

ENTORNOS DE DESARROLLO

UD2: LENGUAJES DE PROGRAMACIÓN

Práctica 1. Elaboración, compilación y ejecución. Tipos de Código

Objetivo

Establecer un primer contacto con herramientas de elaboración y compilación de programas en lenguaje C.

Analizar e identificar las características de los programas escritos en lenguajes de alto y bajo nivel.

Trabajo a realizar

1.- Elaboración del código fuente.

Utilizando un editor de GNU/Linux, crear un archivo de texto con el contenido mostrado en la siguiente imagen que corresponde a un programa escrito en C cuya salida por pantalla será "Hello World".

```
1  #include <stdio.h>
2
3  int main(int argc, char *arg[]){
4
5      printf("Hello world");
6
7  }
```

Por ejemplo, podemos usar el editor nano: abrir un terminal y teclear

```
sudo nano hello_world.c
```

2.- Compilar el programa simple escrito en C en GNU/Linux

Una vez codificado el programa en C, en algún editor de texto, debe ser

Práctica: Lenguajes Compilados

ENTORNOS DE DESARROLLO

UD2: LENGUAJES DE PROGRAMACIÓN

leído por **compilador**. El compilador es un programa que produce un archivo ejecutable y depende totalmente del hardware del ordenador y del sistema operativo que corre sobre él.

A veces el compilador arroja advertencias durante el proceso. A pesar de que se generará el archivo ejecutable puede producir problemas a la hora de ejecución por lo que será mejor investigar su causa. En caso de que se produzcan errores el ejecutable no se generará.

En nuestro caso haremos uso de GCC (GNU Compiler Collection).

GCC (GNU Compiler Collection) son una colección de compiladores de uso libre para sistemas operativos GNU/Linux que viene pre-instalada en la mayoría de distribuciones y existe una versión modificada para windows (MinGW).

Desde la línea de comandos y en el path de la práctica ejecutaremos la siguiente sentencia par compilar el programa que nos producirá un programa ejecutable

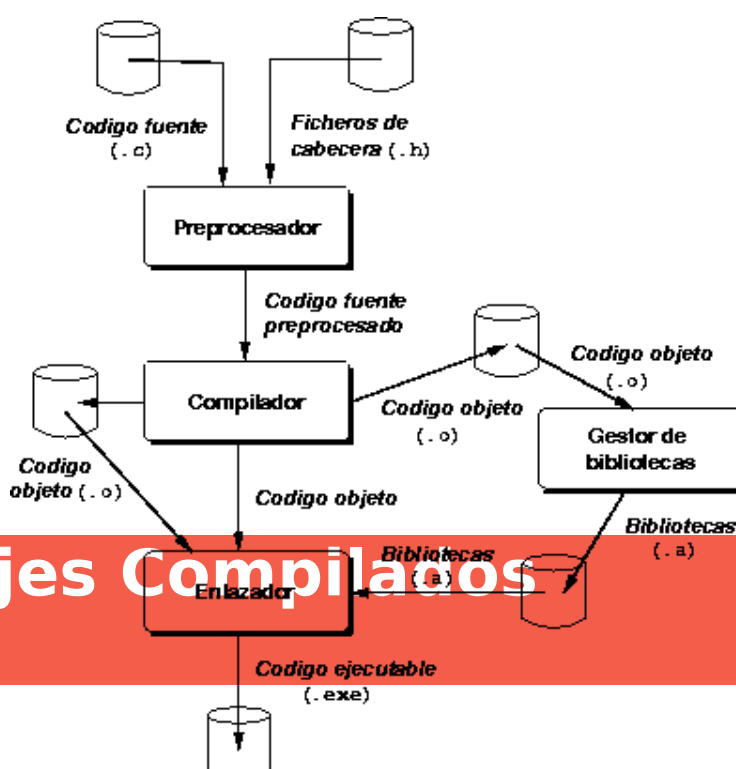
```
gcc hello_world.c -o hello_world.out
```

Una vez hecho esto, ejecutaremos el programa a través de la línea de comandos

```
./hello_world.out
```

3.- Archivos intermedios y fases de compilación

Durante la fase de compilación se han generado una serie de archivos intermedios que han sido borrados una vez se ha creado el ejecutable final. En este paso le diremos al pre-procesador, compilador y



Práctica: Lenguajes Compilados

ENTORNOS DE DESARROLLO

UD2: LENGUAJES DE PROGRAMACIÓN

enlazador que no queremos que los borre.

(Antes de nada borra el ejecutable creado en el paso anterior)

```
rm hello_world.out
```

y vuelve a compilar

```
gcc hello_world.c -o hello_world.out -save-temps
```

4. Inspecciona los archivos creados y completa la siguiente tabla

Fichero	Tipo de código (máquina/ensamblador/alto nivel)	Dependencia máquina	Legibilidad	Funcionalidad dentro del proceso de compilación	Otras características
hello_world.c	alto nivel	usuario	alta	código fuente	
hello_world.i	alto nivel	usuario	alta	código fuente procesado	
hello_world.s	ensamblador	maquina	media	código objeto	

Práctica: Lenguajes Compilados

ENTORNOS DE DESARROLLO

UD2: LENGUAJES DE PROGRAMACIÓN

hello_world.o	lenguaje maquina	maquina	baja	código objeto	
hello_world.o ut	alto nivel	usuario	alta	código ejecutable	

Práctica: Lenguajes Compilados