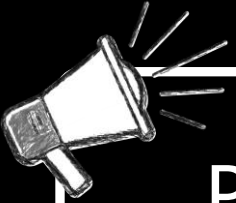


Introducción a la Programación

Introducción a la Programación

- Lenguajes de Programación
- Ciclo de Desarrollo
- Práctica “Hola Mundo”

Lenguajes de Programación



Para escribir un programa o aplicación,
necesitamos un “lenguaje de programación”

Lenguajes de Programación

Lenguaje Maquina

Es lo único que entiende el Procesador

1		00000000	00000100	000000000000000000
2	01011110	00001100	11000010	000000000000000010
3		11101111	00010110	000000000000000101
4		11101111	10011110	000000000000001011
5	11111000	10101101	11011111	00000000000010010
6		01100010	11011111	00000000000010101
7	11101111	00000010	11111011	00000000000010111
8	11110100	10101101	11011111	00000000000011110
9	00000011	10100010	11011111	00000000000100001
10	11101111	00000010	11111011	00000000000100100
11	01111110	11110100	10101101	
12	11111000	10101110	11000101	00000000000101011
13	00000110	10100010	11111011	00000000000110001
14	11101111	00000010	11111011	00000000000110100
15		01010000	11010100	00000000000111011
16			00000100	00000000000111101

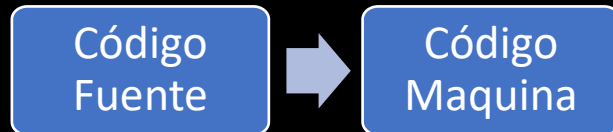
Lenguajes de Programación

Assembler

Código Fuente

```
1      entry    main, ^m<r2>
2      sub12    #12, sp
3      jsb      C$MAIN_ARGS
4      movab    $CHAR_STRING_CON
5
6      pushal   -8(fp)
7      pushal   (r2)
8      calls    #2, SCANF
9      pushal   -12(fp)
10     pushal   3(r2)
11     calls    #2, SCANF
12     mull3    -8(fp), -12(fp), -
13     pusha    6(r2)
14     calls    #2, PRINTF
15     clr1     r0
16     ret
```

Instrucciones

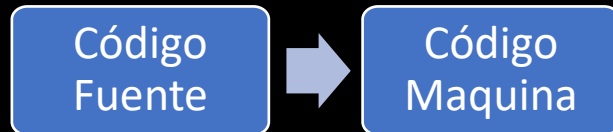


1 Sentencia = 1 Instrucción

Lenguajes de Programación

Lenguaje C

Código Fuente



```
1  /* This program reads two integers from the keyboard
2     and prints their product.
3     Written by:
4     Date:
5  */
6  #include <stdio.h>
7
8  int main (void)
9  {
10 // Local Definitions
11     int number1;
12     int number2;
13     int result;
14
15 // Statements
16     scanf ("%d", &number1);
17     scanf ("%d", &number2);
18     result = number1 * number2;
19     printf ("%d", result);
20     return 0;
21 } // main
```

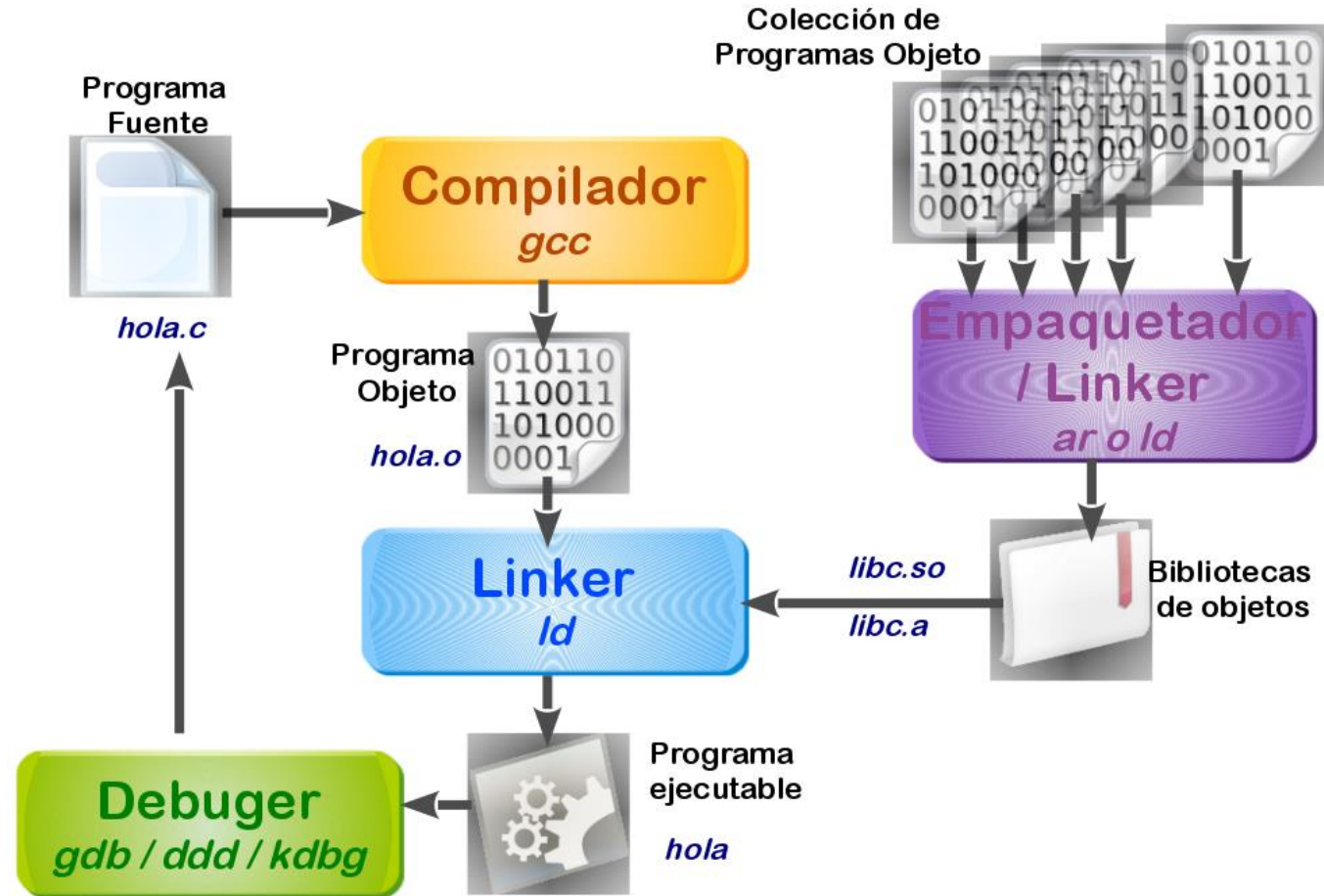
1 Sentencia = 1+ Instrucción/s

Ciclo de Desarrollo

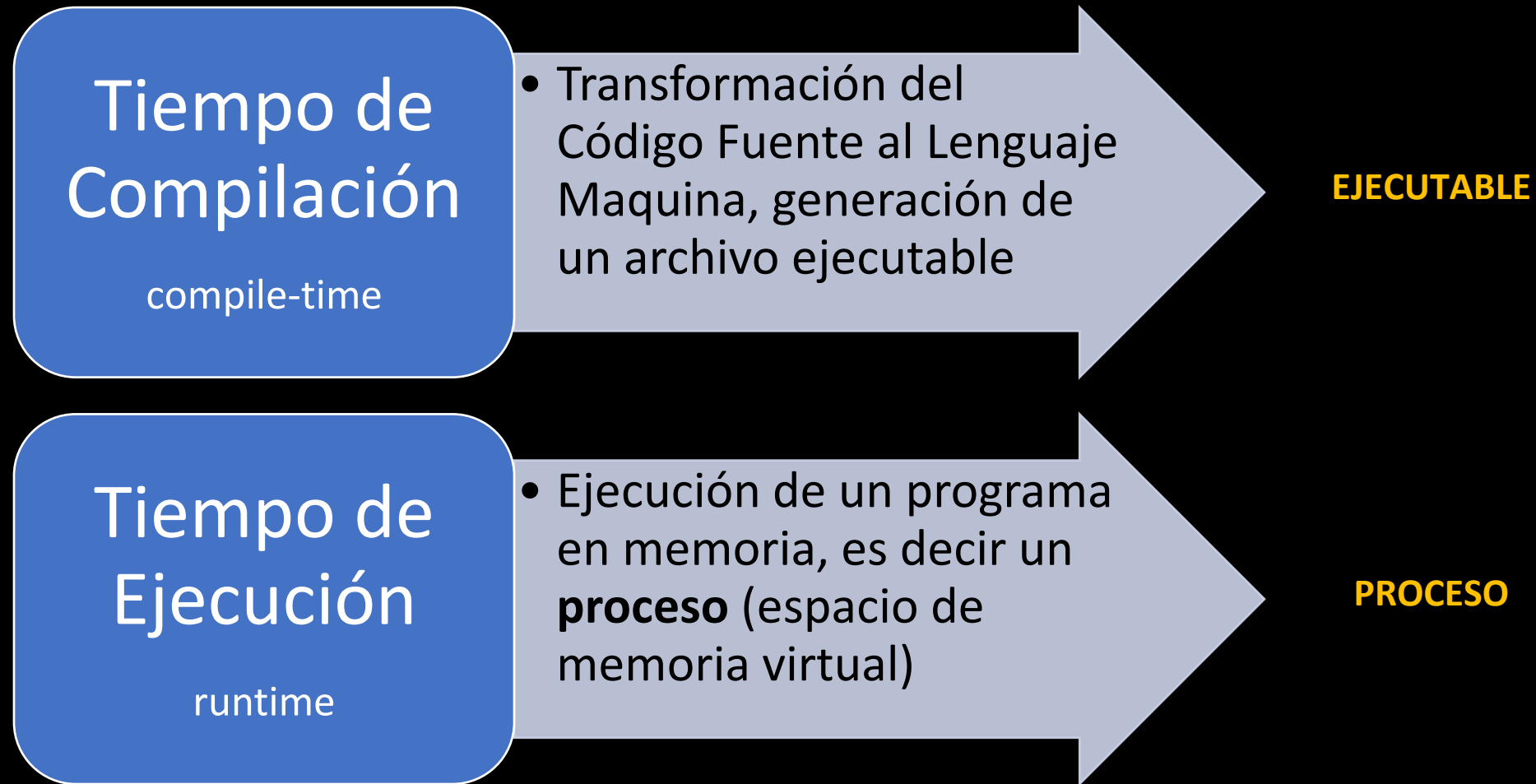


*) Lo dejamos para mas adelante

Ciclo de Desarrollo



Ciclo de Desarrollo



Práctica “Hola Mundo” 45’



- Ejemplo “Hola Mundo”

Práctica “Hola Mundo” 45’



1) EDITAR

Entry Point
(de “C”)

```
#include <stdio.h> //Archivo Cabecera de libc

int main (void) {
    printf( "Hola Mundo"); //Fin de Línea de instrucción;

    return 0;
}
```

Fin de Función

2) COMPILAR (por default libc)

```
$ gcc hola.c a.out

$ gcc -o hola hola.c hola (defino el nombre)
```

3) EJECUTAR

```
$ ./hola ejecutar el programa
```

Práctica “Hola Mundo” 45’



- Sintaxis?
- printf?
- Archivos de cabecera? ¿Donde están?
- ¿Libc? ¿Donde esta? ¿Para que?
- ¿main? ¿Una función? Especial...Sin prototipo y otras..
 - ¿Entry point? ¿Para que?
- int
- void
- {} ¿para que?
- ¿return 0? Para que?
- ¿Errores? En la compilación
- ¿Warnings? En la compilación
- Tipos de archivo (texto .c .h, ejecutable .o)
 - Comando ‘file’