|  |  |  |
| --- | --- | --- |
|  | **Carátula para entrega de prácticas** | |
| Facultad de Ingeniería | | Laboratorio de docencia |

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* | Ing. Rodríguez Espino Claudia |
| *Asignatura:* | Fundamentos de programación |
| *Grupo:* | 1122, Bloque: 139 |
| *No de Práctica(s):* | Sexta práctica |
| *Integrante(s):* | Fonseca Ramírez Gadiel Moisés |
|  |  |
|  |  |
| *Semestre:* | 2018-1 |
| *Fecha de entrega:* | 22/Septiembre/2017 |
| *Observaciones:* |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

**Guía práctica de estudio 06: Entorno de C (editores, compilación y ejecución)**

**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.

**Actividades:**

* Utilizando un editor de GNU/Linux, crear un archivo de texto.
* Modificar/actualizar un archivo ya existente con un editor GNU/Linux.
* Crear, compilar y ejecutar un programa simple escrito en C en GNU/Linux.
* En algún entorno de desarrollo de Windows, crear, compilar y ejecutar un programa simple escrito en C.

**Actividades realizadas en clase:**

* Para realizar esta práctica, primero iniciamos el sistema operativo Ubuntu, que está basado en Linux, y abrimos la terminal de comandos.
* El profesor explicó los conceptos de la práctica y de forma general lo que íbamos a hace. Explicó que al crear un archivo en C, que sería nuestro código fuente del programa, al compilarlo, este se convierte a lenguaje máquina y el compilador revisa la sintaxis del código, y crea un archivo ejecutable (.exe). Pero eso no quiere decir que el programa funcione de forma correcta, puesto que si el código está bien escrito, pero la lógica del programa hace que no se pueda realizar ninguna acción o se generen errores, la computadora sólo hará lo que diga el programa, aunque este no tenga el funcionamiento que debería.
* Para comenzar a crear nuestro programa en C, primero debemos crear un archivo con el comando “touch”, con la extensión “.c” en el nombre de nuestro archivo.

touch suma.c

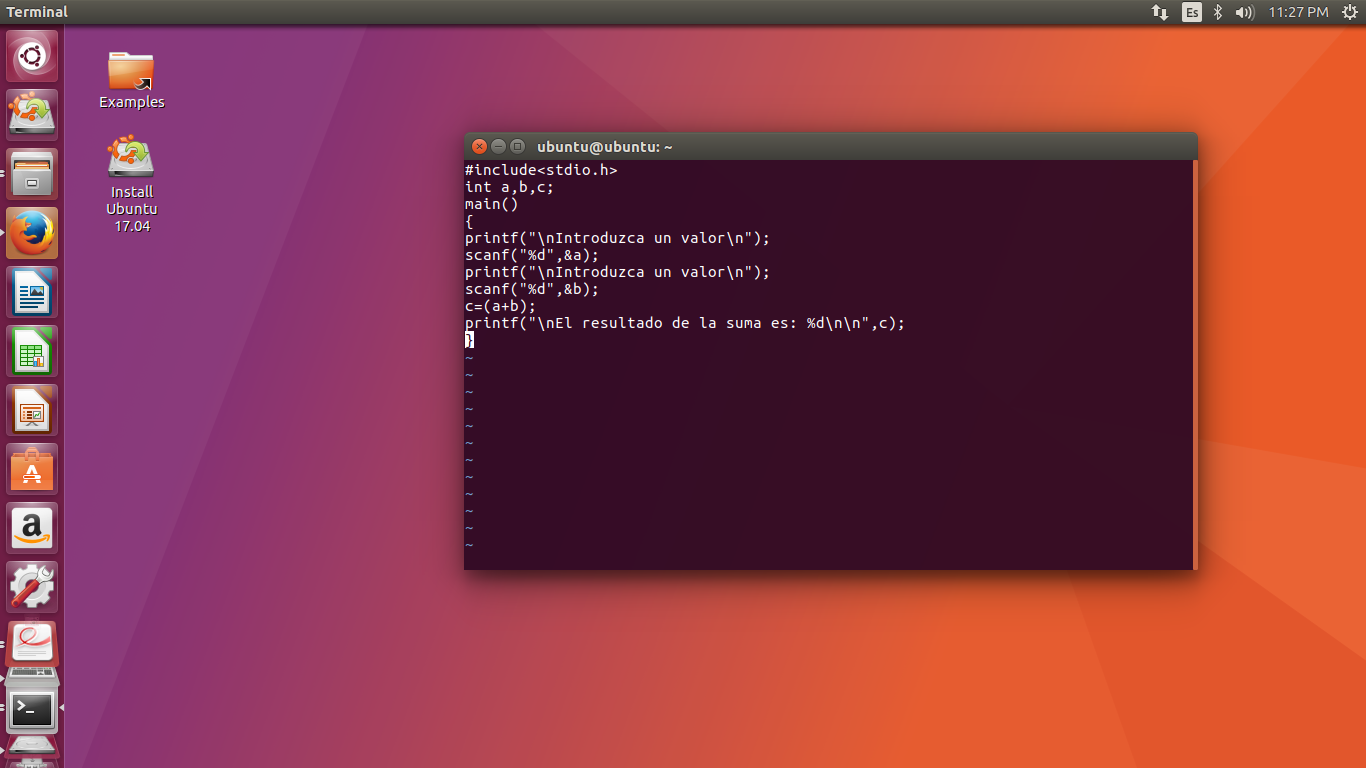
* Una vez creado el archivo, se debe abrir en el “VI” (Editor Visual Interface de GNU/LINUX), esto se hace de la forma (para suma.c):

vi suma.c

* Al abrir el editor (VI), la terminal aparecerá en vacía, pero llena de virgulillas (~). Al entrar estamos en modo de comandos, por lo que no podemos escribir. En este modo es en el que se pueden usar los comandos de “Modo comando”, “Modo de última línea” y “Modo insertar”.
* Para poder comenzar a escribir, presionamos la tecla “i” una vez y ya podemos comenzar a escribir.
  + Los ejercicios que dejó realizar la profesora fueron::

1. Mediante el editor VI de GNU/Linux visualizar el programa 1: suma.c (suma de dos números).

Como este programa ya se había visto en clase, sólo había que escribirlo, cabe señalar que cuando se usa el “VI” para escribir el programa, los comandos y textos tienen distintos colores dependiendo de si son variables, textos, librerías, etc. Pero en mi versión de Ubuntu (17.04), todo el texto era blanco. El profesor también nos enseñó que una vez escrito el programa, este se puede ejecutar desde el “VI” estando en “modo de comandos”.



1. Modificar el programa: suma dedos números cambiando el proceso de suma a resta y guardar el archivo como “resta.c”

Si queremos conservar nuestro archivo original del programa para sumar, podemos hacerle una copia cambiándole el nombre con el comando “mv” de este modo:

cp suma.c resta.c

Así se creará un segundo archivo llamado “resta.c” y para modificarlo, sólo tenemos que entrar al “VI”

vi resta.c

y cambiar el signo de la operación de “+” a “-“, además de en mi caso el mensaje final de “El resultado de la suma es:” a “El resultado de la resta es:”, y para guardar los cambios, estando en “modo de comandos” usamos :w (o :wq para guardar y salir del VI).

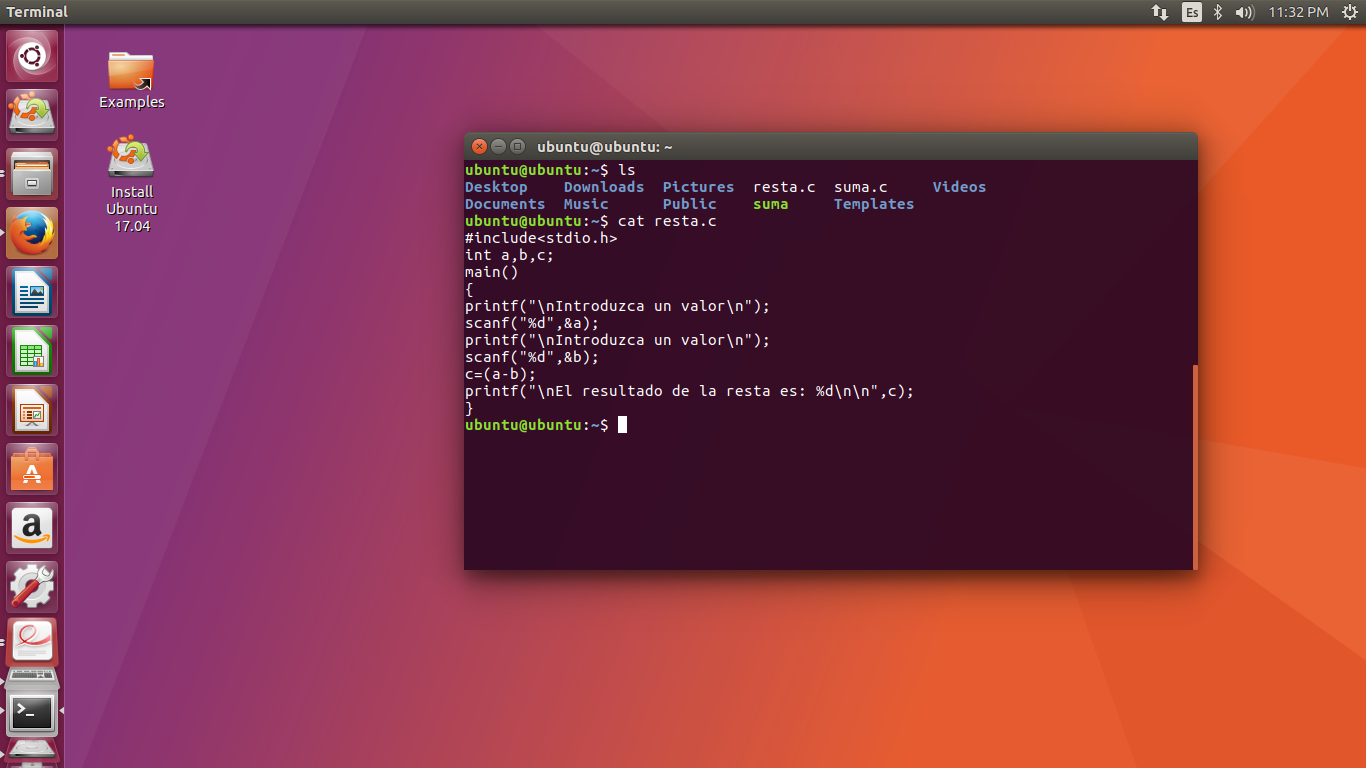
Como segundo método para modificar el archivo, podemos abrir el “VI” y estando en modo de comandos (para pasar de modo texto a modo comandos, presionamos la tecla Esc) tecleamos:

:w resta.c (:w NombreNuevoDelArchivo)

Este comando permite que el archivo “suma.c” se guarde como “resta.c”, esto quiere decir que se crea un nuevo archivo (resta.c) y se conserva el anterior (suma.c). Para salir de “suma.c” sin guardar las modificaciones usamos :q!

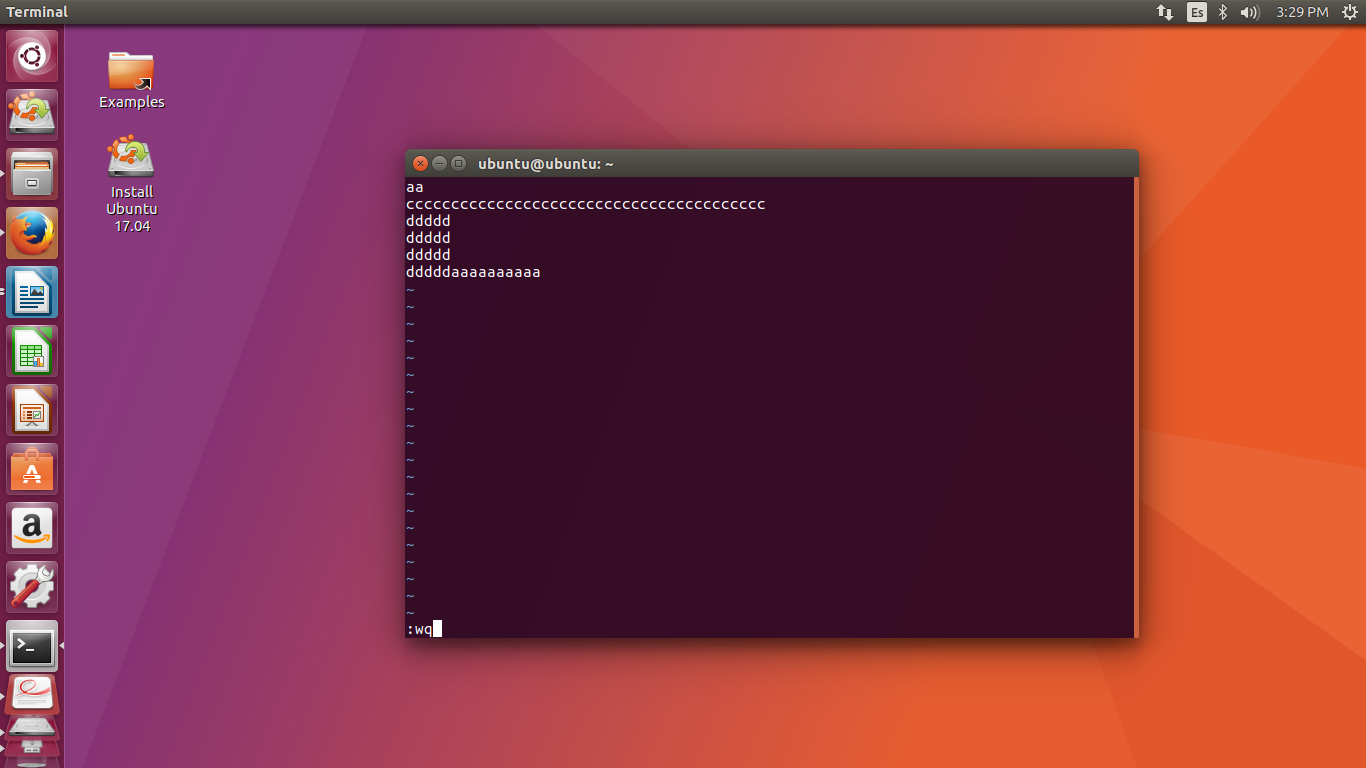
También vimos que estando fuera del “VI” podemos ver el contenido de un archivo con el comando “cat”, de la siguiente forma:

cat resta.c (cat Archivo)



1. Con el programa suma.c practicar los comandos detallados en la parte “modo comando” y “modo última línea”.

Estos comandos permiten realizar las acciones que normalmente se podrían realizar en un sistema operativo de interfaz gráfica con el botón “clic derecho”, con los “atajos de teclado” o los botones que aparezcan en la interfaz del programa. Estos comandos los pusimos en práctica con el profesor usando una serie de letras.



Cabe señalar que cuando se escriben comandos del tipo “modo última línea”, nos vamos hasta la última línea del “VI”.

1. Compilar archivo suma.c con gcc (compiler collection).

Para hacer esto, tenemos que estar fuera del “VI” y usamos el comando

gcc suma.c (gcc ArchivoACompilar).

Para que al momento de compilar el programa, el archivo ejecutable que se genera (a.out) tenga otro nombre, se escribe el comando como

gcc suma.c –o suma (gcc Archivo1 –o Archivo2)

Entonces ahora el ejecutable creado se llamaría “suma” y no “a.out”

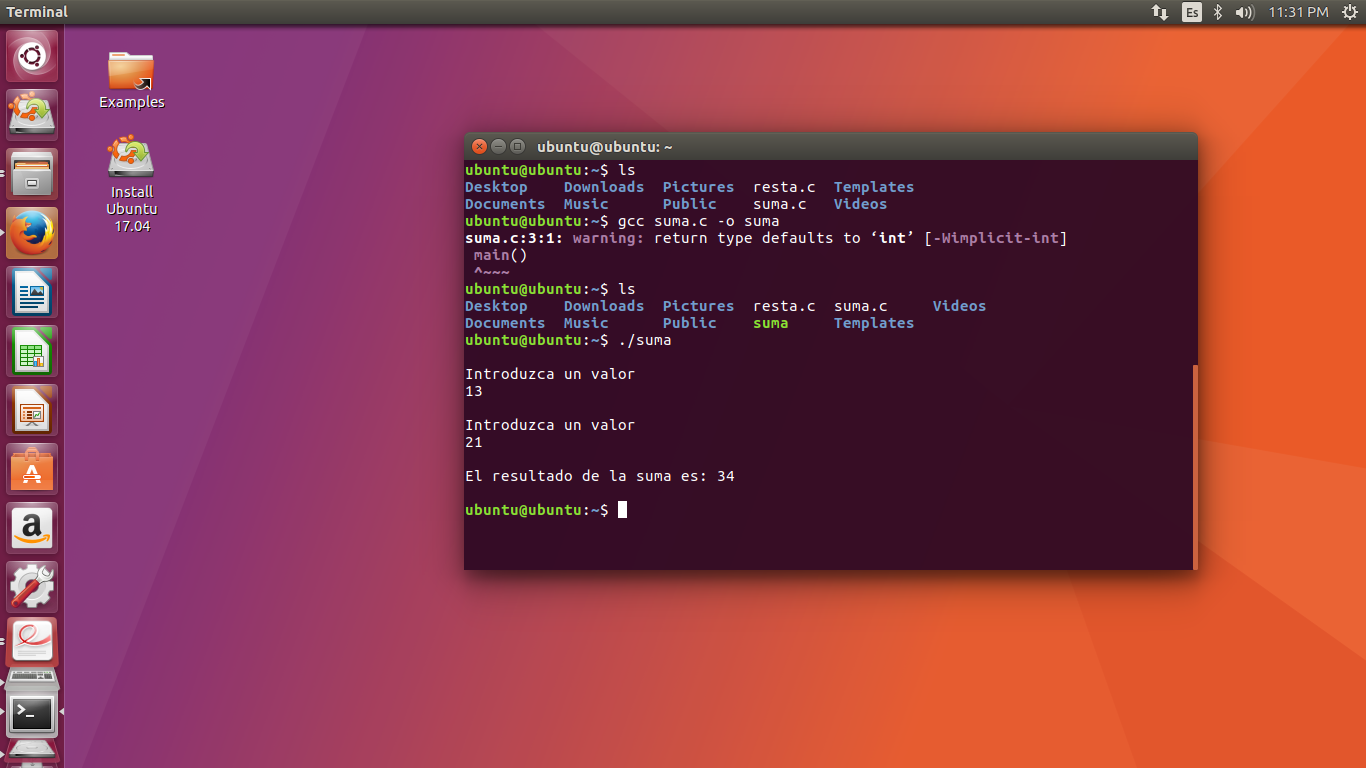
Esto crea un archivo ejecutable con el nombre de “a.out”, que se puede ejecutar con el comando

./a.out (./NombreDelArchivo)

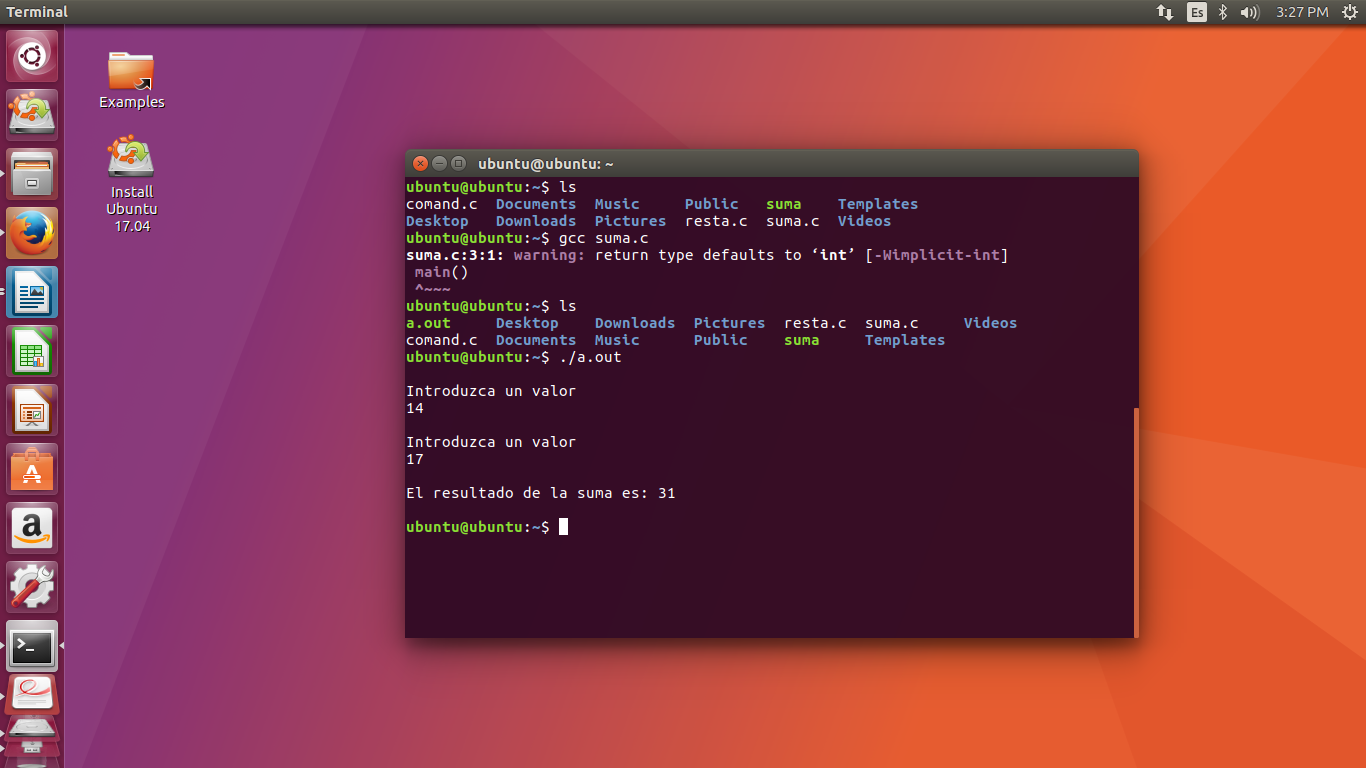
./suma

Esto hace que corra el programa y realice la función que debe cumplir.

Compilado como suma:



Compilado como a.out:



**Conclusión:**

Esta práctica permitió poner en práctica los conocimientos acerca del uso y funciones de las herramientas para desarrollo de programas en lenguaje C, en este caso, con la herramienta de GCC/Linux Compiller Collection. Además de poner en práctica los conocimientos para manejar la terminal de comandos de Linux, permitiendo un mejor entendimiento del sistema o del sistema operativo, sus ventajas y utilidades.