

# Clase #03 de 27

# Introducción al Proceso de

# Compilación

*Abril 1ro, Lunes*

# Agenda para esta clase

- “El Lenguaje de Programación C” aka “El Libro Blanco” aka “K&R”
- El Proceso de Compilación, versión simplificada
- Otras versiones de hello.c
- El Preprocesador
- Intervalo
- Sintaxis, Semántica, y Pragmática
- Repaso Trabajo #0
- Repaso AED: Implementaciones de Stack
- Repaso AED: Organización de Memoria
- Repaso ADC: Registros IP, BP, y SP

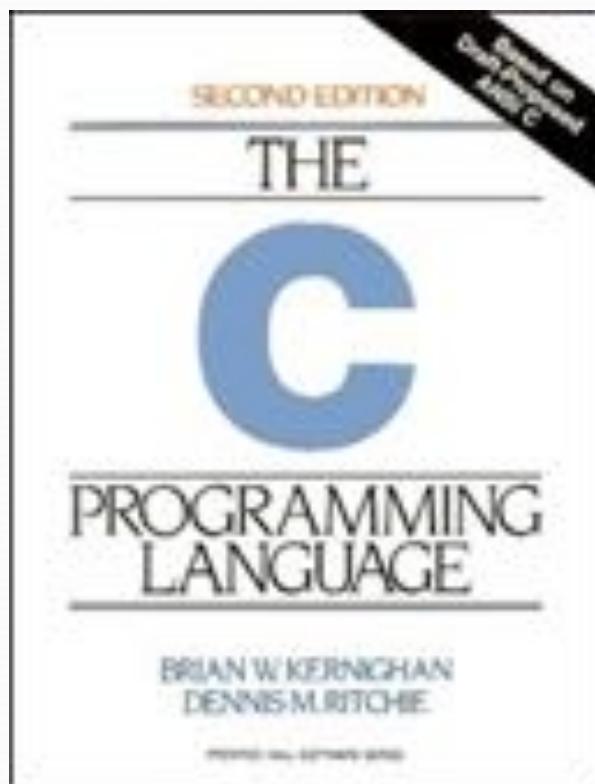
# K&R – Capítulo 1

K&R 1.1 Una Introducción Tipo Tutorial – Empezando

# “El Lenguaje de Programación C”

## aka “El Libro Blanco” aka “K&R”

The C Programming Language, 2nd Edition										
Prefacios e Introducción	#1 Tutorial							Apéndice A Definición del Lenguaje	Apéndice B La Biblioteca Estándar	Apéndice C Resumen de Cambios
	#2 Tipos y Expresiones	#3 Control de Flujo	#4 Funciones y Programa	#5 Punteros y Arreglos	#6 Estructuras	#7 Entrada y Salida	#8 Interfaz con Unix Ejemplos de implementación de parte de la Biblioteca			



- Autores:
  - Dennis Ritchie autor del LP y coautor de Unix con Ken Thompson
  - Kernighan
- Requisitos
  - Conocimiento de programación
  - Lectura atenta
- Ejercicios
- Ediciones y usos
  - 1978 1era Edición
    - Manual de referencia
  - 1988 2da Edición
    - ANSI C
- Preliminares (Front Matter)
  - Prefacio
  - Prefacio de la primera edición
  - Introducción
- General, “Ancho”
  - Capítulo 1: Tutorial
- En “profundidad”: Capítulos 2 a 7
  - 2 Tipos, Operadores y Expresiones
  - 3 Control de Flujo
  - 4 Funciones y Estructura de Programa
  - 5 Punteros y Arreglos
  - 6 Estructuras
  - 7 Entrada y Salida
- Ejemplo Integrador
  - 8 Interfaz de Sistema de UNIX – entrada/salida, sistema de archivos y alocación de memoria
- “Apéndices”
  - A: Especificación del LP: Semántica (LN) y Sintaxis (BNF)
  - B: Biblioteca estándar
  - C: Cambios introducidos en la primera versión del estándar.

# "Hello, World!" – Demostración de Compilación y Ejecución

```
#include <stdio.h>
main( ) {
    printf("Hello, world!\n");
}
```

```
$ cc hello.c
```

```
$ ./hello
Hello, world!
```

# Ejercicios

- 1-1. Experimentar con eliminación de partes y compilar
- 1-2. Probar diferentes \c.

# Otras Versiones de Hello.c

# Diferencias con Estándar C (y con C++)

```
main( ){
    puts("Pre Ansi");
}
```

```
#include <stdio.h>

int main(void){
    puts("Post Ansi");
    return 0;
}
```

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

int main(void){
    puts("Post Ansi");
    return EXIT_SUCCESS;
}
```

```
#include <stdio.h>

int main(void){
    puts("Post Ansi");
}
```

```
#include <iostream>

int main() {
    std::cout << "ANSI C++\n";
}
```

# El Proceso de Compilación, Versión Simplificada

# Hello World

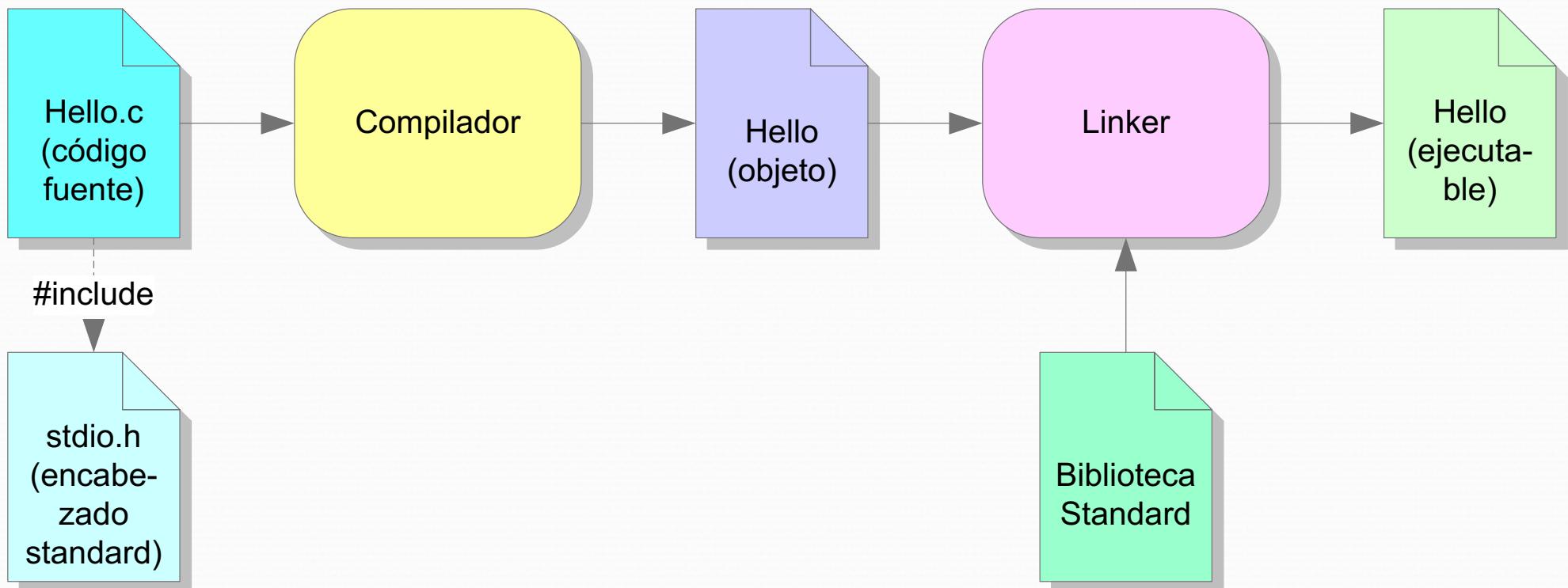
```
/* Hello world
 * JMS
 * 20150402
 */
#include <stdio.h>

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

# Tiempos

- Tiempo de diseño
- Tiempo de traducción (compilación)
  1. Tiempo de Preprocesamiento
  2. Tiempo de Compilación
  3. Tiempo de Vinculación
- Tiempo de ejecución
- Comparaciones
  - Contrastar con Máquina Virtual y Bytecode
  - Contrastar con Lenguajes Interpretados

# El Proceso de Compilación, Versión Simplificada



# El Preprocesador

# Dos Funciones Básicas del Preprocesador

```
#include <_____ .h>
return/*entre*/0;
return0;
return 0;
```

- Incluir archivos, directiva #include
- Reemplazar comentarios.

# Intervalo

## 20 minutos

# Sintaxis, Semántica, y Pragmática

# Otras versiones – ¿Mismas Sintaxis, Semántica, y Pragmática?

```
#include <stdio.h>
main( ) {
    printf( "Hello, world!\n" );
}
```

```
#include <stdio.h>
main( ){
    printf( "Hello,"
            "world!"
            "\n"      );
}
```

```
#include <stdio.h>
main( ){
    printf( "Hello," );
    printf( " world!" );
    printf( "\n"      );
}
```

```
#include <stdio.h>
main( ){
    puts( "Hello, world!" );
}
```

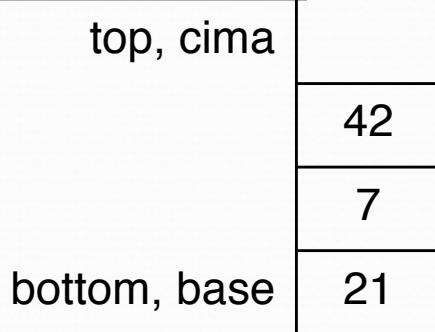
# Repaso Trabajo #0

# Repaso de Algoritmos y Estructura de Datos

Implementaciones de Stack

# Representaciones de Stack: Contigua y Enlazada

```
Stack s;  
Push(s,21);  
Push(s,7);  
Push(s,42);
```



```
// Contigua  
struct Stack{  
    unsigned n{0};  
    std::array<int,MAX> a;  
};
```

```
// Enlazada  
struct Node{  
    int val;  
    Node* next;  
};  
  
struct Stack{  
    Node* top=nullptr;  
};
```

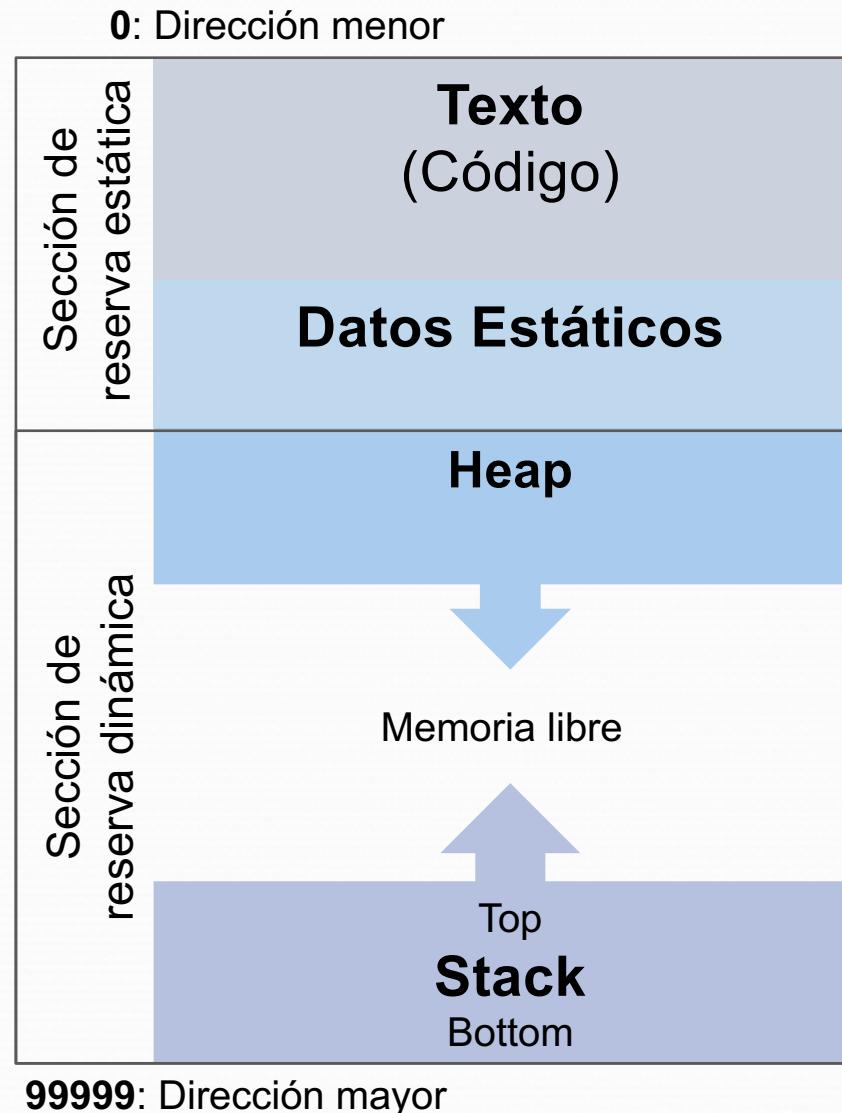
n=3
a[0]=21
a[1]=7
a[2]=42
a[3]=13
a[4]=51
...
a[MAX-1]=2



# **Repaso de Algoritmos y Estructura de Datos**

Organización de Memoria

# Organización de Memoria



# Repaso de Arquitectura de Computadoras

Pila de Invocaciones & Registros IP, BP, y SP

# Call Stack y Registros

- Registros
  - IP
  - BP
  - SP
- Operaciones
  - CALL f
    - PUSH IP
    - JMP f
  - RET
    - POP IP
- Stack
  - PUSH src
    - Mem[--SP] = src
  - POP dst
    - dst = Mem[SP++]
  - ADD
  - SUB
- Call Stack
- Stack Frame
  - BP-SP

The screenshot shows a debugger interface with two panes. The left pane displays the C++ source code for 'Hello.cpp' with syntax highlighting. The right pane shows the generated assembly code for the 'main' function, which includes a call to 'printf' and a string literal. The assembly code is color-coded by register (rbp, rsp, rdi, al, ecx, eax, etc.) and instruction type.

Line	Assembly Instruction	Description
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"	
17		

# Términos de la clase #03

## Definir cada término con la bibliografía

- K&R Capítulo #1
  - “El Lenguaje de Programación C” aka “El Libro Blanco” aka “K&R”
- Otras versiones de Hello.c
  - Tipo int implícito
  - Valor returned por main
  - Concatenación de cadenas
  - Múltiples invocaciones
  - puts
  - EXIT\_FAILURE
  - stdlib.h
- Proceso de Compilación, Versión Simplificada
  - Fuente
  - Compilador
  - Ejecutable
  - Comentarios mínimos en archivos fuente
  - Tiempo de diseño
  - Tiempo de traducción
  - Tiempo de Preprocesamiento
  - Tiempo de Compilación
- Tiempo de Vinculación (linker)
- Biblioteca estándar
- Tiempo de ejecución
- Contraste con Framework
- Función del compilador
- Vinculador (Linker)
- Contraste con Máquina virtual
- Bytecode
- Preprocesador
  - #include
  - Tratamiento de comentarios por parte del preprocesador
- Sintaxis, Semántica, y Pragmática
  - Sintaxis
  - Semántica
  - Pragmática
- Repaso AED: Implementaciones de Stack
  - Implementación Contigua de Stack
- Implementación Enlazada de Stack
- Repaso AED: Organización de Memoria
  - Reserva estática
  - Texto
  - Variables estáticas
  - Reserva dinámica
  - Stack
  - Heap
- Repaso ADC: Registros IP, BP, y SP
  - IP
  - BP
  - SP
  - CALL f
  - RET
  - PUSH src
  - POP dst
  - ADD
  - SUB
  - Call Stack
  - Stack Frame

# Tareas para la próxima clase

1. Repasar de AED: Representaciones de Stack: Contigua y Enlazada
2. Repasar de AED: Organización de Memoria: Text, Heap, Stack
3. Repasar de ADC: Call Stack y Registros
4. Conseguir [MUCH2012] volumen 1

# ¿Consultas?



# Fin de la clase