



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: M.I. Marco Antonio Martínez Quintana

Asignatura: Fundamentos de Programación

Grupo: 3 Bloque: 136

No de Práctica(s): 6 Entorno de C (editores, compilación y ejecución)

Integrante(s): Carranza Ochoa José David

*No. de Equipo de
cómputo empleado:* No aplica

No. de Lista o Brigada: 6

Semestre: 2021-1

Fecha de entrega: 16/11/2020

Observaciones:

CALIFICACIÓN: _____

Objetivo:

Conocer y usar los ambientes y herramientas para el desarrollo y ejecución de programas en Lenguaje C, como editores y compiladores en diversos sistemas operativos.

Introducción

Un lenguaje de programación permite expresar una serie de instrucciones que podrán ser realizadas por una computadora. Uno de los lenguajes de programación mayormente difundidos es el lenguaje C.

Este es muy utilizado ya que la forma de dar instrucciones es muy cercana a lo que un humano podría abstraer, es decir, las instrucciones no son tal cual las que una computadora podría entender, para ello se necesitaría conocer a fondo el microprocesador, el sistema operativo entre otros aspectos. Por esta razón, C es conocido como un lenguaje de alto nivel, esto significa a que las instrucciones podrían ser entendidas fácilmente por un humano. En contraparte, un lenguaje de bajo nivel, son instrucciones que son cercanas a lo que la máquina puede entender y difícilmente pueden ser comprendidas por una persona que no tenga conocimientos de la máquina en que operarán. Algunos autores consideran al lenguaje C como un lenguaje de mediano nivel, ya que no es totalmente transparente sino tiene elementos que tienen que ver con la arquitectura de la máquina a la hora de programar.

Conociendo C desde GCC

A través del sistema operativo Linux se puede desarrollar software por medio de la terminal como compilador, siendo este el compilador GCC quien interpreta el programa de salida que se muestra.

Debemos recordar que dicho compilador debe verificar el funcionamiento correcto del código para después ejecutarlo, es por eso que GCC necesita compilar dicho código mediante la siguiente estructura:

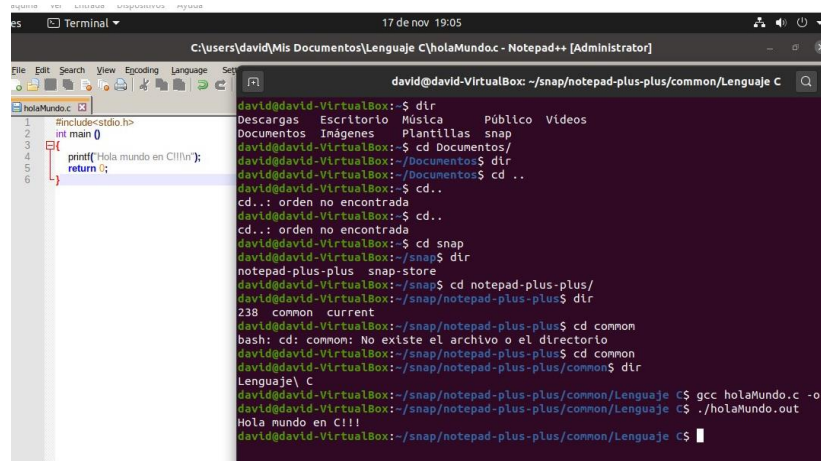
```
gcc nombreArchivo.c -o nombreArchivo.out
```

Donde previamente se debió guardar el archivo en formato “.c” por medio de un editor de texto, tal fue el caso de NotePad++

Ahora para ejecutar el programa compilado desde GCC se ingresa en la terminal (según sea el caso) el comando:

```
nombreArchivo.out
```

Con base a lo anterior, podemos desarrollar nuestro primer “hola Mundo” desde C siguiendo la sintaxis del programa, obteniendo un excelente código de salida.



```
david@david-VirtualBox: ~/snap/notepad-plus-plus/common/Lenguaje C
david@david-VirtualBox:~$ dir
Descargas Escritorio Música Público Videos
Documentos Imágenes Plantillas snap
david@david-VirtualBox:~$ cd Documentos/
david@david-VirtualBox:~/Documentos$ dir
david@david-VirtualBox:~/Documentos$ cd ..
david@david-VirtualBox:~$ cd..
cd.: orden no encontrada
david@david-VirtualBox:~$ cd..
cd.: orden no encontrada
david@david-VirtualBox:~$ cd snap
david@david-VirtualBox:~/snap$ dir
notepad-plus-plus snap-store
david@david-VirtualBox:~/snap$ cd notepad-plus-plus/
david@david-VirtualBox:~/snap/notepad-plus-plus$ dir
238 common current
david@david-VirtualBox:~/snap/notepad-plus-plus$ cd common
bash: cd: common: No existe el archivo o el directorio
david@david-VirtualBox:~/snap/notepad-plus-plus$ cd common
david@david-VirtualBox:~/snap/notepad-plus-plus/common$ dir
Lenguaje\c
david@david-VirtualBox:~/snap/notepad-plus-plus/common/Lenguaje C$ gcc holaMundo.c -o h
david@david-VirtualBox:~/snap/notepad-plus-plus/common/Lenguaje C$ ./holaMundo.out
Hola mundo en C!!!
david@david-VirtualBox:~/snap/notepad-plus-plus/common/Lenguaje C$
```

Conclusiones:

A partir de esta práctica conocí un entorno de desarrollo de software en el cual presenta un lenguaje comprensible entre la máquina y el humano, así mismo, aprendí la diferencia entre un compilador y un editor de texto los cuales son totalmente diferentes, pero van relacionados al desarrollar un programa.

Al tratarse de GCC como compilador, decidí realizar mi práctica desde GNU Linux ya que su interfaz es más intuitiva para realizar software de manera amena; así mismo, a diferencia de Windows, GCC ya viene instalado en paquetes de Ubuntu.

Por medio de C, espero realizar mi proyecto final ya que cuenta con una gran potencia de codificación y es de los lenguajes más emblemáticos al ser pionero de otros tantos.

Bibliografía

♣ Dr. Pedro Alberto Enríquez Palma. Editor VI. Consulta: septiembre de 2016. Disponible en: <http://www.unirioja.es/cu/enriquez/docencia/Quimica/vi.pdf>

♣ Francisconi Hugo Adrian. Nano. Consulta: septiembre de 2016. Disponible en: <http://francisconi.org/linux/comandos/nano>

♣ G2 Crowd. ATOM vs. Notepad++. Consulta: septiembre de 2016. Disponible en: <https://www.g2crowd.com/compare/atom-vs-notepad>

♣ Gerald Pfeifer (GCC team). GCC, the GNU Compiler Collection. Consulta: septiembre de 2015. Disponible en: <https://gcc.gnu.org>

♣ MinGW.org. MinGW - Minimalist GNU for Windows. Consulta: septiembre de 2015. Disponible en: <http://www.mingw.org>

♣ White-Hat Hacking. Uso de gcc bajo Linux. Consulta: septiembre de 2016. Disponible en: <https://whitehathacking.wordpress.com/2010/10/31/uso-de-gcc-bajo-linux/>

♣ Willus.org. Win32/64 C/C++ Compilers Page. Consulta: septiembre de 2016. Disponible en: <http://www.willus.com/ccomp.shtml>

♣ Fabrice Bellard. Tiny C Compiler. Consulta: septiembre de 2015. Disponible en: <http://bellard.org/tcc/>