



Universidad Nacional Autónoma de México

Facultad de Ingeniería

Fundamentos de programación

Práctica 10

Alumna: Rivera González Frida Alison

M.I. Marco Antonio Martínez Quintana

No. lista: 40

No. De equipo empleado: No aplica.

Semestre: 2021-1

Fecha de entrega: viernes 11 de diciembre de 2020

Grupo: 3

Observaciones:

Calificación: _____

Práctica 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. Este ambiente permite conocer exactamente el flujo de ejecución del programa, el valor que las variables adquieren, la pila de llamadas a funciones, entre otros aspectos. Es importante poder compilar el programa sin errores antes de depurarlo.

Antes de continuar, es necesario conocer las siguientes definiciones (extraídas del Glosario IEEE610) ya que son parte latente del proceso de Desarrollo de Software:

- Error. Se refiere a una acción humana que produce o genera un resultado incorrecto.
- Defecto (Fault). Es la manifestación de un error en el software. Un defecto es encontrado porque causa una Falla (failure).
- Falla (failure). Es una desviación del servicio o resultado esperado.

– List

```
Ca. Símbolo del sistema - gdb ./calculadora
This GDB was configured as "mingw32".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from C:\Users\aidee\Desktop\Lenguaje C\Ejemplos\calculadora.exe...done.
(gdb) list
1      /* Programa: Menú calculadora de números enteros */
2
3      #include <conio.h>
4      #include <stdio.h>
5
6      int main()
7      {
8          char opcion, sp=168;
9          int n1, n2, res, i, n;
10         long int Resultado=0;
(gdb) list
11         long int Factorial;
12
13         do
14         {
15             printf( "\n >>> MENU CALCULADORA <<<" );
16             printf( "\n\n  1. Sumar dos n%cmeros.", 163 );
17             printf( "\n  2. Restar dos n%cmeros.", 163 );
18             printf( "\n  3. Multiplicar dos n%cmeros.", 163 );
19             printf( "\n  4. Dividir dos n%cmeros.", 163 );
20             printf( "\n  5. Factorial de un n%cmero.",163);
(gdb) list 4,6
4      #include <stdio.h>
5
6      int main()
(gdb)
```

– b

```
Ca. Símbolo del sistema - gdb ./calculadora
(gdb) list
7      {
8          char opcion, sp=168;
9          int n1, n2, res, i, n;
10         long int Resultado=0;
11         long int Factorial;
12
13         do
14         {
15             printf( "\n >>> MENU CALCULADORA <<<" );
16             printf( "\n\n  1. Sumar dos n%cmeros.", 163 );
(gdb) b 10
Note: breakpoint 1 also set at pc 0x401424.
Breakpoint 2 at 0x401424: file calculadora.c, line 10.
(gdb)
```

- delete

```
C:\> Símbolo del sistema - gdb ./calculadora
```

```
(gdb) d 5
No breakpoint number 5.
(gdb) d 6
No breakpoint number 6.
(gdb) d 14
No breakpoint number 14.
(gdb)
```

- clear

```
C:\> Símbolo del sistema - gdb ./calculadora
```

```
(gdb) clear
No source file specified.
(gdb) _
```

- Info line

```
C:\> Símbolo del sistema - gdb ./calculadora
```

```
(gdb) clear
No source file specified.
(gdb) Quit (expect signal SIGINT when the program is resumed)
(gdb) info line 16
Line 16 of "calculadora.c" starts at address 0x401438 <main+40>
and ends at 0x40144c <main+60>.
(gdb) info line 8
Line 8 of "calculadora.c" starts at address 0x40141f <main+15>
and ends at 0x401424 <main+20>.
(gdb) _
```

- run, c, s, n

```
C:\> Símbolo del sistema - gdb ./calculadora
(gdb) run
Starting program: C:\Users\aidee\Desktop\Lenguaje C\Ejemplos/./calculadora.exe
[New Thread 21748.0x1ce8]
[New Thread 21748.0xfbc]

Breakpoint 1, main () at calculadora.c:10
10      long int Resultado=0;
(gdb) c
Continuing.

>>> MENU CALCULADORA <<<

1. Sumar dos números.
2. Restar dos números.
3. Multiplicar dos números.
4. Dividir dos números.
5. Factorial de un número.
6. Sumatoria de los primeros n números.
7. Salir.

Introduzca opción (1-7): 7
[Inferior 1 (process 21748) exited normally]
(gdb) s
The program is not being run.
(gdb) n
The program is not being run.
```

- print

```
C:\> Símbolo del sistema - gdb ./calculadora
(gdb) print
The history is empty.
(gdb) suma_acumulada
Undefined command: "suma_acumulada". Try "help".
(gdb)
```

- ignore

```
C:\> Símbolo del sistema - gdb ./calculadora
(gdb) ignore 1
Second argument (specified ignore-count) is missing.
(gdb)
```

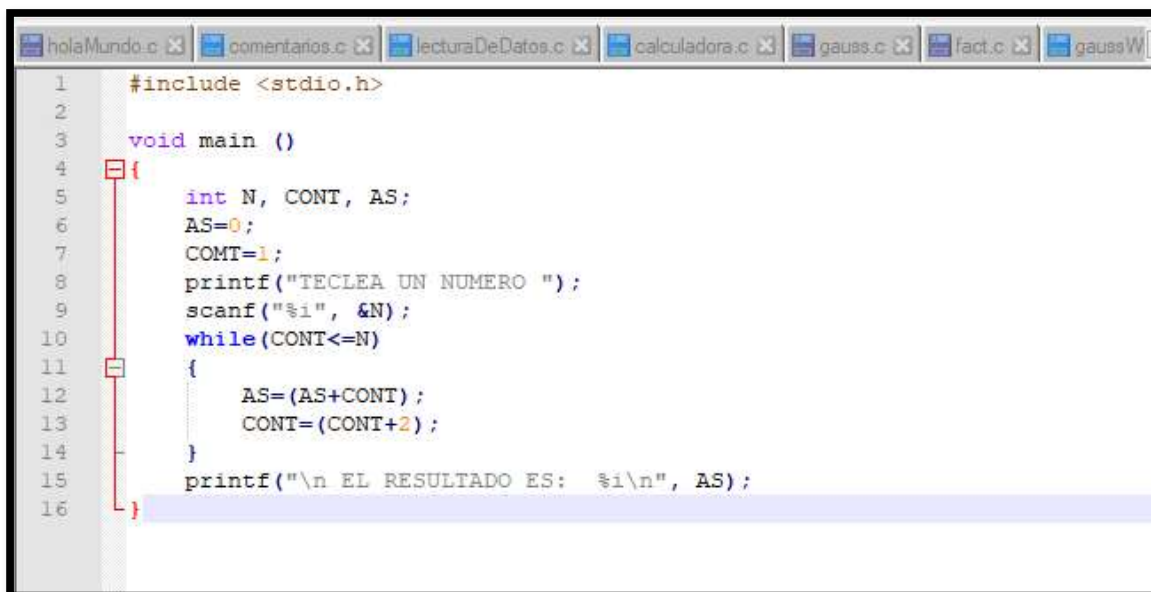
– quit

C:\. Símbolo del sistema


```
(gdb) ignore 1
Second argument (specified ignore-count) is missing.
(gdb) Quit (expect signal SIGINT when the program is resumed)
(gdb) quit
```

C:\Users\aidee\Desktop\Lenguaje C\Ejemplos>_

1.-



```
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+2);
14     }
15     printf("\n EL RESULTADO ES:  %i\n", AS);
16 }
```

 Símbolo del sistema - gdb ./teclea


```
C:\Users\aidee\Desktop\Lenguaje C\Ejemplos>gcc teclea.c -o teclea.exe

C:\Users\aidee\Desktop\Lenguaje C\Ejemplos>teclea.exe
TECLEA UN NUMERO 6

EL RESULTADO ES: 9

C:\Users\aidee\Desktop\Lenguaje C\Ejemplos>gcc -g -o teclea teclea.c

C:\Users\aidee\Desktop\Lenguaje C\Ejemplos>gdb ./teclea
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\aidee\Desktop\Lenguaje C\Ejemplos\teclea.exe...done.
(gdb) list
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)
(gdb) list
11     {
12         AS=(AS+CONT);
13         CONT=(CONT+2);
14     }
15     printf("\n EL RESULTADO ES:  %i\n", AS);
16 }
```

 Símbolo del sistema

```
(gdb) b
No default breakpoint address now.
(gdb) b 8
Breakpoint 1 at 0x40142e: file teclea.c, line 8.
(gdb) d 8
No breakpoint number 8.
(gdb) clear
No source file specified.
(gdb) info line 12
Line 12 of "teclea.c" starts at address 0x401450 <main+64> and ends at 0x401458 <main+72>.
(gdb) run
Starting program: C:\Users\aidee\Desktop\Lenguaje C\Ejemplos\./teclea.exe
[New Thread 7140.0x9c8]
[New Thread 7140.0x4e4]

Breakpoint 1, main () at teclea.c:8
8      printf("TECLEA UN NUMERO ");
(gdb) n
TECLEA UN NUMERO 9      scanf("%i", &N);
(gdb) c
Continuing.
n

EL RESULTADO ES: 0
[Inferior 1 (process 7140) exited with code 026]
(gdb) ignore
Argument required (a breakpoint number).
(gdb) quit

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


Conclusión

En esta ocasión la práctica no se me hizo tan fácil, me costó un poco más entender los conceptos. Al llevarlos a cabo al menos para los dos últimos ejercicios propuestos, al querer correr los programas, me marcaba como acceso denegado, no sé por qué con algunos programas, ocasiones, así me marca la computadora, pero pude realizar todos los puntos a ver, por el programa de mi calculadora, que, viéndolo, ya me dieron ganas de corregirlo, respecto a unas palabras, indicaciones. Los objetivos fueron cumplidos, aunque todas las actividades propuestas no.

Bibliografía

- Gutiérrez Rodríguez, Javier Jesús. Primeros pasos con GDB. Consulta: octubre de 2016. Disponible en: http://www.lsi.us.es/~javierj/ssoo_ficheros/GuiaGDB.htm
- Ferreira, Amelia. Depurador gdb. Consulta: octubre de 2016. Disponible en: <http://learnassembler.com/gdbesp.html>
- Ferreira, Amelia. Depurador gdb - uso de la opción -g de gcc. Consulta: octubre de 2016. Disponible en: <http://learnassembler.com/opc.html>
- Gutiérrez, Erik Marín. Depuración de programas Dev C++. Consulta: octubre de 2016. Disponible en: <http://programacionymetodos.blogspot.mx/2012/05/depuracion-de-programas-dev-c.html>
- González Cárdenas, Miguel Eduardo; Marín Lara, Claudia Lorena; Noguerón Pérez, Pedro. Apuntes De Computadoras Y Programación. Universidad Nacional Autónoma de México.
- Pozo Coronado, Salvador. Primeros pasos con GDB. Consulta: octubre de 2016. Disponible en: <http://www.c.conclase.net/devcpp/?cap=depurar>