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

# Laboratorios de computación salas A y B

*Profesor:* Marco Antonio Martínez Quintana.

*Asignatura:* Fundamentos de programación

*Grupo:* 3

*No de Práctica(s):* 10

*Integrante(s):* Moreno Razo Laura Mildred

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

*No. de Lista o Brigada:* No aplica

*Semestre:* 2021-1

*Fecha de entrega:* Lunes 7 de Diciembre del 2020

*Observaciones:*

CALIFICACIÓN: _____

## Objetivo

Aprender las técnicas básicas de depuración de programas en C para revisar de manera precisa el flujo de ejecución de un programa y el valor de las variables; en su caso corregir posibles errores.

## Actividades:

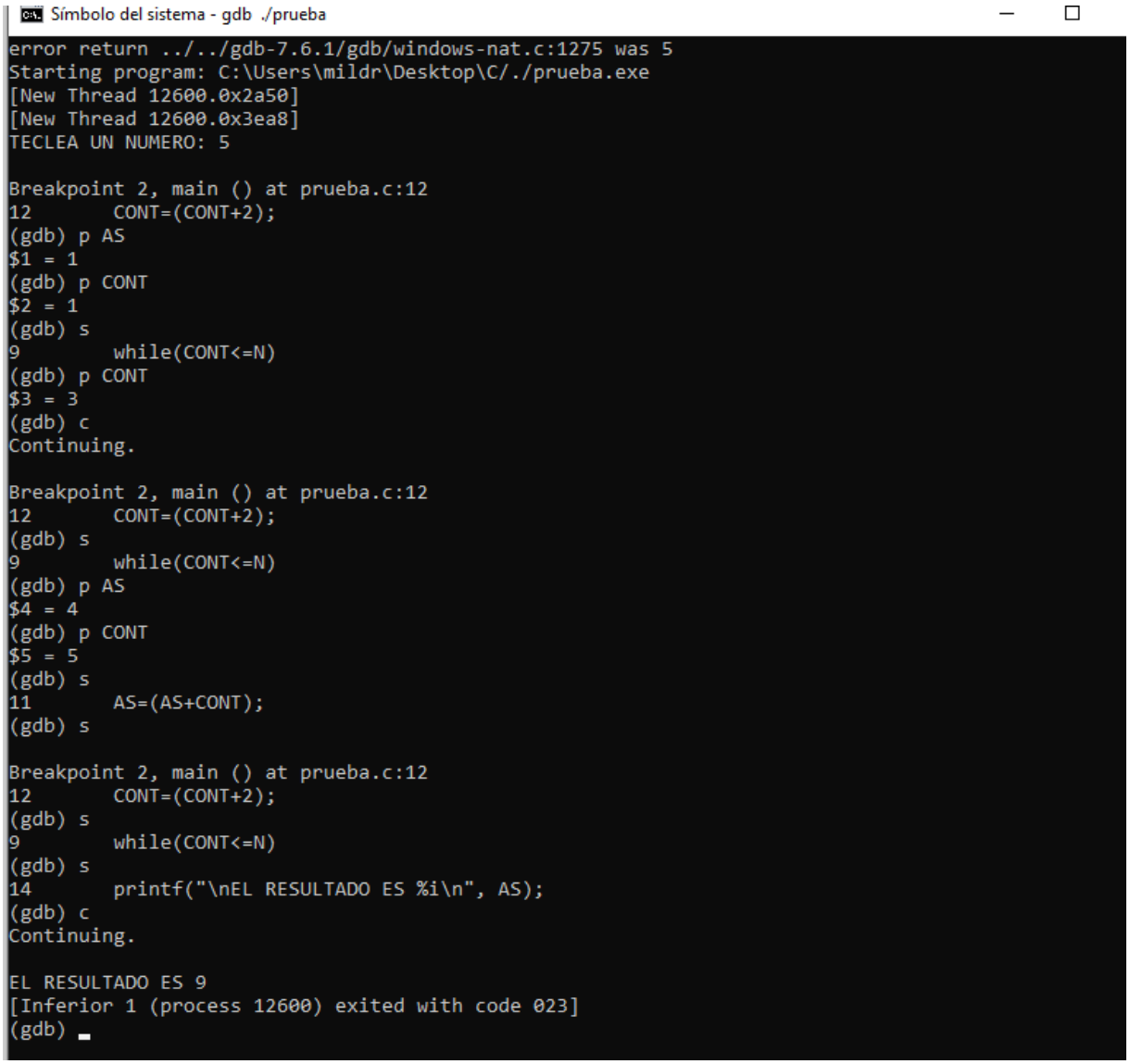
- Revisar, a través de un depurador, los valores que va tomando una variable en un programa escrito en C, al momento de ejecutarse.
- Utilizando un depurador, revisar el flujo de instrucciones que se están ejecutando en un programa en C, cuando el flujo depende de los datos de entrada.

## Introducción

Un depurador es un programa que permite detectar y diagnosticar fallos en programas informáticos. El objetivo de estas herramientas es garantizar, a largo plazo, que el software funcione en todos los dispositivos y plataformas para los que está pensado. Por este motivo, muchos depuradores no solo analizan el código fuente del programa, sino también su interacción con el sistema operativo que lo ejecuta y con los elementos de hardware. El proceso de depuración o debugging ocurre mientras el programa se ejecuta, de forma que no es necesario cerrarlo para llevar a cabo el análisis.

La manera de implementar un depurador para identificar errores de software dependerá, del tipo de debugger en concreto y, por otro, del sistema operativo en el que se ejecute el software analizado (y el depurador). Como usuario, se ha de comunicar a la herramienta sobre qué proceso ha de actuar: ya sea en forma de identificador o ID de proceso; o usando un nombre, en cuyo caso el depurador habrá de averiguar, en primer lugar, el ID correspondiente. A continuación, se inicia la sesión de depuración mediante una llamada al sistema.

Actividad 1.




La utilidad de este programa es realizar la suma, a partir del número 1 de todos los números impares hasta cierto número dado, incluyendo al número en caso de también cumplir la condición de ser impar.

Esto los podemos inferir gracias al análisis paso a paso del programa, en este caso añadí un breakpoint en la línea 12 para observar como se transformaba el valor de la variable AS después de la operación, ya que es donde se acumula el resultado, además utilicé el comando “s” para ejecutar la línea siguiente y saber que valor tomaba CONT después de la operación, ya que CONT estaba dentro de la condición

del while y el valor tomado era útil para saber si se cumplía y seguía ejecutándose la iteración o no.

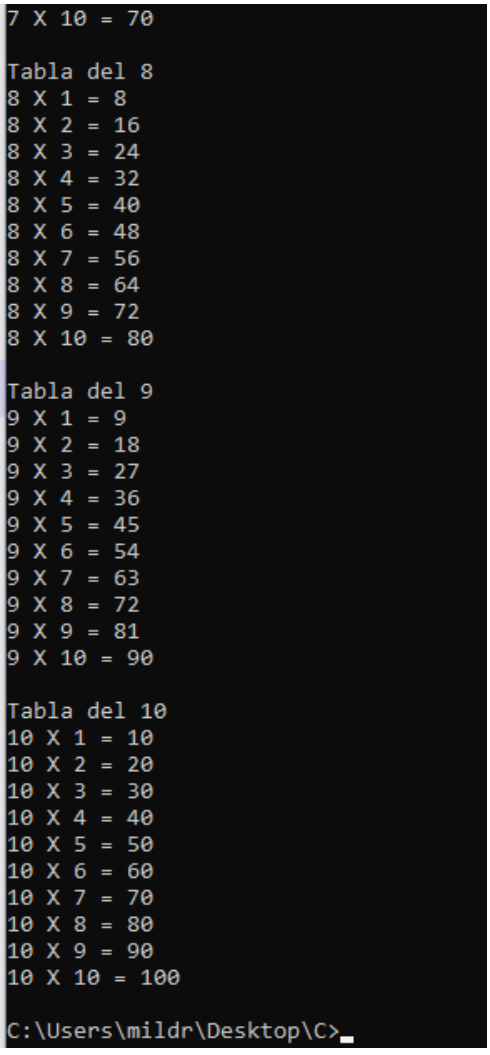
El comando “c” también útil para continuar con la ejecución completa.

Actividad 2.

La primera corrección es que en la línea 5 la i toma valores menores al 10, entonces no se muestra el valor de i cuando sea igual a 10, por lo que el operador debe cambiar, a ser menor o igual a 10

El siguiente error lo encontramos en el for, ya que primero se asigna a j con valor de 1, para a continuación poner como condición que el ciclo se ejecute si j es igual a 10, por lo que la operación nunca se realizará ni se mostrará, así que hay que cambiarlo para que la operación se realice mientras j sea menor o igual a 10.

Con los errores corregidos, funciona correctamente



Actividad 3.

```
Reading symbols from C:\Users\mildr\Desktop\C\corregir.exe...done.
(gdb) b 10
Breakpoint 1 at 0x40144a: file corregir.c, line 10.
(gdb) r
Starting program: C:\Users\mildr\Desktop\C/./corregir.exe
[New Thread 16496.0x3810]
[New Thread 16496.0x22d4]
EL TERMINO GENERICO DE LA SERIE ES: X^K/K!
N=5

Program received signal SIGSEGV, Segmentation fault.
0x76ce07fa in ungetwc () from C:\Windows\SysWOW64\msvcrt.dll
(gdb) s
Single stepping until exit from function ungetwc,
which has no line number information.
```

```
Program received signal SIGSEGV, Segmentation fault.
0x76ce07fa in ungetwc () from C:\Windows\SysWOW64\msvcrt.dll
(gdb) b 8
Breakpoint 2 at 0x40142a: file corregir.c, line 8.
(gdb) r
The program being debugged has been started already.
Start it from the beginning? (y or n) y
error return ../../gdb-7.6.1/gdb/windows-nat.c:1275 was 5
Starting program: C:\Users\mildr\Desktop\C/./corregir.exe
[New Thread 16588.0x1c8c]
[New Thread 16588.0x1c60]
EL TERMINO GENERICO DE LA SERIE ES: X^K/K!
Breakpoint 2, main () at corregir.c:8
8       printf("\nN=");
(gdb)
```

Poniendo un breakpoint en la línea 10 esperamos observar los valores que toma la variable N, para buscar cual es el fallo, ahí me percate que ni siquiera alcanzaba a ejecutar esa línea, fallaba desde antes , así que coloque un breakpoint en la línea 8, el código hasta ahí se ejecutaba bien, por lo que el problema venia en las líneas siguientes , al leerlo, me di cuenta de que faltaba el “&” para poder asignar la dirección en memoria a la variable, colocándola ya funciona correctamente.

```
C:\Users\mildr\Desktop\C>gcc corregir.c -o corregir.exe

C:\Users\mildr\Desktop\C>gcc corregir.c -o corregir.exe

C:\Users\mildr\Desktop\C>corregir.exe
EL TERMINO GENERICO DE LA SERIE ES: X^K/K!
N=5
X=3
SUM=1.840000e+001
C:\Users\mildr\Desktop\C>
```

## Conclusión

Gracias a la realización de las actividades, aprendimos a hacer uso de las diferentes herramientas y comandos que nos proporcionan distintos depuradores, en mi caso utilice gdb en Windows, el cual fue bastante fácil de entender.

Cumplir con los objetivos de la practica era importante, ya que es una parte crucial en la creación de cualquier programa, la comprobación de su funcionamiento, conocer el flujo del algoritmo y el valor de las variables en cada momento nos ayuda a detectar errores y optimizar procesos, lo cual nos ayuda a tener un funcionamiento eficiente.

## Bibliografía


- Conoce qué es Debug en programación y para qué sirve. (2020). Consultado el 13 de Diciembre 2020, de : https://www.hostgator.mx/blog/que-es-debug-en-programacion/