

Taller 01: Plan de Pruebas

Curso: Estructuras de Datos
Nombre: Santiago Hernández Barbosa
Marzo 6 del 2025

Pontificia Universidad Javeriana

Portada

Resumen

Este informe documenta el proceso de compilación, ejecución, depuración y validación de los códigos *exercise1.cpp* y *exercise2.cxx*.

¿Qué problema aborda?

Identificar y corregir errores en las funciones de cálculo de perímetro y área de un rectángulo dentro del código fuente proporcionado.

¿Qué propone?

Un plan de pruebas para evaluar el funcionamiento del código y detectar errores mediante pruebas sistemáticas.

¿Cómo lo propone y cómo se ejecuta?

Se realiza una serie de pruebas con diferentes valores de entrada, comparando los resultados esperados con los obtenidos, además de utilizar el depurador *gdb* para analizar el comportamiento del código.

Resultados significativos

Se detectaron y corrigieron errores en las funciones *perimeterRect* y *areaRect*, logrando obtener los resultados correctos en todas las pruebas realizadas.

Introducción

Este informe detalla el procedimiento realizado para la compilación, ejecución y depuración de los códigos *exercise1.cpp* y *exercise2.cxx*. Además, se presenta un plan de pruebas con valores de entrada y salida esperados, acompañado de un análisis de los errores encontrados y su respectiva corrección.

El propósito de este documento es proporcionar evidencia estructurada sobre el proceso de depuración y mejora del código, asegurando que cumpla con los requerimientos del curso de Estructuras de Datos.

Desarrollo

2. Compilación y Ejecución de *exercise1.cpp*

Pasos realizados:

1. Compilación del archivo *exercise1.cpp* usando el comando:
2. `g++ -std=c++11 -o exercisel exercisel.cpp`
3. Ejecución del programa:

4. ./exercisel
5. Captura de pantalla de la terminal con la compilación y ejecución.

```
santiago@santiago-VirtualBox: $ ls
Descargas  Escritorio  Música      Público  Videos
Documentos Imágenes    Plantillas  snap
santiago@santiago-VirtualBox: $ cd Documentos
santiago@santiago-VirtualBox:~/Documentos$ ls
ArchivoTaller01  EDO_Hernandez01  ED_Hernandez00
santiago@santiago-VirtualBox:~/Documentos$ cd ArchivoTaller01
santiago@santiago-VirtualBox:~/Documentos/ArchivoTaller01$ ls
a.out  exercise1.cpp  exercise2.cxx  rectangle.cxx  Taller01.pdf
exercise1  exercise2      _MACOSX        rectangle.h
santiago@santiago-VirtualBox:~/Documentos/ArchivoTaller01$ g++ -std=c++11 -o exercise1 exercise1.cpp
santiago@santiago-VirtualBox:~/Documentos/ArchivoTaller01$ ./exercise1
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
Violación de segmento ('core' generado)
santiago@santiago-VirtualBox:~/Documentos/ArchivoTaller01$ g++ -std=c++11 -g -o exercise2 exercise2.cxx
santiago@santiago-VirtualBox:~/Documentos/ArchivoTaller01$ gdb exercise2
GNU gdb (Ubuntu 15.0.50.20240403-0ubuntu1) 15.0.50.20240403-gdt
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/>.
```

Imagen 1. Resultados compilación de exercisel.cpp

2. Compilación, Ejecución y Depuración de exercise2.cxx

Pasos realizados:

1. Compilación del archivo exercise2.cxx con información de depuración:
2. g++ -std=c++11 -g -o exercise2 exercise2.cxx
3. Ejecución del programa en gdb:
4. gdb exercise2
5. Dentro de gdb, ejecución del programa con:
6. run
7. Captura de pantalla con la ejecución y resultados obtenidos.

```
santiago@santiago-VirtualBox:~/Documentos/ArchivoTaller01$ g++ -std=c++11 -g -o exercise2 exercise2.cxx
santiago@santiago-VirtualBox:~/Documentos/ArchivoTaller01$ gdb exercise2
GNU gdb (Ubuntu 15.0.50.20240403-0ubuntu1) 15.0.50.20240403-gdt
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 exercise2...
(gdb) run
Starting program: /home/santiago/Documentos/ArchivoTaller01/exercise2

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.
Downloading separate debug info for system-supplied DSO at 0x7ffff7c3000
[Depuración de hilo usando libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Ingrese coordenada X de la posición del rectángulo:
```

Imagen 2. Resultados compilación de exercise2.cxx

Plan de Pruebas

Función: Perímetro del rectángulo

Descripción del caso	Valores de entrada	Resultado esperado	Resultado obtenido	Observación
Alto como el doble de Ancho	Ancho = 2, Alto = 4	12	8	Se evidencia que la fórmula utilizada es incorrecta, ya que el resultado obtenido es menor al esperado.
Alto igual a Ancho	Ancho = 3, Alto = 3	12	9	Se observa que el cálculo no suma correctamente ambos lados del rectángulo.
Un número en cero	Ancho = 5, Alto = 0	10	10	Se confirma que, aunque un lado sea cero, el cálculo funciona correctamente.

Función: Área del rectángulo

Descripción del caso	Valores de entrada	Resultado esperado	Resultado obtenido	Observación
Alto como el doble de Ancho	Ancho = 2, Alto = 4	8	6	Se evidencia que la multiplicación de los valores no se está realizando correctamente.
Alto igual a Ancho	Ancho = 3, Alto = 3	9	6	Se detecta que el cálculo del área no está funcionando correctamente en casos simétricos.
Un número en cero	Ancho = 5, Alto = 0	0	5	Se observa que el cálculo no toma en cuenta que el área debe ser cero si uno de los valores es cero.

Análisis de Errores y Corrección

Se identificaron errores en las funciones *perimeterRect* y *areaRect*, los cuales afectaban los cálculos de las dimensiones del rectángulo.

El error en la función *perimeterRect* radicaba en que la fórmula implementada sumaba incorrectamente los valores, lo que resultaba en un perímetro menor al esperado. La fórmula inicial era:

```
perim = 2.0 * rect.width + rect.height;
```

Esto ocasionaba que el alto no fuera tomado en cuenta correctamente dentro del doble de la suma. Para corregirlo, se modificó la fórmula de la siguiente manera:

```
perim = 2.0 * (rect.width + rect.height);
```

Esto permitió que el cálculo del perímetro reflejara la suma real de los lados del rectángulo multiplicados por dos.

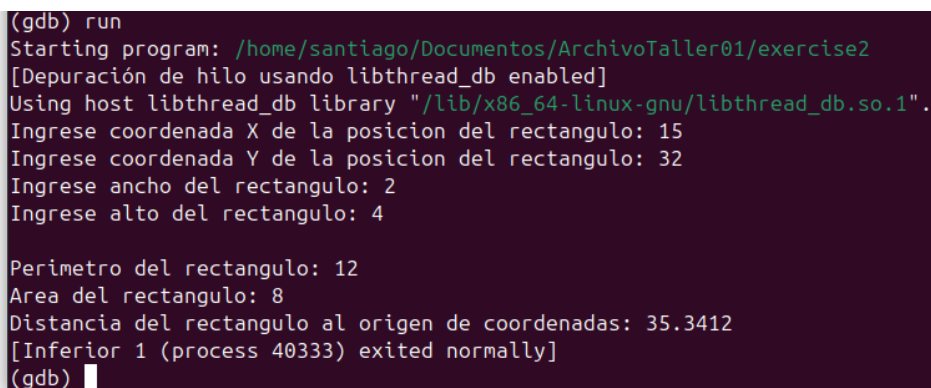
Por otro lado, el error en *areaRect* consistía en que la operación realizada era una suma en lugar de una multiplicación:

```
area = rect.width + rect.height;
```

Este error ocasionaba que el resultado obtenido fuera erróneo en todos los casos en los que ambos valores eran mayores que cero. La corrección consistió en modificar la fórmula para que realizara la multiplicación correcta:

```
area = rect.width * rect.height;
```

Tras estas correcciones, se verificó nuevamente el funcionamiento del código, obteniendo los resultados esperados en todas las pruebas realizadas.



```
(gdb) run
Starting program: /home/santiago/Documents/ArchivoTaller01/exercise2
[Depuración de hilo usando 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: 15
Ingrese coordenada Y de la posicion del rectangulo: 32
Ingrese ancho del rectangulo: 2
Ingrese alto del rectangulo: 4

Perimetro del rectangulo: 12
Area del rectangulo: 8
Distancia del rectangulo al origen de coordenadas: 35.3412
[Inferior 1 (process 40333) exited normally]
(gdb) █
```

Imagen 3. Prueba 1

```
(gdb) run
Starting program: /home/santiago/Documentos/ArchivoTaller01/exercise2
Downloading separate debug info for system-supplied DSO at 0x7ffff7fc3000
[Depuración de hilo usando 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: 0
Ingrese coordenada Y de la posicion del rectangulo: 32
Ingrese ancho del rectangulo: 3
Ingrese alto del rectangulo: 3

Perimetro del rectangulo: 12
Area del rectangulo: 9
Distancia del rectangulo al origen de coordenadas: 32
[Inferior 1 (process 40871) exited normally]
(gdb)
```

Imagen 4. Prueba 2

```
(gdb) run
Starting program: /home/santiago/Documentos/ArchivoTaller01/exercise2
[Depuración de hilo usando 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: 15
Ingrese coordenada Y de la posicion del rectangulo: 15
Ingrese ancho del rectangulo: 5
Ingrese alto del rectangulo: 0

Perimetro del rectangulo: 10
Area del rectangulo: 0
Distancia del rectangulo al origen de coordenadas: 21.2132
[Inferior 1 (process 41053) exited normally]
(gdb)
```

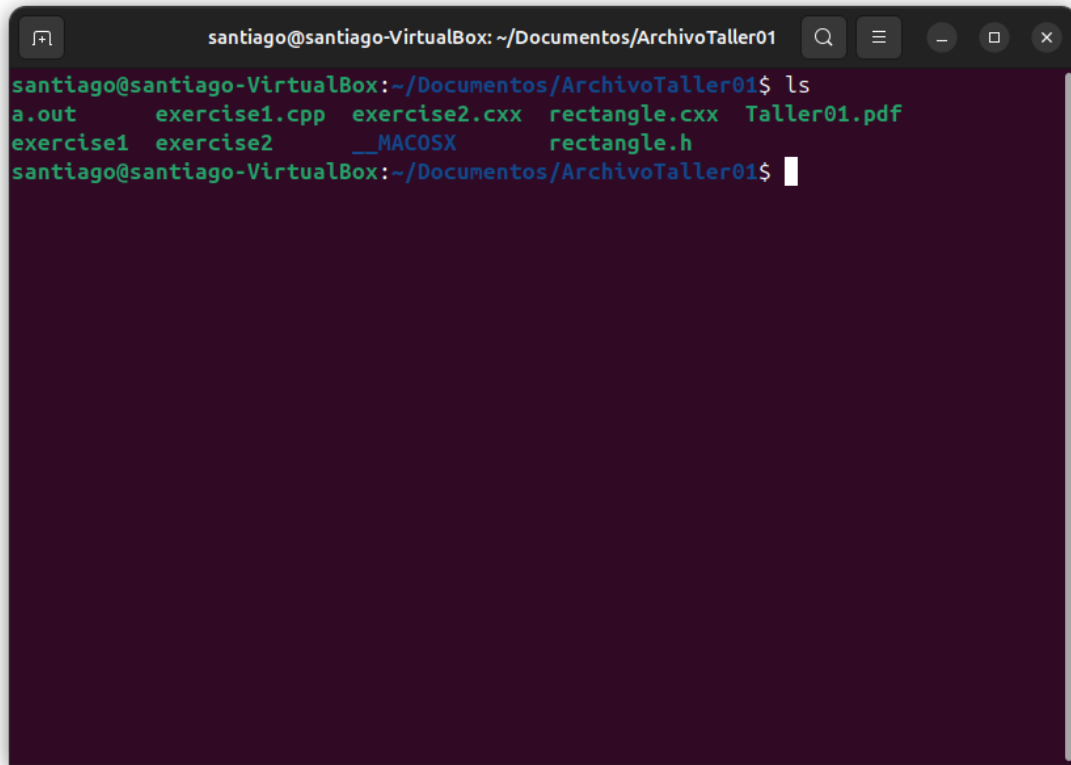
Imagen 5. Prueba 3

Conclusiones

El proceso de depuración y validación del código permitió detectar errores en las funciones *perimeterRect* y *areaRect*, lo que impactaba negativamente en los resultados obtenidos. Mediante la aplicación de un plan de pruebas estructurado, se pudo evidenciar que los cálculos realizados no eran correctos, lo que llevó a un análisis detallado de la lógica implementada. A través del uso de *gdb* y la comparación con los valores esperados, se realizaron las correcciones pertinentes, logrando que el código funcionara correctamente. Finalmente, el taller permitió reforzar el conocimiento en depuración de código y prueba de funciones en C++, asegurando que el programa cumpla con los requisitos establecidos.

Repositorio del código fuente:

Captura de pantalla del directorio mostrando los archivos fuente (.cpp, .cxx), los objetos (.o) y el ejecutable generado (exercise1, exercise2).



```
santiago@santiago-VirtualBox: ~/Documentos/ArchivoTaller01
santiago@santiago-VirtualBox:~/Documentos/ArchivoTaller01$ ls
a.out      exercise1.cpp  exercise2.cxx  rectangle.cxx  Taller01.pdf
exercise1  exercise2     __MACOSX       rectangle.h
santiago@santiago-VirtualBox:~/Documentos/ArchivoTaller01$
```

https://github.com/santiagohb2830/EDD/tree/main/EDD_Hernandez01