciclo de Desarrollo Diseño

- Nunca empezar escribiendo código
 - “Spaghetti code”
- Pseudocódigo:
 - Estructuras de control de lenguajes de alto nivel pero con libertades para incluir recursos hardware

```
mientras (true)
{
  complemento valor de salida
  b = semiperíodo
  mientras ( b !=0 )
  {
    decremento b
  }
}
```


Ciclo de Desarrollo Diseño

- Diagramas de flujo
 - Menor nivel de abstracción
 - Más fácil pasarlo a assembler
 - Más confuso para programas complejos



Ciclo de Desarrollo Diseño

Usen el que prefieran,

pero **USEN!!**

Ciclo de Desarrollo Codificación

- Una vez diseñado el programa hay que escribirlo
- Editor de texto para crear un **archivo fuente** (programa fuente)
- En nuestro caso el programa será escrito en **ensamblador**.

Ciclo de Desarrollo Traducción

- Traducción = Ensamblado
- Se genera Lista de Errores
 - Si hay errores → corregir la codificación (programa fuente)
 - Si no hay errores → Código Máquina.



Ciclo de Desarrollo Traducción

- Programa en archivos separados
- Ensamblador o Assembler
 - Traduce cada módulo
 - Direcciones relativas al comienzo del módulo
 - Símbolos externos (definidos en otro módulo)
- Linker
 - Encadena los módulos
 - Resuelve pendientes
 - Direcciones de comienzo ahora conocidas
 - Todos los símbolos tienen que estar definidos en algún módulo
- Bibliotecas ya ensambladas
 - De terceros o propias

Ciclo de Desarrollo Prueba y Depuración

- Objetivo: verificar que el programa hace lo que debe hacer.
- Enfoque tradicional:
 - Ejecución controlada del programa
 - Comparación de resultados con resultados esperados
- Depende del ambiente de desarrollo
 - Sistema de desarrollo: donde escribo y pruebo los programas
 - Sistema destino: donde va a correr el programa final

Ciclo de Desarrollo Prueba y Depuración

Herramientas:

- Simulador (no sirve para tiempo)
- Depurador o Debugger (paso a paso, breakpoint)
- Sistema destino
 - Kit de desarrollo
 - Placa prototipo
- Analizador Lógico
- Emulador (no sirve para tiempos)
- Caso laboratorio:
 - Sistema de desarrollo: PC
 - Sistema destino
 - Simulador
 - Kit de desarrollo en placa DE0

Ciclo de Desarrollo

Documentación y Mantenimiento

- Si no se documenta se hace muy difícil hacer mantenimiento.
- Documentar lleva mucho tiempo y ahorra mucho más.
- No documentar para ahorrar tiempo = perder tiempo
- ¿Como documentar?
 - Papel y lápiz
 - Editor de texto y/o planilla de cálculo
 - Software especializado.