



## Introducción

### Seminario de Lenguajes opción C

#### Características del lenguaje

Función main  
Operadores y tipos

#### Compilación de programas

Preprocesador  
Compilador  
Ensamblador y  
link-editor

#### Herramientas

GCC  
Editores e IDEs  
Terminal

#### Resumen

# Introducción

## Herramientas y compilación

### Seminario de Lenguajes opción C

Facultad de Informática  
Universidad Nacional de la Plata

2022



# Índice

## Introducción

## Seminario de Lenguajes opción C

## Características del lenguaje

Función main

Operadores y tipos

## Compilación de programas

Preprocesador

Compilador

Ensamblador y  
link-editor

## Herramientas

GCC

Editores e IDEs

Terminal

## Resumen

### 1 Características del lenguaje

Función main

Operadores y tipos

### 2 Compilación de programas

Preprocesador

Compilador

Ensamblador y link-editor

### 3 Herramientas

GCC

Editores e IDEs

Terminal

### 4 Resumen



# Estructura mínima de un programa

## Introducción

### Seminario de Lenguajes opción C

#### Características del lenguaje

##### Función main

##### Operadores y tipos

#### Compilación de programas

##### Preprocesador

##### Compilador

##### Ensamblador y link-editor

#### Herramientas

##### GCC

##### Editores e IDEs

##### Terminal

#### Resumen

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

```
int main(int argc, char *argv[]) {  
    printf("Hola\n");  
    return 0;  
}
```

- Se distinguen mayúsculas y minúsculas.
- La función `main()` tiene el código principal.
- Un programa debe tener exactamente una función `main`.
- `main()` retorna un entero.
- Por convención `main()` retorna 0 si no hay errores.
- Otras funciones deben declararse antes del `main`.



# Tipos de datos simples

## Introducción

### Seminario de Lenguajes opción C

### Características del lenguaje

Función main

Operadores y tipos

### Compilación de programas

Preprocesador

Compilador

Ensamblador y  
link-editor

### Herramientas

GCC

Editores e IDEs

Terminal

### Resumen

- `int` → enteros.
- `short` → enteros (puede tener menos bytes que un `int`)
- `long` → enteros (puede tener más bytes que un `int`)
- `long long` → enteros (puede tener más bytes que un `long`)
- `char` → caracteres, 1 byte cada uno (K&R).
- `float` → punto flotante.
- `double` → punto flotante doble precisión.
- `long double` → punto flotante cuádruple precisión.
- No hay booleanos: cualquier entero **distinto** a 0 es TRUE.
- Punteros (los vamos a ver más adelante).



# Característica interesantes del lenguaje

Pre y pos incremento/decremento

## Introducción

### Seminario de Lenguajes opción C

### Características del lenguaje

Función main

Operadores y tipos

### Compilación de programas

Preprocesador

Compilador

Ensamblador y  
link-editor

### Herramientas

gcc

Editores e IDEs

Terminal

### Resumen

```
int a, b, c, d;  
a = 5;  
b = 2;  
b--;  
c = ++a;  
d = b++;
```

- $a \leftarrow 5$
- $b \leftarrow 2$
- $b \leftarrow b - 1$
- $a \leftarrow a + 1; \quad c \leftarrow a$
- $d \leftarrow b; \quad b \leftarrow b + 1$



# Característica interesantes del lenguaje

Tipado débil (conversiones automáticas)

## Introducción

### Seminario de Lenguajes opción C

#### Características del lenguaje

Función main

Operadores y tipos

#### Compilación de programas

Preprocesador

Compilador

Ensamblador y link-editor

#### Herramientas

gcc

Editores e IDEs

Terminal

#### Resumen

```
int a; // Asumamos que sizeof(int) == 4
short int b; // Asumamos que sizeof(short int) == 2
a = 100000;
b = a; // ¿Cuanto vale b?
```

- $a \leftarrow 0000000000000000011000011010100000_2$
- $b \leftarrow 1000011010100000_2$
- Eso es  $34464_{10}$ , pero...
- Son enteros con signo en complemento 2
- Por lo tanto  $b \leftarrow -31072_{10}$



# Característica interesantes del lenguaje

## Conversiones explícitas (castings)

### Introducción

### Seminario de Lenguajes opción C

### Características del lenguaje

Función main

Operadores y tipos

### Compilación de programas

Preprocesador

Compilador

Ensamblador y  
link-editor

### Herramientas

gcc

Editores e IDEs

Terminal

### Resumen

```
float a = 3.5;  
float b;  
int c;  
char d;  
b = (int) a;  
c = a;  
d = a;
```

- $a \leftarrow 3.5$
- $b \leftarrow 3$
- $c \leftarrow 3$
- $d \leftarrow 3$



# Característica interesantes del lenguaje

char es un tipo entero

## Introducción

## Seminario de Lenguajes opción C

## Características del lenguaje

Función main

Operadores y tipos

## Compilación de programas

Preprocesador

Compilador

Ensamblador y  
link-editor

## Herramientas

GCC

Editores e IDEs

Terminal

## Resumen

```
char x = 'a';  
x++;  
x = x - 32;
```

- $x \leftarrow 97_{10}$  (ASCII de 'a')
- $x \leftarrow 98_{10}$  (ASCII de 'b')
- $x \leftarrow 66_{10}$  (ASCII de 'B')

¿Es minúscula?

```
if (x >= 'a' && x <= 'z') { // ...
```

Investigar las funciones de la familia `islower()` con el comando:

```
$ man islower
```





# Característica interesantes del lenguaje

## Operadores bit a bit

### Introducción

### Seminario de Lenguajes opción C

### Características del lenguaje

### Función main

### Operadores y tipos

### Compilación de programas

### Preprocesador

### Compilador

### Ensamblador y link-editor

### Herramientas

### GCC

### Editores e IDEs

### Terminal

### Resumen

```
unsigned int a; // Suponemos sizeof(unsigned) == 4
a = ~0;         // not
a = 1 & 5        // and
a = 1 | 4        // or
a = 1 ^ 3        // xor
a = 1 << 3       // shift a izquierda
a = 8 >> 1       // shift a derecha
```

- $\sim 0 = 0xFFFFFFFF$
- $001_2 \wedge 101_2 = 001_2$
- $001_2 \vee 100_2 = 101_2$
- $001_2 \oplus 011_2 = 010_2$
- $0001_2 \ll 3 = 1000_2$
- $1000_2 \gg 1 = 0100_2$

No confundir con los operadores lógicos: `&&`, `||`, `!`.



# Estructuras de control elementales

## Introducción

### Seminario de Lenguajes opción C

#### Características del lenguaje

Función main

Operadores y tipos

#### Compilación de programas

Preprocesador

Compilador

Ensamblador y  
link-editor

#### Herramientas

gcc

Editores e IDEs

Terminal

#### Resumen

```
if (condición) {  
    // si condición != 0  
}  
else {  
    // si condición == 0  
}  
  
while (condición) { } // Mientras condición != 0  
  
for (inicialización; condición; incremento) { }
```

Ver ejemplos.



# Operadores lógicos y comparaciones

## Introducción

## Seminario de Lenguajes opción C

## Características del lenguaje

Función main

## Operadores y tipos

## Compilación de programas

Preprocesador

Compilador

Ensamblador y  
link-editor

## Herramientas

GCC

Editores e IDEs

Terminal

## Resumen

- `&&` AND lógico.
- `||` OR lógico.
- `!` NOT lógico.
- `==` igualdad.
- `!=` desigualdad.
- `<` y `>` menor y mayor.
- `<=` y `>=` menor-o-igual y mayor-o-igual.



# Proceso de compilación

## Introducción

## Seminario de Lenguajes opción C

## Características del lenguaje

Función main  
Operadores y tipos

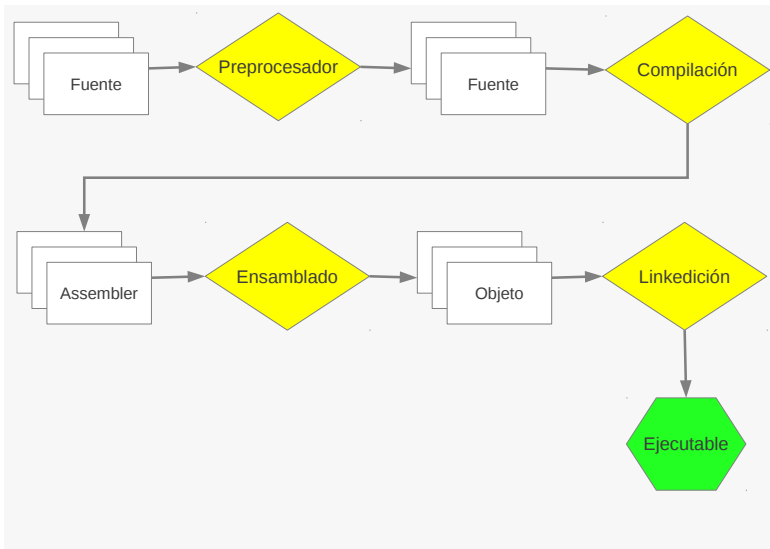
## Compilación de programas

Preprocesador  
Compilador  
Ensamblador y link-editor

## Herramientas

GCC  
Editores e IDEs  
Terminal

## Resumen





# Preprocesador

## Introducción

### Seminario de Lenguajes opción C

#### Características del lenguaje

Función main  
Operadores y tipos

#### Compilación de programas

##### Preprocesador

Compilador  
Ensamblador y  
link-editor

#### Herramientas

GCC  
Editores e IDEs  
Terminal

#### Resumen

- Transforma el código.
- Substituye texto.
- Es independiente del lenguaje C pero indispensable.
- Procesa las directivas que empiezan con #.
- Ejemplos:

```
#include <stdio.h>
#include "ejemplo.h"
#define test 5
```



# Compilador

## Introducción

### Seminario de Lenguajes opción C

#### Características del lenguaje

Función main

Operadores y tipos

#### Compilación de programas

Preprocesador

**Compilador**

Ensamblador y  
link-editor

#### Herramientas

GCC

Editores e IDEs

Terminal

#### Resumen

- Verifica la sintaxis del programa.
- Advierte sobre potenciales errores semánticos.
- Transforma el código C en assembler.
- Puede optimizar el código generado.
- Puede agregar información extra para el debugger.



# Ensamblador y link-editor

## Introducción

### Seminario de Lenguajes opción C

### Características del lenguaje

### Función main Operadores y tipos

### Compilación de programas

### Preprocesador

### Compilador

### Ensamblador y link-editor

### Herramientas

### GCC

### Editores e IDEs

### Terminal

### Resumen

## Ensamblador:

- Traduce código assembler a código de máquina.
- El binario generado se denomina “código objeto”.

## Link-editor

- Genera un archivo ejecutable a partir de uno o más códigos objeto.
- Se puede dividir el programa en módulos y compilar cada uno por separado, al modificar un módulo sólo hay que compilar ese módulo y volver a “linkearlo” con los otros códigos objeto.
- El proceso anterior ahorra tiempo en proyectos grandes.



## Introducción

### Seminario de Lenguajes opción C

#### Características del lenguaje

Función main  
Operadores y tipos

#### Compilación de programas

Preprocesador  
Compilador  
Ensamblador y  
link-editor

#### Herramientas gcc

Editores e IDEs  
Terminal

#### Resumen

El “GNU C Compiler”, además de compilar, puede invocar a las otras herramientas. Invocando a GCC se puede hacer una sola etapa de las anteriores, o todas las etapas de una vez.

Argumentos de gcc:

- -E → Preprocesa.
- -S → Compila (genera código assembler)
- -c → Ensambla (genera código objeto).
- -o → Permite indicar el nombre del archivo de salida.
- Sin argumentos realiza todos los pasos.
- -std=c99 → Estándar C usado.
- -Wall → Muestra las advertencias.

Ejemplos





# Entornos de desarrollo

## Introducción

## Seminario de Lenguajes opción C

## Características del lenguaje

Función main  
Operadores y tipos

## Compilación de programas

Preprocesador  
Compilador  
Ensamblador y  
link-editor

## Herramientas

GCC  
Editores e IDEs  
Terminal

## Resumen

- Existen IDEs y editores para todos los gustos.
- La cátedra no fuerza el uso de ninguna de estas opciones, pero es obligatorio saber compilar con gcc desde la terminal.



# Terminal/Línea de Comando

## Introducción

### Seminario de Lenguajes opción C

### Características del lenguaje

Función main

Operadores y tipos

### Compilación de programas

Preprocesador

Compilador

Ensamblador y  
link-editor

### Herramientas

GCC

Editores e IDEs

**Terminal**

### Resumen

Comandos básicos (dentro de Cygwin es lo mismo):

- `pwd` → Directorio actual.
- `ls` → Muestra el contenido de un directorio.
- `cat archivo` → Muestra el contenido del archivo.
- `cd directorio` → Se posiciona en el directorio indicado.
- `cd ..` → Vuelve al directorio padre.
- `manpages` o `http://man.cx`
- `man 3 printf`
- `man ls` , `man gcc`
- `programa_instalado`
- `./programa_sin_instalar`



# Resumen

## Introducción

### Seminario de Lenguajes opción C

#### Características del lenguaje

Función main

Operadores y tipos

#### Compilación de programas

Preprocesador

Compilador

Ensamblador y  
link-editor

#### Herramientas

GCC

Editores e IDEs

Terminal

## Resumen

- Qué hace el preprocesador /compilador/ensamblador/linker.
- Cómo invocarlos desde la terminal.
- ¿Es necesario aprender a usar algún IDE?.
- ¿Es necesario que aprenda a invocar gcc desde la terminal?
- ¿Cómo se escribe un programa básico?