

95.11 Algoritmos y Programación I (Electrónica)

Proceso de compilación

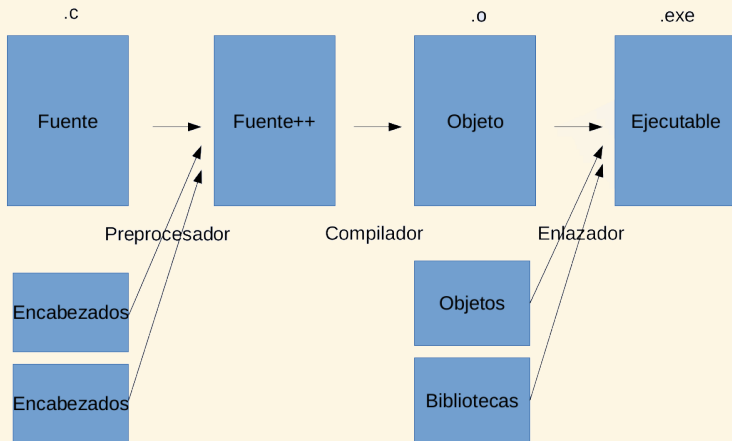
Sebastián Santisi

21 de marzo de 2019

Compilador

- ▶ El *compilador* es el programa que traduce de código fuente a un ejecutable
- ▶ El proceso de compilación en C consta de varias etapas:
 - Preproceso: Durante el preproceso se realizan, mayormente, reemplazos y se seleccionan fragmentos del código
 - Compilación: Durante la compilación se traducen los fuentes a código máquina o algún lenguaje intermedio
 - Enlace: Durante el enlace se integran los diferentes fragmentos compilados para formar un único ejecutable
- ▶ El compilador trabaja siempre de arriba hacia abajo

Fases de la compilación



Archivos

- .c: Fuentes del lenguaje C. Contienen la implementación.
- .h: Encabezados del lenguaje H. Contienen declaraciones, prototipos, constantes, tipos, etc.
- .o: Códigos objeto. Son fragmentos ya compilados de diferentes funciones.
- .lib, .a, .dll, etc.: Bibliotecas. Son fragmentos ya compilados de diferentes funciones, generalmente las desarrolló un tercero.
- “.exe”: Son el ejecutable completo.

Hola mundo

```
#include <stdio.h>

// Implementa el hola mundo en C.

int main(void) {
    printf("Hola mundo\n");
    return 0;
}
```

Desmenuzando el hola mundo

- ▶ Las sentencias al preprocesador comienzan con `#`
- ▶ `#include` trae un archivo de encabezados que contiene la “documentación” de las funciones, tipos, etc.
- ▶ `<stdio.h>` standard input/output, encabezados de entrada-salida
- ▶ `main()` función punto de entrada
- ▶ `printf()` función para imprimir
- ▶ `return 0` al sistema operativo hay que devolverle algo (convención: si devuelvo 0 todo estuvo bien)
- ▶ `// ... comentario hasta el final de la línea` (`/* ... */` comentario multilinea)

Compilemos el hola mundo

Compilemos el hola mundo

```
$ ls
hola.c
$ gcc -Wall -std=c99 -pedantic hola.c -o hola
$ ls
hola hola.c
$ ./hola
Hola mundo
$
```

gcc Compilador

-Wall Todas las advertencias

-std=c99 Estándar ISO-C99

-pedantic Poner al compilador en modo
“pedante”

hola.c El fuente que compilamos

-o hola El nombre del ejecutable de salida

Constantes y etiquetas

- ▶ Las “variables” pueden ser “constantes”

```
const float pi = 3.141593;
```

- ▶ Es una variable como las demás pero o puede cambiarse su valor a posteriori
- ▶ También se pueden crear etiquetas

```
#define PI 3.141593
```

- ▶ Directiva del procesador
- ▶ Hace copy-paste de la expresión donde sea que la encuentre
- ▶ No consume memoria, no tiene un tipo asociado (más allá del del literal), etc.
- ▶ Convención de nombres: MAYUSCULAS