



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Marco Antonio Martínez Quintana

Profesor:

Fundamentos de Programación

Asignatura:

3

Grupo:

Práctica 11

No de Práctica(s):

Santiago Díaz Edwin Jaret

Integrante(s):

*No. de Equipo de
cómputo empleado:*

No aplica

50

No. de Lista o Brigada:

2021-1

Semestre:

Lunes 14 de diciembre del 2020

Fecha de entrega:

Observaciones:

CALIFICACIÓN: _____

Depuración de Programas

Objetivos.

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.

Resultados.

Ejercicio propuesto 1

```
[Thread 14880.0x4f20 exited with code 22]
[Inferior 1 (process 14880) exited with code 026]
(gdb) l
1      #include <stdio.h>
2
3      void main () {
4      int N, CONT, AS;
5      AS = 0;
6      CONT = 1;
7      printf("Teclea un numero: ");
8      scanf("%i",&N);
9      while (CONT <= N)
10     {
(gdb) l
11         AS = (AS + CONT);
12         CONT = (CONT +2);
13     }
14     printf ("\n El resultado es %i \n", AS);
15 }
(gdb) l
Line number 16 out of range; propuesto.c has 15 lines.
(gdb) r
Starting program: C:\Users\Gerardo\Desktop\INGENIERIA UNAM\Fundamentos de Programaci3n\p1.exe
[New Thread 21328.0x3904]
[New Thread 21328.0x220c]
Teclea un numero: 1

El resultado es 1
[Thread 21328.0x220c exited with code 21]
[Inferior 1 (process 21328) exited with code 025]
(gdb) r
Starting program: C:\Users\Gerardo\Desktop\INGENIERIA UNAM\Fundamentos de Programaci3n\p1.exe
[New Thread 21884.0x20d4]
[New Thread 21884.0x5758]
Teclea un numero: 2

El resultado es 1
[Thread 21884.0x5758 exited with code 21]
[Inferior 1 (process 21884) exited with code 025]
(gdb) r
Starting program: C:\Users\Gerardo\Desktop\INGENIERIA UNAM\Fundamentos de Programaci3n\p1.exe
[New Thread 21640.0x1a9c]
[New Thread 21640.0x2afc]
Teclea un numero: 3

El resultado es 4
[Thread 21640.0x2afc exited with code 21]
[Inferior 1 (process 21640) exited with code 025]
(gdb) r
Starting program: C:\Users\Gerardo\Desktop\INGENIERIA UNAM\Fundamentos de Programaci3n\p1.exe
[New Thread 20868.0x4c08]
[New Thread 20868.0x5568]
Teclea un numero: 4

El resultado es 4
[Thread 20868.0x5568 exited with code 21]
[Inferior 1 (process 20868) exited with code 025]
(gdb) r
Starting program: C:\Users\Gerardo\Desktop\INGENIERIA UNAM\Fundamentos de Programaci3n\p1.exe
[New Thread 11932.0x3b60]
[New Thread 11932.0x3220]
Teclea un numero: 5

El resultado es 9
[Thread 11932.0x3220 exited with code 21]
[Inferior 1 (process 11932) exited with code 025]
(gdb) r
Starting program: C:\Users\Gerardo\Desktop\INGENIERIA UNAM\Fundamentos de Programaci3n\p1.exe
[New Thread 19100.0x4ad8]
```

Ejercicio propuesto 2

```
Reading symbols from p2...done.
(gdb) r
Starting program: C:\Users\Gerardo\Desktop\INGENIERIA UNAM\Fundamentos de Programación\p2.exe
[New Thread 16088.0x3fac]
[New Thread 16088.0x4024]

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
[Thread 16088.0x4024 exited with code 13]
[Inferior 1 (process 16088) exited with code 015]
(gdb) _
```

A la hora de correr el programa vemos que no se cumple lo que esperamos y además, existe más de 1 error.

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

Analizamos el código y fijamos el posible error.

```
Breakpoint 1, main () at propuesto2.c:5
5      for(i=1; i<10; i++)
(gdb) s
7      printf("\nTabla del %i\n", i);
(gdb) n

Tabla del 1
Breakpoint 2, main () at propuesto2.c:8
8      for(j=1; j==10; j++)
(gdb)
```

Al hacer breakpoint, observamos que en los casos for existe un error de sintaxis, ya que en la línea 5 se entiende que debe de llegar de 1 hasta 9 pero lo esperado es llegar hasta el 10. Y en la línea 8 no puede haber un doble igual para una variable, debe de ser `<=` o `>=` del número deseado, por ende, está mal escrito.

La resolución de este programa sería la siguiente:

```
Tabla del 7
7 X 1 = 7
7 X 2 = 14
7 X 3 = 21
7 X 4 = 28
7 X 5 = 35
7 X 6 = 42
7 X 7 = 49
7 X 8 = 56
7 X 9 = 63
7 X 10 = 70

Tabla del 8
8 X 1 = 8
8 X 2 = 16
8 X 3 = 24
8 X 4 = 32
8 X 5 = 40
8 X 6 = 48
8 X 7 = 56
8 X 8 = 64
8 X 9 = 72
8 X 10 = 80

Tabla del 9
9 X 1 = 9
9 X 2 = 18
9 X 3 = 27
9 X 4 = 36
9 X 5 = 45
9 X 6 = 54
9 X 7 = 63
9 X 8 = 72
9 X 9 = 81
9 X 10 = 90

Tabla del 10
10 X 1 = 10
10 X 2 = 20
10 X 3 = 30
10 X 4 = 40
10 X 5 = 50
10 X 6 = 60
10 X 7 = 70
10 X 8 = 80
10 X 9 = 90
10 X 10 = 100
[Thread 21032.0x2c10 exited with code 14]
[Inferior 1 (process 21032) exited with code 016]
(gdb) l
1      #include <stdio.h>
2      void main()
3      {
4          int i, j;
5          for(i=1; i<=10; i++)
6          {
7              printf("\nTabla del %i\n", i);
8              for(j=1; j<=10; j++)
9              {
10             printf("%i X %i = %i\n", i, j, i*j);
(gdb)
```

Ejercicio propuesto 3

```
The program is not being run.
(gdb) r
Starting program: C:\Users\Gerardo\Desktop\INGENIERIA UNAM\Fundamentos de Programación\pro3.exe
[New Thread 656.0x453c]
[New Thread 656.0x1eb0]
EL TERMINO GENERICO DE LA SERIE ES: X^K/K!
N=
Breakpoint 1, main () at propuesto3.c:9
9      scanf("%d",N);
(gdb) n
n
10     printf("x=");
(gdb) n
x=
Breakpoint 2, main () at propuesto3.c:11
11     scanf("%d",X);
(gdb)
```

En este ejercicio se encontró que en la línea 9 se encontraba scanf, el cual le hacía falta un ampersand "&" para determinar que se tiene que guardar en la variable N. Esto mismo sucede en la línea 11, en el scanf le hace falta un ampersand "&" y no puede guardar el valor que ingrese el usuario.

El programa corregido quedaría así:

```

C:\Users\Gerardo\Desktop\INGENIERIA UNAM\Fundamentos de Programación>gdb p3
GNU gdb (GDB) 7.8.1
Copyright (C) 2014 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-w64-mingw32".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://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 p3...done.
(gdb) r
Starting program: C:\Users\Gerardo\Desktop\INGENIERIA UNAM\Fundamentos de Programación\p3.exe
[New Thread 21152.0x3770]
[New Thread 21152.0x5244]
EL TERMINO GENERICO DE LA SERIE ES: X^K/K!
N=5
X=2
SUM=7.266667e+000[Thread 21152.0x5244 exited with code 17]
[Inferior 1 (process 21152) exited with code 021]
(gdb) l
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=");
(gdb) l
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         }
(gdb) l
21         printf("SUM=%le",AS);
22     }(gdb)

```

Conclusiones.

La depuración nos ayuda a analizar las líneas del código, poder separar el código agregando rupturas para que no se ejecute de manera secuencial. Es una buena práctica depurar de esta manera.