

Clase #04 de 27

Traductores

Abril 8, Lunes

Agenda para esta clase

- Análisis y Síntesis de Hello.c
- Intervalo
- Traductores
- Las Fases de Traducción y los Tipos de Errores: Trabajo #1

Análisis y Síntesis de Hello.c

Análisis de ‘Hello, World!’

```
#include <stdio.h>
```

```
main() {  
    printf("Hello, world!\n");  
}
```

Incluye información acerca de la biblioteca estándar

Define una función llamada *main* que no recibe valores argumento. Las sentencias de *main* se encierran entre llaves

- Estructura de un programa
 - vs. Pascal
 - Lineal vs. Jerárquico
 - Rol de *main*
 - Preprocesador
 - Biblioteca estándar: Entrada/Salida
 - vs Framework
 - Rol de llaves { }
 - Función *printf*
 - Pasaje de argumentos a funciones
- Constantes de cadena o literal cadena
 - Secuencia de escape
 - Punto y coma como terminador
 - vs. Pascal
 - Analizar que es “palabra” del LP y que no
 - Identificadores:
main y *printf*--
¿Cuál es el autómatafinito que los reconoce?

main llama a la función de biblioteca estándar *printf* para imprimir esa secuencia de caracteres. \n representa el carácter nueva-linea

C++ source #1

① ② A A A M C ±

```
1 /* Hello.cpp
2   C11
3   JMS
4   2015
5 */
6
7 #include <stdio.h>
8
9 int main(void){
10   printf("Hello, World!\n");
11 }
```

#1 with x86-64 clang 4.0.0 x

x86-64 clang 4.0.0

-x c -std=c11

11010 .LX0: .text # Intel

A A A C

```
1 main:                                # @main
2   push    rbp
3   mov     rbp, rsp
4   sub     rsp, 16
5   movabs  rdi, .L.str
6   mov     al, 0
7   call    printf
8   xor     ecx, ecx
9   mov     dword ptr [rbp - 4], eax # 4-byte Spill
10  mov    eax, ecx
11  add    rsp, 16
12  pop    rbp
13  ret
14
15 .L.str:
16  .asciz "Hello, World!\n"
```

Generalización de hello.c

Programa General

- Un programa es una secuencia de funciones. Forma general:
main
f
g
...
• Forma general de una función
Tipo Nombre (Parámetros)
{ Cuerpo }
• Cuando se corre (ejecuta) un programa, por convención, main es la primera función invocada por el ambiente de ejecución (e.g., sistema operativo). Todo programa debe tener un main con o sin parámetros

hello.c

- Este programa define solo la función main
- Como toda función, main puede tener o no parámetros
- Este main invoca a printf con una cadena literal (cadena constante) como argumento. "Entre comillas".

Intervalo

20 minutos

Traductores

Ensambladores, Intérpretes & Compiladores

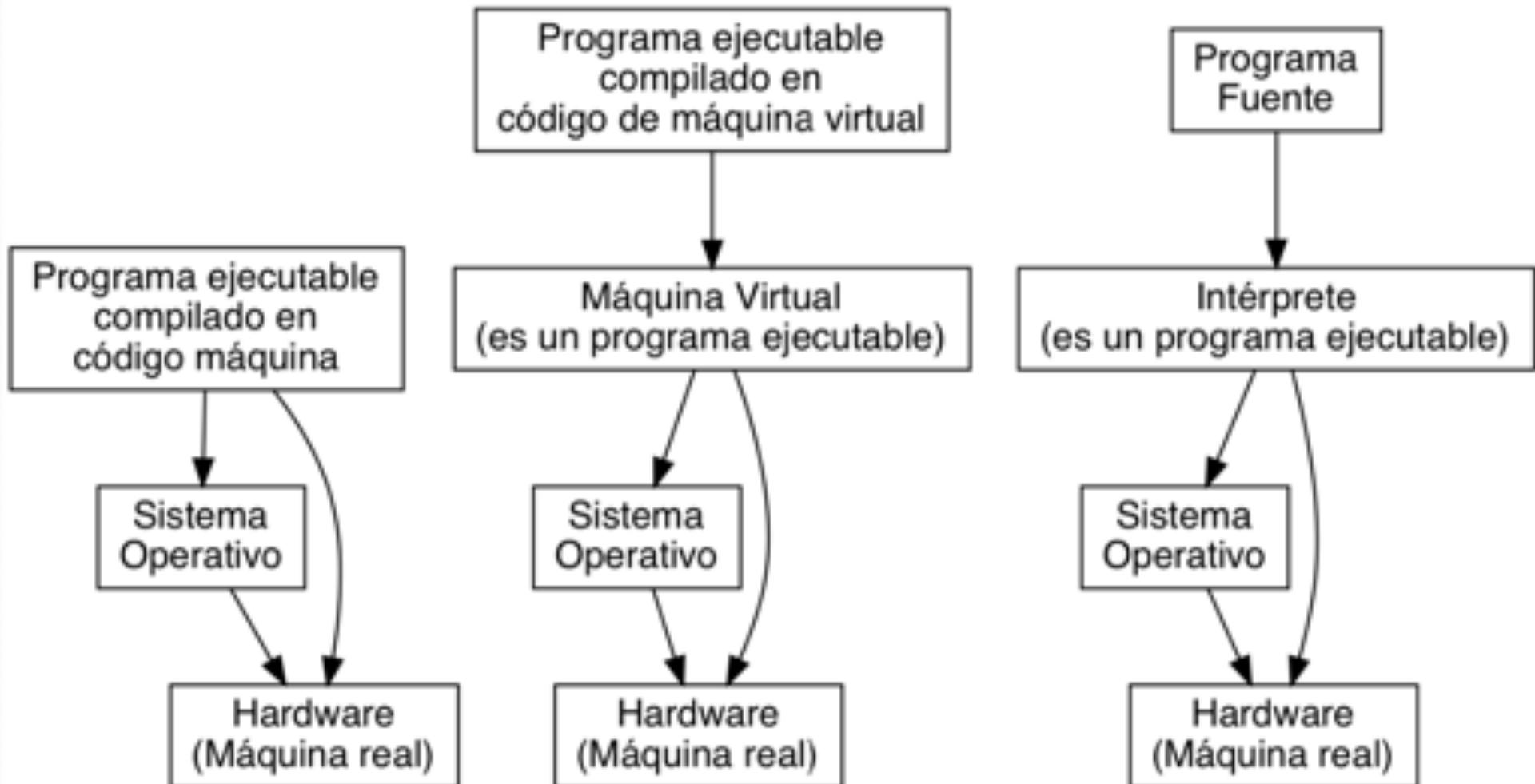
Traductores: Ensambladores, Intérpretes & Compiladores

- Los tres buscan **traducir** un archivo con un **texto** que es el programa en **lenguaje fuente** para que sea ejecutado y exponga un **comportamiento**
- Características sobresalientes de cada traductor:
 - Ensambladores
 - **Bajo nivel**
 - Intérpretes
 - **Dinámico**
 - Compiladores
 - **Eficiente**

Máquinas: Virtuales y Reales

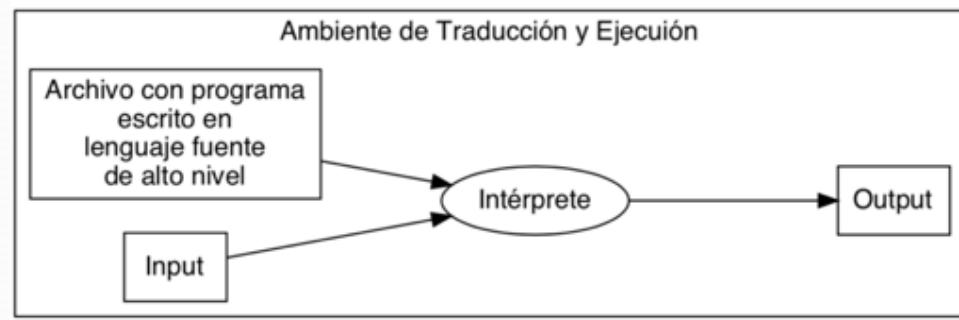
- Real
 - Física
 - Hardware
 - Sin capa de abstracción
 - Comportamiento físico directo
- Virtual
 - Lógica
 - Software
 - Capa de abstracción sobre máquina real
 - Comportamiento físico dado por la máquina real.

Ejecución – Acceso al hardware

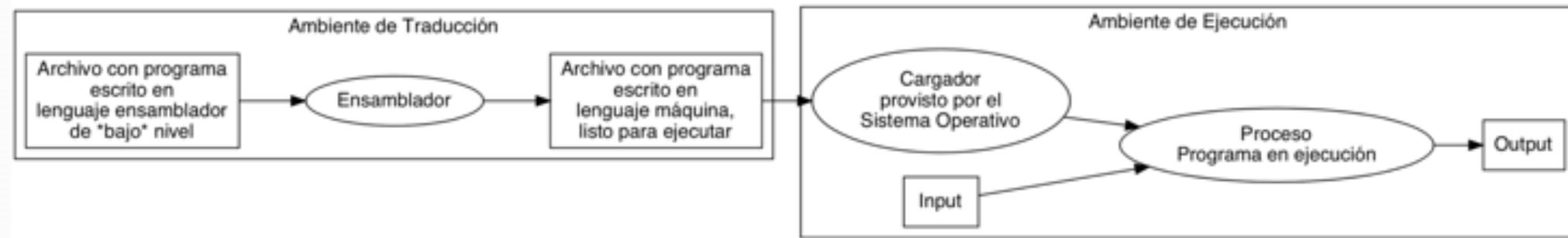


Traducción y Ejecución

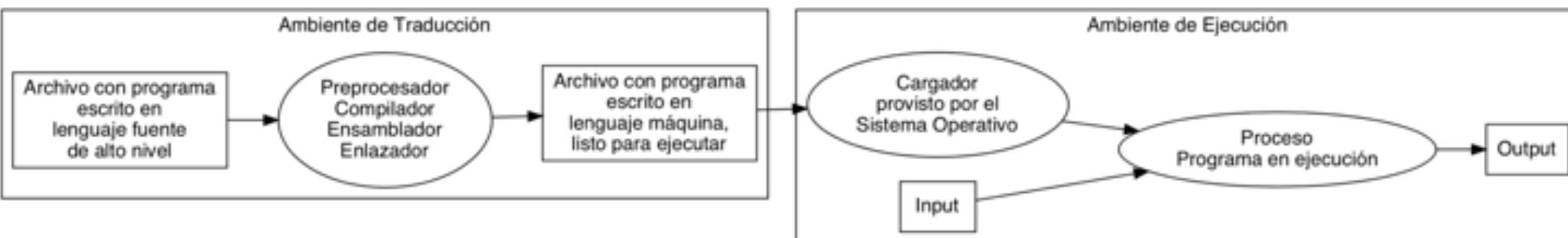
Intérprete



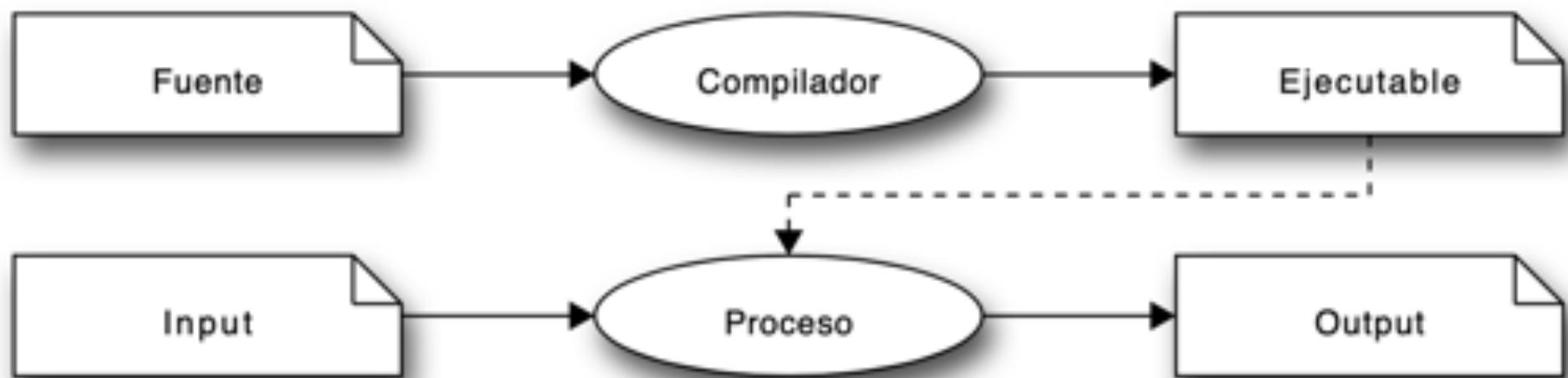
Ensamblador



Compilador



Ambiente de Traducción y Ejecución del Compilador



Ambiente de Traducción del Compilador



Comparación de Traductores

Aspecto	Ensamblador	Intérprete	Compilador
Función	Traduce “uno a uno” mnemónico a instrucción en código máquina.	Procesa línea por línea, verifica errores y ejecuta; frena si una línea tiene error.	Procesa todo el programa, informa errores o genera ejecutable.
Nivel de abstracción	▼ Bajo	▲ Alto	▲ Alto
Tiempo de traducción	▲ Muy rápido.	▼ En la ejecución.	▲ Rápido.
Veces que se traduce	▲ Una	▼ Con cada ejecución	▲ Una
Tiempo de ejecución	▲ Muy rápido.	▼ Lento.	▲ Rápido.
Espacio de ejecución	▲ Bajo.	▼ El del Intérprete.	▲ Bajo.
Espacio de traducción	▲ Bajo.	▼ En la ejecución.	▲ Bajo.
Depuración y cambios en ejecución	▼ Poco flexible.	▲ Más flexible.	▼ Menos flexible.
Producto intermedio	▼ Sí	▲ No	▼ Sí
Ejemplos	Lenguaje assembler para x86-64, ARM MOV AX, 42	JavaScript, PHP, Python a=42	C, C++, Java, C#, Objective-C, Swift. int a=42;

Las Fases de Traducción y los Tipos de Errores

Trabajo #1

Las Fases de Traducción



- Opciones del compilador para la ejecución de cada paso por separado
- Ejemplo con cc.

Identificación de Errores

- Errores por Tiempo
 - Diseño
 - Traducción
 - Ejecución
- Errores por Fase de Traducción
 - Preprocesamiento
 - Compilación
 - Ensamblado
 - Vinculación
- Errores por Nivel del Lenguaje
 - Léxico
 - Sintáctico
 - Semántico
 - Pragmático

Tareas para la próxima clase

1. Entrega próxima clase, 13:00hs último commit.

Términos de la clase #04

Definir cada término con la bibliografía

- Traductores
 - Ensambladores
 - Intérpretes
 - Compiladores
 - Máquinas Virtuales
- Las Fases de Traducción y los Tipos de Errores: Trabajo #1
 - Preprocesador
 - Compilador
 - Esnamblador
 - Vinculador
 - Opciones del compilador para cada fase
 - Errores por Tiempo
 - Errores por Fase de Traducción
 - Errores por Nivel del Lenguaje
- Análisis y Síntesis de hello.c
 - main
 - printf
 - Función del #include
 - puts
 - Secuencia de escape
 - Terminador de sentencia de C
 - Separador de sentencias de Pascal
 - Archivo .h (Encabezado)
 - String literal: Cadena Literal, o Literal de cadena, o Constante cadena, o Cadena constante
 - Lenguaje Ensamblador
 - Stack
 - Estructura lineal de un programa C
 - Estructura jerárquica de un programa Pascal.



¿Consultas?



Fin de la clase