

Clase #03 de 27

Introducción a ANSI C y al Proceso de Compilación

Marzo 28, Lunes

Agenda para esta clase

- Conformación de Equipos
- Introducción al Lenguaje de Programación C
- “El Lenguaje de Programación C” aka “El Libro Blanco” aka “K&R”
- El Proceso de Compilación, versión simplificada
- El Preprocesador
- Intervalo
- Análisis y Síntesis de Hello.c
- Autómata Finito para Identificadores
- Sintaxis, Semántica, y Pragmática
- Versiones del Estándar

Conformación de Equipos

1. Cinco integrantes como mínimo
2. Por equipo, enviar un mail a
utnfrbassl-owner@yahoo.com
con los apellidos, nombres y usuario github de cada integrante.

Introducción al Lenguaje de Programación C

Descripción general_

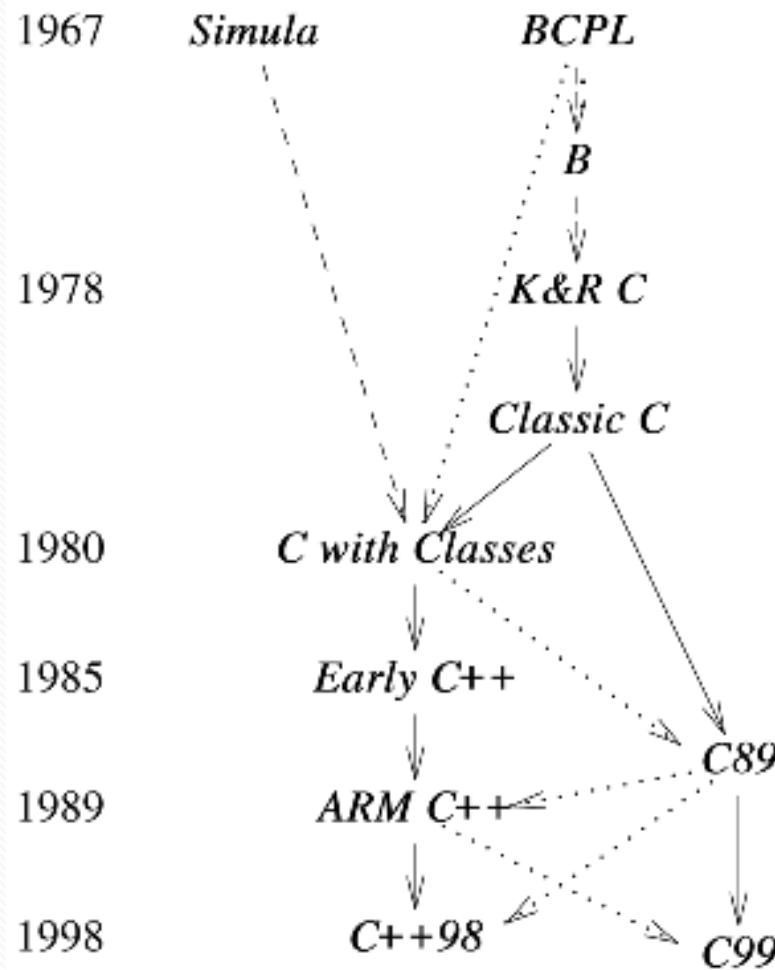
- LP de propósito general, no está especializado
- Economía en las expresiones, pero expresivo
- Control de flujo (¿de qué?)
- Estructuras de datos
- Gran cantidad de operadores
- Flexible
- No es de muy alto nivel (¿de qué?)
- No es grande (¿en qué sentido?)
- Su falta de restricciones y su generalidad lo hacen efectivo
- Independiente de máquina, portable (procesador y sistema operativo).

Frases sobre C_

- C no es un LP grande, y no le queda bien un libro grande (K&R)
- C tiene vueltas, falencias y un enorme éxito (Ritchie)
- C es un arma filosa, con la se pueden hacer programas eficientes y elegantes o una “carnicería” (Pike)
- C mejora a medida que uno gana experiencia con C (K&R) (Curva de aprendizaje empinada).

Evolución de C y LP relacionados_

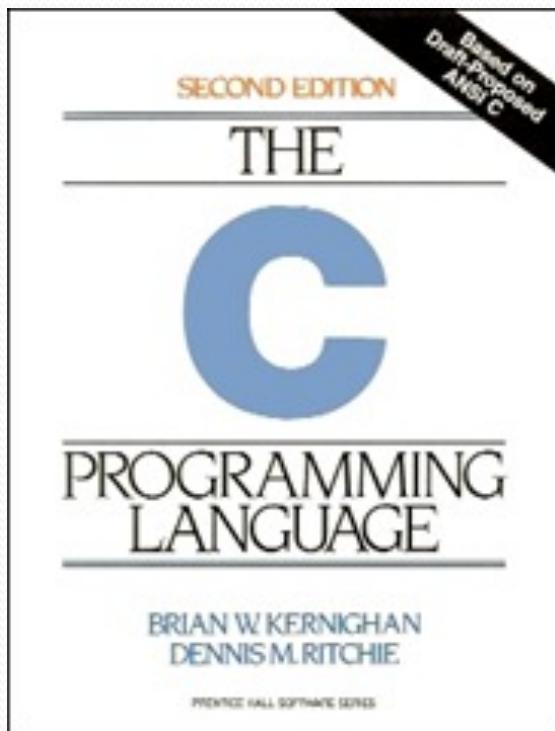
- Otros LP
 - C++
 - Objective-C
 - C#
 - D
 - Java



- 1966-1973
 - Elaboración. Basado en B, a su vez, basado en BCPL
 - C Pre estandarización
- 1978
 - K&R 1era edición
- 1983
 - Comienza estandarización
- 1988
 - K&R 2da edición
- 1989-90
 - C89-C90 (1era versión estándar)
- 1999
 - C99 (2da versión estándar)
- 2011
 - C11 (3era versión estándar)

“El Lenguaje de Programación C”

aka “El Libro Blanco” aka “K&R”



- 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.

K&R – Capítulo 1

K&R 1.1 Una Introducción Tipo Tutorial – Empezando

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

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

```
> cc hello.c
```

```
> hello.exe
Hello, world!
```



hello.c

Ejercicios

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

El Proceso de Compilación, Versión Simplificada

Hello World

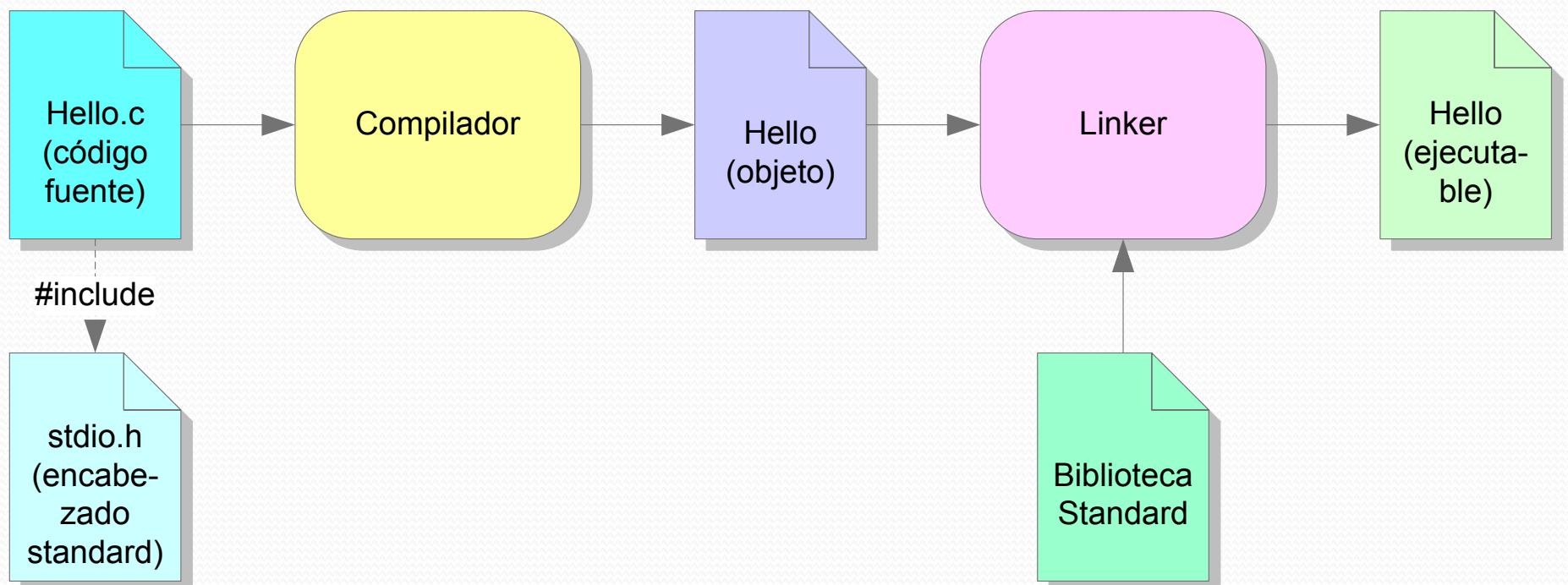
```
/* Hello world
 * JMS
 * 20150402
 */
#include <stdio.h>

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
- Comparacione
 - Contrastar con Máquina Virtual y Bytecode
 - Constrastar 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.

Análisis y Síntesis de Hello.c

Análisis de ‘Hello, World!’

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

Incluye información acerca de la biblioteca estándar

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

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

Generalización de ‘Hello, World!’

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

Programa Hello

- 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".

Autómata para Identificadores

Componentes de un Autómata Finito

- Estados
 - Inicial
 - Finales
- Transición
 - Origen
 - Evento
 - Destino
 - Transición reflexiva
- Alfabeto
- Ejemplo: Identificadores

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

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

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

Versiones del Estándar

Diferencias con ANSI 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";  
}
```

Tareas para la próxima clase

1. Conformar Equipos
2. Resolver [K&R1988] 1-1 a 1-3
3. Estudiar [K&R1988]
 - 1.4 Constantes Simbólicas
 - 1.5 Entrada y Salida de a Caracteres

Términos de la clase #03

Definir cada término con la bibliografía

- Parte I
 - Nivel de abstracción
 - Independiente de Máquina
 - Control de flujo de ejecución
 - Historia de C y de ANSI C
 - Relación entre C y C++
 - ANSI C, C89 ó C90
 - C99
 - C11
- Parte II
 - “El Lenguaje de Programación C” aka “El Libro Blanco” aka “K&R”
- Parte II
 - Fuente
 - Compilador
 - Ejecutable
- Parte III
 - 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
- Parte IV
 - #include
 - Tratamiento de comentarios por parte del preprocesador
- Parte V
 - main
 - printf
 - Función del #include
 - puts
 - Secuencia de escape
 - Terminador de sentencia de C
 - Separador de sentencias de Pascal
 - Estructura lineal de un programa C
 - Estructura jerárquica de un programa Pascal
 - Archivo .h (Encabezado)
 - String literal: Cadena Literal, o Literal de cadena, o Constante cadena, o Cadena constante
- Parte VI
 - Identificadores
 - Autómata Finito
 - Estados
 - Inicial
- Finales
- Transición
 - Origen
 - Evento
 - Destino
 - Transición reflexiva
- Alfabeto
- Parte VII
 - Comparación Sintaxis, Semántica & Pragmática
- Parte VIII
 - puts
 - EXIT_SUCCESS



¿Consultas?



Fin de la clase