

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

PRÁCTICA 1

Sistema de procesamiento del lenguaje

ALUMNOS:

CÁRDENAS CÁRDENAS JORGE
MURRIETA VILLEGAS ALFONSO
REZA CHAVARRIA SERGIO GABRIEL
VALDESPINO MENDIETA JOAQUIN

PROFESOR:

ADRIAN ULISES MERCADO MARTINEZ

1. Objetivo

Que el alumno identifique el funcionamiento de los distintos programas que permiten llevar a cabo el proceso de traducción en el sistema de procesamiento de lenguaje.

2. Introducción

Un compilador es un programa (traductor) que se encarga de hacer la traducción de un programa fuente escrito en lenguaje de alto nivel a un programa escrito en lenguaje objeto que por lo general es un lenguaje de bajo nivel (lenguaje ensamblador). Para realizar esta traducción el compilador se auxilia de otros programas como lo son el preprocesador, que se encarga de recolectar el programa escrito en módulos en archivos separados (archivos .h y .c), expandir fragmentos de código abreviados de uso frecuentes, llamados macros y la inclusión de las bibliotecas.

El programa modificado por el preprocesador ingresa al compilador, este produce a este producir a el el programa destino escrito en lenguaje ensamblador, que a continuación es procesado por el ensamblador que genera el código de máquina para la arquitectura destino.

Una vez que el programa ha sido ensamblado, es necesario vincular los archivos de código máquina con otros archivos objeto y de biblioteca para que se produzca el código ejecutable. El código máquina que es generado por el ensamblador no es un código que pueda ser ejecutado ya que contiene direcciones de memoria relativas por ello recibe el nombre de código re-localizable.

Finalmente el cargador lleva el archivo objeto a la memoria para su ejecución, asignando direcciones de memoria absolutas.

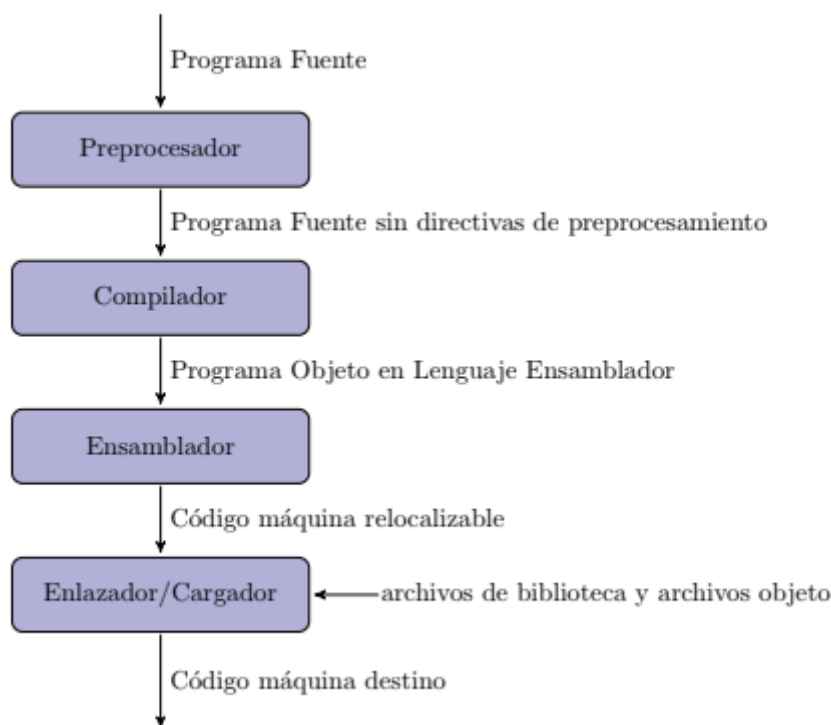
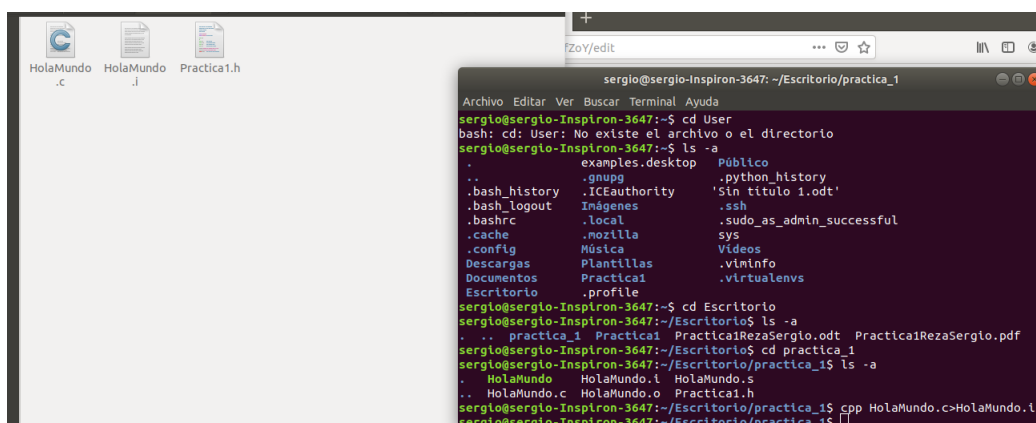


Figura 1: Sistema de procesamiento de un lenguaje.

3. Desarrollo

1. Deberá tener instalado el compilador gcc y trabajar en un ambiente Linux.
2. Crear una carpeta llamada practica 1.
3. Ejecute `cd practica 1`.
4. Escriba un programa tipo "Hola Mundo" en lenguaje C, que tenga comentarios, manejo de macros, uso de bibliotecas, uso de macros y de las directivas de preprocesador como `ifndef`, `define`, `else` y `endif`. El programa debe estar separado en un `.h` y un `.c`
5. Usar el siguiente comando: `cpp programa.c >programa.i`
 - (a) Busque los archivos `.h` de las bibliotecas que uso en su programa y revise su contenido
 - Se utilizó la biblioteca `stdio.h` y la biblioteca generada por nosotros
 - (b) Compare el contenido de `programa.i` con el de `.h` de sus bibliotecas e indique de forma general las similitudes entre ambos archivos.
 - Los archivos `.h` contienen funciones y/o estructuras, en el caso del archivo `.i`, este contiene todo la unión del código ubicado en los archivos `.h` y a su vez el del código que llama a las bibliotecas.
 - (c) ¿Qué pasa con los comentarios en el archivo `.i`?
 - Tanto los comentarios del `.h` como los del `.c` son omitidos en el archivo `.i`
 - (d) ¿Qué ocurre con las macros en el archivo `.i`?
 - En donde son llamados en el código son sustituidos por los valores correspondientes o asignados previamente.
 - (e) ¿Qué ocurre cuando se invoca el comando `cpp` ?
 - Se llama al preprocesador para posteriormente generar el archivo `.i`



```
sergio@sergio-Inspiron-3647: ~/Escritorio/practica_1
Archivo Editar Ver Buscar Terminal Ayuda
sergio@sergio-Inspiron-3647:~$ cd User
bash: cd: User: No existe el archivo o el directorio
sergio@sergio-Inspiron-3647:~$ ls -a
.  ..  examples.desktop  .gnupg  .python_history  .ssh  .viminfo
sergio@sergio-Inspiron-3647:~$ cd Escritorio
sergio@sergio-Inspiron-3647:~/Escritorio$ ls -a
.  ..  practica_1  Practica1  Practica1RezaSergio.odt  Practica1RezaSergio.pdf
sergio@sergio-Inspiron-3647:~/Escritorio$ cd practica_1
sergio@sergio-Inspiron-3647:~/Escritorio/practica_1$ ls -a
.  ..  HolaMundo  HolaMundo.i  HolaMundo.s  Practica1.h
sergio@sergio-Inspiron-3647:~/Escritorio/practica_1$ cpp HolaMundo.c > HolaMundo.i
sergio@sergio-Inspiron-3647:~/Escritorio/practica_1$
```

Figura 2: Comando `cpp programa.c >programa.i` y la creación del archivo con extensión `.i`

6. Ejecute la siguiente instrucción: `gcc -Wall -S programa.i`

- a) ¿Para qué sirve la opción `-Wall`?
 - Se utiliza para que durante el proceso de compilación de un archivo pueda mostrar las advertencias que el programa tenga durante este.
- b) ¿Qué le indica a gcc la opción `-S`?
 - (c) Describa de forma general el contenido del programa de salida.
 - Que se cree un archivo con extensión `.s`
- c) Describa de forma general el contenido del programa de salida.
 - Código escrito que ya no tiene la estructura de los archivos escritos anteriores
- d) ¿Cuál es la extensión del archivo de salida?
 - la extensión es `.s`
- e) ¿Qué programa del sistema de procesamiento de lenguaje es invocado en este punto?
 - Se utiliza el compilador para generar un archivo de texto en lenguaje ensamblador.

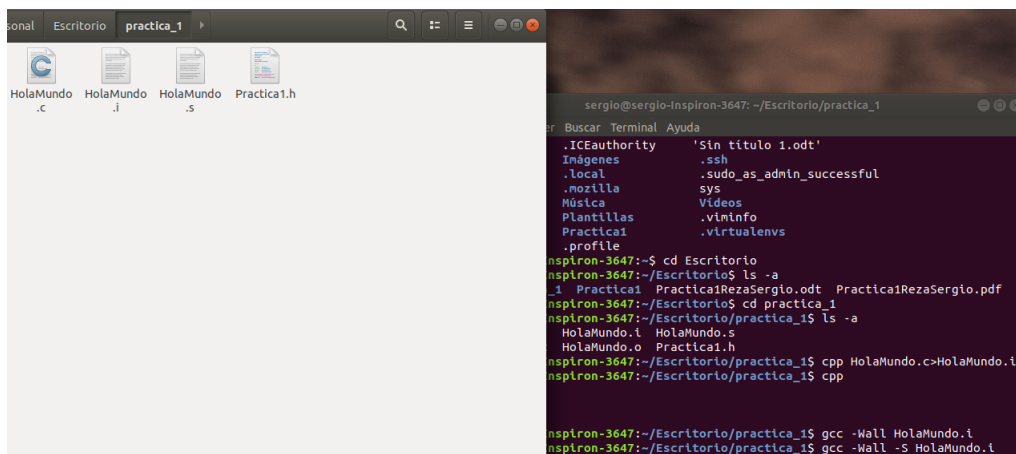


Figura 3: Comando `gcc -Wall -S programa.i` y la creación del archivo con extensión `.s`

7. Ejecute la siguiente instrucción: `as programa.s -o programa.o`

- a) ¿En teoría que debe contener el archivo `.o`?
 - Código máquina relocable, ya que con el comando anterior se utiliza el ensamblador.
 - b) ¿Con un editor de archivos hexadecimal abra el archivo `.o` y diga que contiene?
 - Contiene la representación del código fuente (En notación Hexadecimal), después de haber pasado por el preprocesador, compilador y ensamblador.
- NOTA: Para ver el archivo `.o` se utilizó el editor GHex

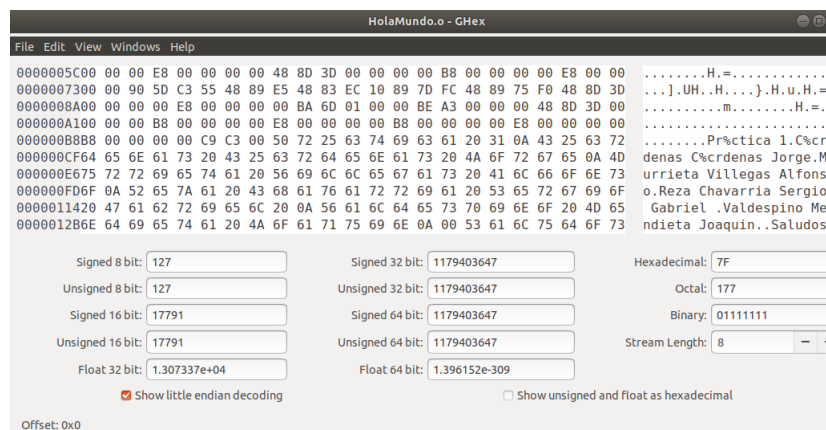


Figura 4: Visualización del programa generado por el ensamblador con GHex

c) ¿Qué programa se invoca con `as`?

- El ensamblador, que maneja el Programa Objeto en Lenguaje ensamblador a código máquina relocable.

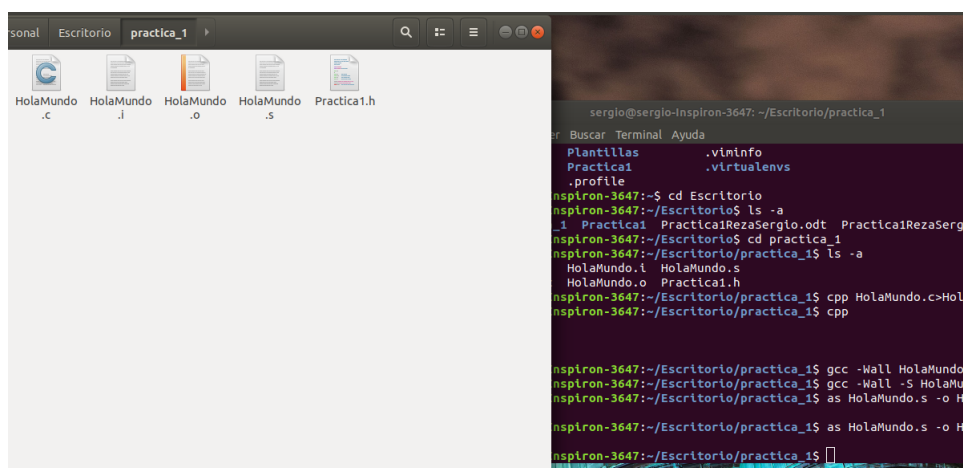


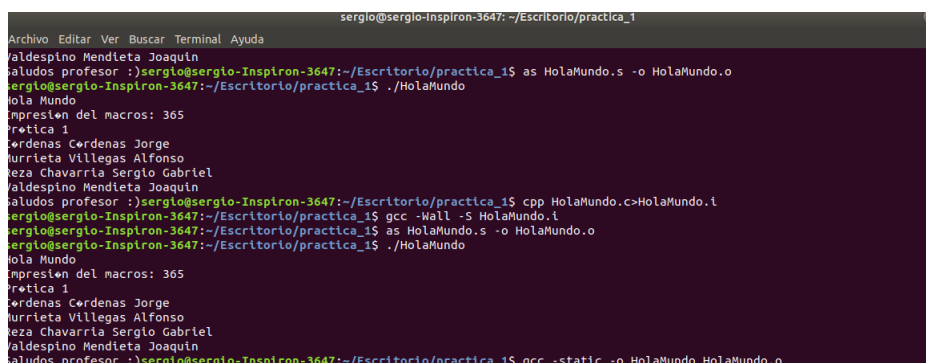
Figura 5: Comando `as programa.s -o programa.o` y la generación del archivo `.o`

8. Enlazador

a) Investigue como ejecutar el enlazador en linux para generar el programa ejecutable.

- `gcc -o NombreDelArchivo NombreDelArchivo .o`

b) Realice el enlazado del programa `.o`



```
sergio@sergio-Inspiron-3647: ~/Escritorio/practica_1
Archivo Editar Ver Buscar Terminal Ayuda
valdespino Mendieta Joaquín
saludos profesor :)sergio@sergio-Inspiron-3647:~/Escritorio/practica_1$ as HolaMundo.s -o HolaMundo.o
sergio@sergio-Inspiron-3647:~/Escritorio/practica_1$ ./HolaMundo
Hola Mundo
Impresión del macros: 365
practica 1
Cárdenas Cárdenas Jorge
Murrieta Villegas Alfonso
Reza Chavarria Sergio Gabriel
valdespino Mendieta Joaquín
saludos profesor :)sergio@sergio-Inspiron-3647:~/Escritorio/practica_1$ gcc -Wall -S HolaMundo.i
sergio@sergio-Inspiron-3647:~/Escritorio/practica_1$ as HolaMundo.s -o HolaMundo.o
sergio@sergio-Inspiron-3647:~/Escritorio/practica_1$ ./HolaMundo
Hola Mundo
Impresión del macros: 365
practica 1
Cárdenas Cárdenas Jorge
Murrieta Villegas Alfonso
Reza Chavarria Sergio Gabriel
valdespino Mendieta Joaquín
saludos profesor :)sergio@sergio-Inspiron-3647:~/Escritorio/practica_1$ gcc -static -o HolaMundo HolaMundo.o
```

Figura 6: Enlace de archivos y ejecución de programa

9. Escriba sus resultados y conclusiones.

4. Conclusiones

4.1. Cárdenas Cárdenas Jorge

Con la realización de esta práctica se lograron reforzar y aclarar algunos de los procesos y conceptos básicos de compilación que previamente se habían analizado en las clases de teoría, tal como lo se hace con las directivas del preprocesador, las macros o los comentarios declarados en un programa C, y lo que posteriormente se realiza para generar el código objeto que se encargara en sí de realizar las instrucciones a nivel procesador; en lo personal esta practica también me permitió entender la tarea del enlazador, el cual se encarga de enlazar los diferentes ficheros objeto para crear un único fichero, que sera el ejecutable.

4.2. Murrieta Villegas Alfonso

En la presente práctica se observaron y analizaron las distintas etapas por las que pasa un programa a través del Sistema de Procesamiento del Lenguaje, en este caso concreto al haber realizado la práctica con lenguaje C, el programa fuente pasó a través un preprocesador (Como resultado el archivo .i), posteriormente por un compilador (Como resultado el archivo .s), por un ensamblador (Como resultado el archivo .o) y finalmente por un enlazador o cargador. Es destacable mencionar que sin duda esta práctica nos ayudó a ver cada una de las etapas mencionadas y además de verlas funcionar a través de un ejemplo propio.

4.3. Reza Chavarria Sergio Gabriel

En conclusión se pudo comprender y analizar el procedimiento de un programa en un lenguaje de programación, en este caso del lenguaje C, para poder utilizarlo como un ejecutable. Para esto se debe de seguir con los procesos de un programa en un lenguaje compilado. Gracias a esto se pudo revisar los puntos vitales para este proceso, como lo son el preprocesador, compilador, ensamblador y enlazador, junto con sus diferentes programas generados por cada uno de estos.

4.4. Valdespino Mendieta Joaquín

En conclusión el sistema de procesamiento de lenguaje es algo fundamental para poder ejecutar un programa, en este caso de lenguaje C, es importante observar y comprender que cada uno de los subprocessos de este conlleva a una parte del proceso general, traduciendo cada instrucción desde código fuente hasta el código maquina que se ejecutara, pasando por las directivas del preprocesador, compilador, ensamblador y por ultimo el enlazador, en fin se pudo comprender el proceso de compilación y ejecución de un programa desde otra perspectiva.

5. Referencias

Universidad de la República- Facultad de Ingeniería Instituto de Ingeniería eléctrica. El compilador GCC.
Encontrado en: <https://iie.fing.edu.uy/vagonbar/gcc-make/gcc.htm>