



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):

Pineda Cruz Tania

*No. de Equipo de
cómputo empleado:*

No aplica

No. de Lista o Brigada:

34

Semestre:

1er semestre

Fecha de entrega:

07/12/2020

Observaciones:

CALIFICACIÓN: _____

Guía práctica de estudio 10: Depuración de programas

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

Depurar un programa significa someterlo a un ambiente de ejecución controlado por medio de herramientas dedicadas a ello; permite conocer exactamente el flujo de ejecución del programa, el valor que las variables adquieren, la pila de llamadas a funciones, entre otros aspectos.

Depuración de programas escritos en C con GCC y GDB

Para depurar un programa usando las herramientas desarrolladas por GNU, éste debe compilarse con información para depuración por medio del compilador GCC.

Depuración de programas escritos en C con Dev-C++ 5.0.3.4

Dev-C++, es una IDE especializada para desarrollar programas escritos en C o C++. Incorpora un editor de textos y un compilador integrados, también posee un depurador. Dev-C++ se basa en el compilador GCC y el depurador en GDB.

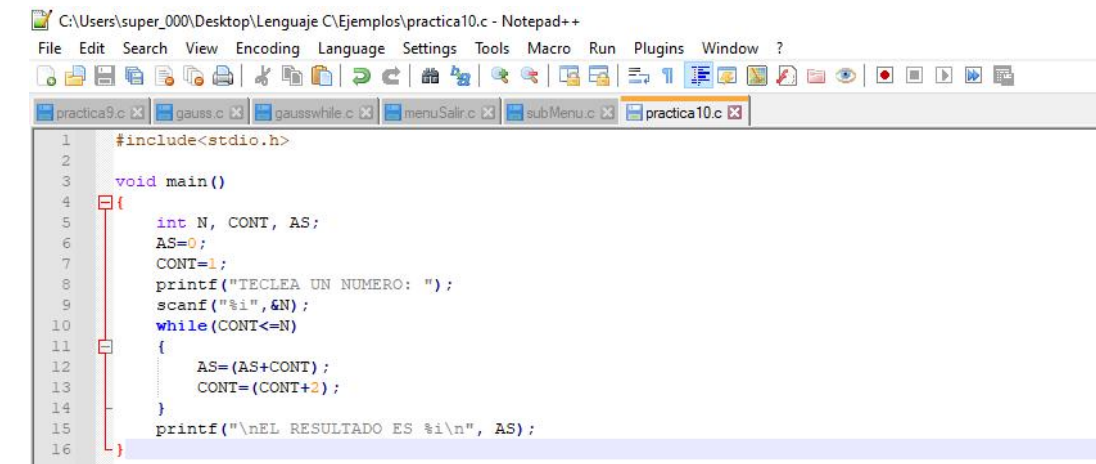
Depuración de programas escritos en C con Code::Blocks 13.12

Code::Blocks, puede basarse en las mismas herramientas GNU que Dev-C++. Permite cambiarse por otros motores de compilación si se desea. Proporciona un editor de texto, un compilador integrado, herramientas de depuración, etc.

Depuración de programas escritos en C con Xcode de Mac

Cuenta con un editor y un compilador entre otros.

Resultados



```
1 #include<stdio.h>
2
3 void main()
4 {
5     int N, CONT, AS;
6     AS=0;
7     CONT=1;
8     printf("TECLEA UN NUMERO: ");
9     scanf("%i",&N);
10    while (CONT<=N)
11    {
12        AS=(AS+CONT);
13        CONT=(CONT+1);
14    }
15    printf("\nEL RESULTADO ES %i\n", AS);
16 }
```

CA Símbolo del sistema

```
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos>gdb ./practica10.exe
GNU gdb (GDB) 7.6.1
Copyright (C) 2013 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 "mingw32".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from C:\Users\super_000\Desktop\Lenguaje C\Ejemplos\practica10.exe...done.
(gdb) list 4,8
4      {
5          int N, CONT, AS;
6          AS=0;
7          CONT=1;
8          printf("TECLEA UN NUMERO: ");
(gdb) info line 15
Line 15 of "practica10.c" starts at address 0x401467 <main+87> and ends at 0x40147b <main+107>.
(gdb) run
Starting program: C:\Users\super_000\Desktop\Lenguaje C\Ejemplos\./practica10.exe
[New Thread 11576.0x1934]
TECLEA UN NUMERO: 7

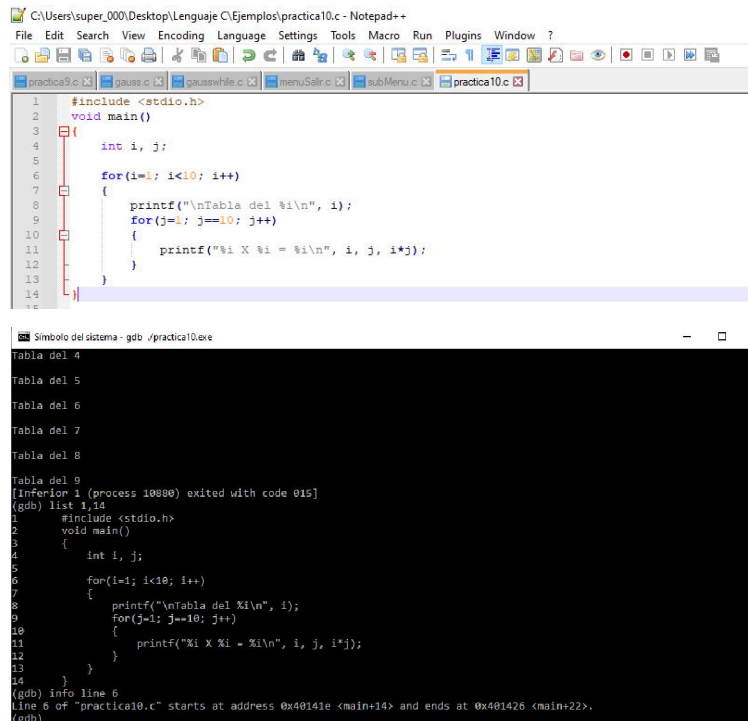
EL RESULTADO ES 16
[Inferior 1 (process 11576) exited with code 024]
(gdb) quit

C:\Users\super_000\Desktop\Lenguaje C\Ejemplos>
```

El comando *list* me permitió mostrar las líneas que yo deseaba, en este caso fue de la línea 4 a la 8. Posteriormente

El comando *info line* me permitió mostrar la información de la línea que indiqué, en este caso fue la línea 15.

El comando *run* me permitió ejecutar el programa y posteriormente, el comando *quit* me permitió terminar con la ejecución de GDB.



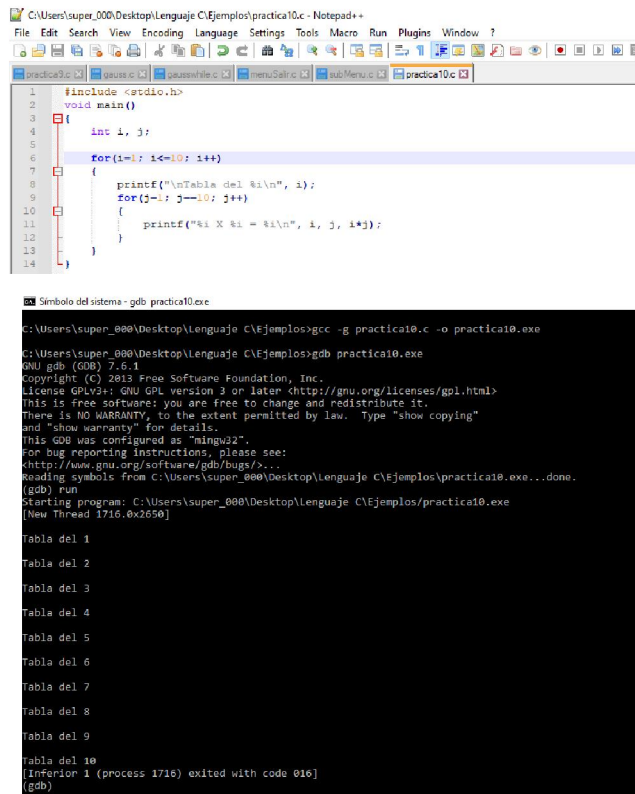
The top window is Notepad++ editing 'practica10.c'. The code is as follows:

```
1 #include <stdio.h>
2 void main()
3 {
4     int i, j;
5
6     for(i=1; i<10; i++)
7     {
8         printf("\nTabla del %i\n", i);
9         for(j=1; j==10; j++)
10         {
11             printf("%i X %i = %i\n", i, j, i*j);
12         }
13     }
14 }
```

The bottom window is a GDB terminal titled 'Simbolo del sistema - gdb ./practica10.exe'. It shows the execution of the program, printing tables from 4 to 8. It then hits a breakpoint at line 6, showing the source code context and memory addresses.

```
Tabla del 4
Tabla del 5
Tabla del 6
Tabla del 7
Tabla del 8
Tabla del 9
[Inferior 1 (process 18080) exited with code 015]
(gdb) list 1,14
1  #include <stdio.h>
2  void main()
3  {
4      int i, j;
5
6      for(i=1; i<10; i++)
7      {
8          printf("\nTabla del %i\n", i);
9          for(j=1; j==10; j++)
10          {
11              printf("%i X %i = %i\n", i, j, i*j);
12          }
13      }
14  }
(gdb) info line 6
line 6 of "practica10.c" starts at address 0x40141e <main+14> and ends at 0x401426 <main+22>.
(gdb)
```

Con el primer código, no mostraba la tabla del 10, por lo que tuve que utilizar el comando de *info line* para poder observar en donde se encontraba el error. Este estaba en la línea 6, al momento de cambiarlo y compilarlo, la tabla del 10 se mostró.



The top window is Notepad++ editing 'practica10.c'. The code is the same as before, but the loop condition on line 9 has been corrected from 'j==10' to 'j<=10'.

```
1 #include <stdio.h>
2 void main()
3 {
4     int i, j;
5
6     for(i=1; i<=10; i++)
7     {
8         printf("\nTabla del %i\n", i);
9         for(j=1; j<=10; j++)
10         {
11             printf("%i X %i = %i\n", i, j, i*j);
12         }
13     }
14 }
```

The bottom window is a GDB terminal titled 'Simbolo del sistema - gdb practica10.exe'. It shows the compilation and successful execution of the program, printing tables from 1 to 10.

```
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos>gcc -g practica10.c -o practica10.exe
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos>gdb practica10.exe
GNU gdb (GDB) 7.6.1
Copyright (C) 2013 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 "mingw32".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from C:\Users\super_000\Desktop\Lenguaje C\Ejemplos\practica10.exe...done.
(gdb) run
Starting program: C:\Users\super_000\Desktop\Lenguaje C\Ejemplos\practica10.exe
[New Thread 1716.0x2650]

Tabla del 1
Tabla del 2
Tabla del 3
Tabla del 4
Tabla del 5
Tabla del 6
Tabla del 7
Tabla del 8
Tabla del 9
Tabla del 10
[Inferior 1 (process 1716) exited with code 016]
(gdb)
```

```
*C:\Users\super_000\Desktop\Lenguaje C\Ejemplos\practica10.c - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
practica8.c gauss.c gaussH.c menuSalir.c subMenu.c practica10.c
1 #include <stdio.h>
2 #include <math.h>
3 void main()
4 {
5     int K, X, AP, N;
6     float AS;
7     printf("EL TERMINO GENERICO DE LA SERIE ES: X^K/K!");
8     printf("\nN=");
9     scanf("%d", &N);
10    printf("X=");
11    scanf("%d", &X);
12    K=0;
13    AP=1;
14    AS=0;
15    while (K<=N)
16    {
17        AS=AS+pow(X, K) / AP;
18        K=K+1;
19        AP=AP*K;
20    }
21    printf("SUM=%le", AS);
22 }
```

```
Símbolo del sistema - gdb practica10.exe
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos>gdb practica10.exe
GNU gdb (GDB) 7.6.1
Copyright (C) 2013 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 "mingw32".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from C:\Users\super_000\Desktop\Lenguaje C\Ejemplos\practica10.exe...done.
(gdb) c
The program is not being run.
(gdb) run
Starting program: C:\Users\super_000\Desktop\Lenguaje C\Ejemplos\practica10.exe
[New Thread 12744.0x1ac4]
EL TERMINO GENERICO DE LA SERIE ES: X^K/K!
N=[New Thread 12744.0x187c]
3
X-SUM=1.#INF00e+000[Inferior 1 (process 12744) exited with code 021]
(gdb) list 9,11
9         scanf("%d", &N);
10        printf("X=");
11        scanf("%d", &X);
(gdb) d 9
No breakpoint number 9.
(gdb)
```

En este último código presentado, contenía dos errores que impedía que se pudiera compilar de manera exitosa, para ello utilicé nuevamente el depurador GDB. De esa manera, pude observar y analizar en donde se encontraba el error (línea 9 y 11) y poder corregirlo para lograr su ejecución.

```
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos\practica10.c - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
practica8.c lecturaDeDatos.c sumadora.c menorEdad.c menu.c calculadora.c practica3.c gauss.c ga
1 #include <stdio.h>
2 #include <math.h>
3 void main()
4 {
5     int K, X, AP, N;
6     float AS;
7     printf("EL TERMINO GENERICO DE LA SERIE ES: X^K/K!");
8     printf("\nN=");
9     scanf("%d", &N);
10    printf("X=");
11    scanf("%d", &X);
12    K=0;
13    AP=1;
14    AS=0;
15    while (K<=N)
16    {
17        AS=AS+pow(X, K) / AP;
18        K=K+1;
19        AP=AP*K;
20    }
21    printf("SUM=%le", AS);
22 }
```

```
Símbolo del sistema
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos>gcc practica10.c -o practica10.exe
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos>practica10.exe
EL TERMINO GENERICO DE LA SERIE ES: X^K/K!
N=3
X=5
SUM=3.933333e+001
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos>
```

Conclusiones

Los depuradores nos ayudan a detectar y diagnosticar fallos en el código de los programas, así como fueron en los ejercicios propuestos. Este proceso ocurre mientras el programa se ejecuta y se analiza el código paso por paso.

En la realización de esta practica utilicé el depurador GDB, con ayuda de este pude obtener los errores que presentaban los códigos, es decir, me mostraba en que línea se encontraba el error y de esa forma poder corregirlo de manera más fácil y óptima, sin embargo, al principio me costó trabajo poder entender muy bien esa parte para buscar y encontrar, no obstante, con la práctica y la investigación detallada del mismo, pude llegar a un criterio estable y poder utilizar el depurador de manera exitosa.

Referencias

Manual de las prácticas de Fundamentos de Programación, UNAM.