LICENCIATURA EN INFORMATICA

PROGRAMADOR UNIVERSITARIO

INGENIERÍA EN INFORMÁTICA

PROGRAMACIÓN

Unidad 2 - Parte 2: Etapas de compilación.

2023

Lic. Mariela A. Velázquez

Repasemos lo visto

Holamundo.c

```c
#include<stdio.h>

int main(){

    printf("Hola Mundo ");

    return 0;
}
```

Compilador

- El compilador para C del proyecto GNU se llama “gcc” y su nombre proviene de “GNU C Compiler”[1]

- Un compilador es un programa informático que traduce un programa de un lenguaje a otro, generando un lenguaje equivalente que la maquina será capaz de interpretar.

- Este proceso de traducción se conoce como compilación.

[1] Hoy en día “gcc” también quiere decir “GNU Collection Compiler” porque también representa una colección de compiladores

Código Fuente

- El compilador gcc es capaz de compilar cualquier programa en el lenguaje C escrito en un archivo de texto convencional.

- El código fuente escrito en un archivo de texto lleva el sufijo ".c" para identificar que su contenido corresponde al código de un programa escrito en el lenguaje C.

- También existen herramientas más especializadas para escribir y editar código fuente de manera más eficiente que un simple editor de texto.

Compilando un programa en C

```
$ gcc –Wall primerPrograma.c –o primerPrograma
```

El comando anterior compila el código fuente a código maquina y lo almacena el archivo ejecutable.

-Wall : opción para activar las advertencias del compilador . El compilador puede generar advertencias sobre los elementos del código fuente que pueden llegar a ser errores en la codificación.

-o: permite especificar el archivo de salida. Usualmente es la única opción en la línea de comando, si se omite esta opción, el archivo de salida por defecto es a.exe.

Compilando un programa en C

```
$ gcc –Wall primerPrograma.c –o primerPrograma
```

primerPrograma.c: nombre del archivo fuente.

primerprograma: nombre del archivo de salida.


VAMOS PROGRAMADOR, DI LO TUYO
NO COMPILA

Etapas de compilación

Cuando invocamos el comando “gcc”, normalmente se realiza **preprocesamiento, compilación, ensamblado** y **enlazado.**

Cada etapa tiene un código fuente como entrada y un código objeto como resultado, y en las etapas intermedias el código objeto sirve de fuente para la etapa siguiente.

Analicemos estas etapas con el siguiente ejemplo: Calcular el área de un circulo.

Preprocesamiento

El preprocesamiento es realizado por el preprocesador. Esta primera etapa traduce el archivo fuente que es una forma ampliada del lenguaje.

```
$ gcc –E circulo.c –o circulo.i
```

Que puede observar en el archivo circulo.i?, Encontró algún cambio respecto al código fuente?

Luego del preprocesamiento del código fuente es posible ver en el archivo preprocesado circulo.i que la constante simbólica PI, definida con la directiva de preprocesador #define, es sustituida por su valor en todos los lugares donde aparece su nombre. En la etapa de preprocesamiento, entre otras cosas, se resuelven todas las directivas de preprocesador que aparezcan en el código fuente.

Preprocesamiento

```
$ gcc –E circulo.c –o circulo.i
```

Opciones que se usaron para preprocesar el archivo de código fuente:

-E: Opción para detener el proceso de compilación luego de realizado el preprocesamiento. La salida es en la forma de un archivo preprocesado. Los archivos que no requieren preprocesamiento son ignorados.

-o : Opción para especificar el archivo de salida.

circulo.c : Nombre del archivo de entrada a ser preprocesado.

circulo.i : Nombre del archivo de salida. Es el archivo objeto preprocesado, se utiliza el sufijo ".i" para identificar a los archivos preprocesados.

Compilación

La compilación transforma el código C preprocesado en el lenguaje ensamblador propio del procesador de nuestra maquina.

```
$ gcc –S circulo.c
```

Con este comando se realizan las dos primeras etapas y se crea un archivo circulo.s

-S : Opción para detener luego de realizada la etapa de compilación, no realiza la etapa de ensamblado. La salida será en la forma de código assembly(5) . Por defecto, el nombre del archivo de salida será del mismo nombre que el archivo fuente, pero con sufijo ".s". Los archivos que no requieren compilación son ignorados.

circulo.c : Nombre del archivo de entrada para ser compilado.

(5): Los lenguajes ensambladores son un tipo de lenguaje de programación de bajo nivel.

Ensamblado

En la etapa de ensamblado se traduce el programa escrito en el lenguaje ensamblar, de la etapa anterior, a código binario en lenguaje de maquina entendible por el procesador.

```
$ gcc –c circulo.c
```

Con este comando se realizan las tres primeras etapas, creando el archivo circulo.o. Este archivo contiene código binario listo para ser enlazado en la próxima etapa.

-c: opción para detener luego de realizada la etapa de ensamblado.

Enlazado

Las funciones como printf, se encuentran compiladas y ensambladas en librerías existentes. Para que el archivo resultado del proceso de compilación completo sea ejecutable es necesario incorporar el código binario de estas funciones en el lenguaje final. De esto se trata la etapa de enlace.

```
$ gcc  circulo.c –o circulo
```

Esta es la línea de comando que comúnmente se usará al compilar un archivo en C para obtener un archivo ejecutable, agregándole la opción -Wall como se vio en la sección compilando un programa en C.

Corrección y pruebas

Retomemos el ejemplo del área del circulo.

ejemploclase.c > main()

```c
#include <stdio.h>
#define PI 3.1416

int main()
{
    float area, radio;
    printf("Ingresar el radio del circulo: ");
    scanf("%f", &radio);
    area = PI* (radio * radio);
    printf("Area circulo = %.2f", area);
    return 0;

}

```

Corrección y pruebas

El mensaje producido por el compilador gcc no solo proporciona el error sino también la línea donde se encuentra.

```c
#include <stdio.h>

int main()
{
    printf("Hasta la próxima clase!!\n");

    return 0;
}
```