

# Técnicas y Prácticas de Programación

## Lenguaje C - Parte Básica

Luis Garreta

[luis.garreta@javerianacali.edu.co](mailto:luis.garreta@javerianacali.edu.co)

Ingeniería de Sistemas y Computación  
Pontificia Universidad Javeriana – Cali

26 de julio de 2017

# Primer Programa: hola.c

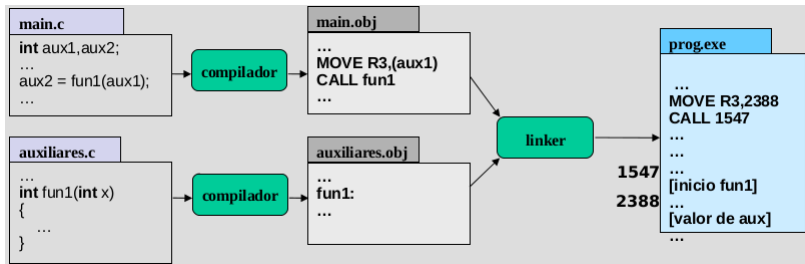
```
#include <stdio.h>

void main(void) {
    printf("Hola mundo!!\n");
}
```

## Segundo Programa: declaraciones.c

```
#include <stdio.h>
#define NUM 2
void main(void) {
    int a, b, suma;
    a = NUM;
    b = 2;
    suma = a + b;
    printf("suma vale %d", suma);
}
```

# Proceso de Compilación de un Programa en C



# Declaración / Definición de Variables y Funciones

- Declaración: especifica al compilador que un variable o función existe.

```
int i, j;  
int aux;  
int optimo(int a, int b);
```

- Definición: la variable o función misma.

```
int optimo(int a, int b){  
    int aux = a;  
    /* sigue... */  
    return aux;  
}
```

## Atributos de una Variable

Atributo	Descripción
Tipo	char, int, unsigned int, etc.
Nombre	Identificador para acceder el objeto.
Valor	Datos contenidos en el objeto.
Dirección	La ubicación en memoria donde reside el objeto.
Alcance	El código fuente donde el nombre del objeto es reconocido.
Tiempo de vida	Cuando el objeto está disponible (dependiendo de cuando el objeto es creado y destruido).

# Tipos de Variables

- Tipo de variables fundamentales:
  - enteros: `int`, `char`.
  - flotantes: `float`, `double`.
- Modificadores (adjetivo):
  - `short`, `long`, `unsigned`, `signed`.
  - No todas las combinaciones de tipos y modificadores son válidas.

## Tabla de Tipos de Variables

int	Numero Entero	int a=3;
float	Numero Punto Flotante	float a=4.3;
char	Un byte de memoria suficiente para guardar un caracter	char a;
short	Entero de tamaño reducido	short int i;
long	Entero de tamaño aumentado	long int i;
unsigned	Entero sin rango negativo, ➔ mayor rango positivo	unsigned int i;
double	Numero de punto flotante de doble precisión	double i;



# Tamaño de los Tipos de Variables

Tipo de dato		Tamaño	Rango
unsigned	char	8 bits	0 a 255
	short int	16 bits	0 a 65,535
	int	16 bits o 32 bits	Idem unsigned short int; Idem unsigned long int.
	long int	32 bits	0 a 4,294,967,295
signed	char	8 bits	-128 a +127
	short int	16 bits	-32,768 a +32,767
	int	16 bits o 32 bits	Idem signed short int; Idem signed long int.
	long int	32 bits	-2,147,483,648 a +2,147,483,647

- Atención:
  - int tamaño del bus del CPU.
  - short int menor o igual a int.
  - long int mayor o igual a int.

# Operadores de Asignación

- Operadores: =, +=, -=, \*=, /=  
– Ejemplos:

```
int i = 1;  
i += 2; //equivalente a: i = i + 2;  
i *= 10; //equivalente a: i = i * 10;
```

# Operadores Aritméticos

- Operadores: +, -, \*, /, %

– Ejemplos:

```
int a,b,n;  
a = 10;  
b = 3;  
n = a%b; // resultado: n = 1;
```

- Operadores pos, pre-incremento: ++, --

– Ejemplos:

```
int i = 10;  
i++; // equivalente a: i = i + 1;
```

# Operadores Relacionales

- Operadores: `==`, `!=`, `<`, `<=`, `>`, `>=`

– Ejemplo:

```
int x, y, b;  
x = 10;  
y = 3;  
b = (x==y); // resultado: b = 0;
```

– Observación:

- No existe el tipo boolean: resultado entero
  - false: `0`
  - true: `!0`
- `a = b` no es lo mismo que `a == b`.

# Operadores Lógicos

- Operadores: && (AND), || (OR), ! (NOT).

- Ejemplos:

```
int x, b;  
// sigue: x = ...  
b = (x < 0) && (x > 10);
```

- Observación:

- Expresiones con operadores lógicos pueden ser corto circuitadas (algunos compiladores). Por ejemplo: si  $x < 0$  es falso, toda la expresión será falsa.

# Formas de Control de Flujo

- Secuencia
  - {...}
- Condicionales
  - if-else, switch-case
- Iteraciones
  - while, do-while, for

# Condicionales **if-else**

- Condicionales: `if(cond)-else`,
  - Ejemplos:

```
int x, y;  
if (x < 0){  
    y = x;  
} else {  
    y = x*10;  
}
```

# Condicionales **switch-case**

- Condicionales: switch-case

– Ejemplo:

```
char c;  
//...  
if (c == 'a'){  
    // sentencias si 'a'  
} else if (c == 'b'){  
    // sentencias si 'b'  
} else if (c == 'c'){  
    // sentencias si 'c'  
} else {  
    // otras letras  
}
```

```
char c;  
//...  
switch ( c ) {  
    case 'a':  
        // sentencias si 'a'  
        break;  
    case 'b':  
        // sentencias si 'b'  
        break;  
    case 'c':  
        // sentencias si 'a'  
        break;  
    default:  
        // otras letras  
}
```



# Ciclos **while** y **do-while**

- Iteración: `while(cond)`, `do-while(cond)`
  - Ejemplos:

```
int cond;  
// cond = ...; Se determina si entra a la iterac.  
while (cond){  
    // sentencias a repetir y modificación de cond  
}  
  
int cond; // siempre se entra una vez a la iterac.  
do {  
    // sentencias a repetir y modificación de cond  
while (cond);
```

# Ciclos **for**

- Iteración: `for(ini;cond;modif)`

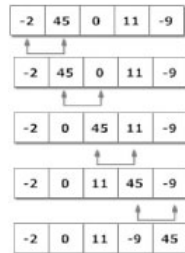
- Ejemplos:

```
int i;  
// cond = ...; Se determina si entra a la iterac.  
for(i = 0; i<MAX;i++){  
    // sentencias a repetir MAX veces.  
    // i puede ser utilizado como índice.  
}  
for(;;){  
    // bucle infinito.  
}
```

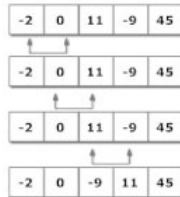
# Ejemplo Programa de Ordenamiento por Burbuja (BubbleSort)

Arreglo a ordenar

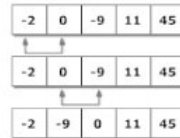
-2	45	0	11	-9
----	----	---	----	----



Step 1



Step 2



Step 3



Step 4

Figure: Working of Bubble sort algorithm