



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Alejandro Pimentel

Asignatura: Fundamentos de programación

Grupo: 3

No de Práctica(s): Practica 10

Integrante(s): 9955 3989
Garcés Gallardo Julio César y Velazco Gómez Noé Abimael

*No. de Equipo de
cómputo empleado:*

No. de Lista o Brigada: 14 Y 53

Semestre: Primer semestre

Fecha de entrega: 28 de Octubre del 2019

Observaciones: Muy bien

CALIFICACIÓN: 10

Tema: 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.

Introducción: Atraves de la terminal, se analizarán programas y esta nos dirá si existen errores en el programa y nos indicara precisamente en donde se encuentran. Esto a través de GDB que tiene diferentes funciones como Run, que es para ejecutar el programa, List que te imprime todo el programa, entre otros.

Desarrollo:

Actividad1: Utilizar GDB para encontrar la utilidad del programa y describir su funcionalidad.

Primero compilé y corrí el programa en GDB.

```
julio@jullo-VirtualBox ~ $ cd Descargas/
jullo@jullo-VirtualBox ~/Descargas $ ls
actividad1  rs10  rs13.c  rs15.c  'rs17(1).c'  rs19  rs6  rs9.c
actividad1.c  rs10.c  rs14  rs16  rs17.c  rs19.c  rs6.c
main  rs11.c  rs14.c  rs16.c  rs18  rs5  rs7.c
main.c  rs12.c  rs15  rs17  rs18.c  rs5.c  rs8.c
jullo@jullo-VirtualBox ~/Descargas $ gcc -g actividad1.c -o actividad1
jullo@jullo-VirtualBox ~/Descargas $ gdb ./actividad1
GNU gdb (Ubuntu 8.1-0ubuntu3.1) 8.1.0.20180409-git
Copyright (C) 2018 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.
Para las instrucciones de informe de errores, vea:
<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"...
Leyendo símbolos desde ./actividad1...hecho.
jullo@jullo-VirtualBox ~/Descargas $ gdb ./actividad1
GNU gdb (Ubuntu 8.1-0ubuntu3.1) 8.1.0.20180409-git
Copyright (C) 2018 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.
Para las instrucciones de informe de errores, vea:
<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"...
Leyendo símbolos desde ./actividad1...hecho.
(gdb) break 10
Punto de interrupción 1 at 0x768: file actividad1.c, line 10.
(gdb) run
Starting program: /home/julio/Descargas/actividad1
Ingresa un número: 8

Breakpoint 1, main () at actividad1.c:10
10      while(CONT<=N)
jullo@jullo-VirtualBox ~/Descargas $ _
```

```
Peppermint Terminal
actividad1.c
4      {
5          int N, CONT, AS;
6          AS=0;
7          CONT=1;
8          printf("Ingresa un número: ");
9          scanf("%i",&N);
10         while(CONT<=N)
11         {
12             AS=(AS+CONT);
13             CONT=(CONT+2);
14         }
15         printf("\nEl resultado es: %i\n", AS);
16     }
17
18
native process 2078 In: main L12 PC: 0x55555555476a
(gdb) next
(gdb) print
La historia est vaca.
(gdb) print AS
$1 = 0
(gdb) print CONT
$2 = 1
(gdb) _
```

Aquí, en la opción “control+x+a” y “next”, nos dimos cuenta que si ingresabamos un número, el programa hacia un ciclo donde iba ir haciendo una suma de números nones, el primero fue +3, Despues +5, luego +7 y así sucesivamente.

```
native process 2533 In: main
(gdb) start
Punto de interrupción temporal 2 at 0x722: file actividad1.c, line 4.
Starting program: /home/julio/Descargas/actividad1
Temporary breakpoint 2, main () at actividad1.c:4
(gdb) next
Ingresa un número: 8
Breakpoint 1, main () at actividad1.c:10
(gdb) display AS
native process 2533 In: main
Breakpoint 1, main () at actividad1.c:10
(gdb) display AS
(gdb) next
1: AS = 1
1: AS = 1
1: AS = 4
1: AS = 4
1: AS = 4
1: AS = 9
1: AS = 9
1: AS = 9
1: AS = 16
1: AS = 16
1: AS = 16
El resultado es: 16
1: AS = 16
(gdb) _
```

Aquí, comprobamos el programa:

```
julio@julio-VirtualBox ~/Descargas $ ./actividad1
Ingresa un número: 1

El resultado es: 1
julio@julio-VirtualBox ~/Descargas $ ./actividad1
Ingresa un número: 2

El resultado es: 1
julio@julio-VirtualBox ~/Descargas $ ./actividad1
Ingresa un número: 3

El resultado es: 4
julio@julio-VirtualBox ~/Descargas $ ./actividad1
Ingresa un número: 4

El resultado es: 4
julio@julio-VirtualBox ~/Descargas $ ./actividad1
Ingresa un número: 5

El resultado es: 9
julio@julio-VirtualBox ~/Descargas $ ./actividad1
Ingresa un número: 6

El resultado es: 9
julio@julio-VirtualBox ~/Descargas $ ./actividad1
Ingresa un número: 7

El resultado es: 16
julio@julio-VirtualBox ~/Descargas $ ./actividad1
Ingresa un número: 8

El resultado es: 16
```

Actividad2: Utilizar GDB para corregir el programa. Para compilar el código de la actividad, ejecutar: gcc -w actividad2.c -o actividad2 -lm

```
#include <stdio.h>
#include <math.h>

void main()
{
    int K, AP, N;
    double X, AS;
    printf("Ingresa cuántos términos calcular de la serie: X^K/K!");
    printf("\nN=");
    scanf("%i", N);
    printf("X=");
    scanf("%lf", X);
    K=0;
    AP=1;
    AS=0;
    while(K<=N)
    {
        AS=AS+pow(X,K)/AP;
        K=K+1;
        AP=AP*K;
    }
    printf("Resultado=%le", AS);
}
```

Al compilar y correr el programa, no leía el valor que ingresaba de N y X

```
GNU gdb (Ubuntu 8.1-0ubuntu3.1) 8.1.0.20180409-git
Copyright (C) 2018 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.
Para las instrucciones de informe de errores, vea:
<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"...
Leyendo símbolos desde ./actividad2...(no se encontraron símbolos de depuración)hecho.
(gdb) run
Starting program: /home/julio/Descargas/actividad2
Ingrese cuántos términos calcular de la serie: X^K/K!
N=4

Program received signal SIGSEGV, Segmentation fault.
0x00007ffff76b58c2 in _IO_vfscanf_internal (s=<optimized out>,
    format=<optimized out>, argptr=argptr@entry=0x7ffffffffffe5b0, errp=errp@entry=0x0)
    at vfscanf.c:1898
1898 _vfscanf.c: No existe el archivo o el directorio.
julio@julio-VirtualBox ~/Descargas $ gcc -w actividad2.c -o actividad2 -lm
julio@julio-VirtualBox ~/Descargas $ ./actividad2
Ingrese cuántos términos calcular de la serie: X^K/K!
N=3
Violación de segmento
julio@julio-VirtualBox ~/Descargas $ _
```

Esto se debía al signo "&" que hacía falta en el scanf.

```
#include <stdio.h>
#include <math.h>

void main()
{
    int K, AP, N;
    double X, AS;
    printf("Ingrese cuántos términos calcular de la serie: X^K/K!");
    printf("\nN= ");
    scanf("%i",& N);
    printf("X=");
    scanf("%lf",& X);
    K=0;
    AP=1;
    AS=0;
    while(K<=N)
    {
        AS=AS+pow(X,K)/AP;
        K=K+1;
        AP=AP*K;
    }
    printf("Resultado=%le",AS);
}
```

```

Resultado=1.300000e+01julio@julio-VirtualBox ~/Descargas $ gdb ./act2
GNU gdb (Ubuntu 8.1-0ubuntu3.1) 8.1.0.20180409-git
Copyright (C) 2018 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.
Para las instrucciones de informe de errores, vea:
<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"...
Leyendo símbolos desde ./act2...(no se encontraron símbolos de depuración)hecho.
(gdb) run
Starting program: /home/julio/Descargas/act2
Ingrese cuántos términos calcular de la serie: X^K/K!
N= 3
X=5
Resultado=3.933333e+01[Inferior 1 (process 2176) exited normally]
(gdb) _

```

Actividad3: Utilizar GDB para corregir el programa.

```

#include <stdio.h>

int main()
{
    int numero;

    printf("Ingrese un número:\n");
    scanf("%i", &numero);

    long int resultado = 1;
    while(numero>=0){
        numero--;
        resultado *= numero;
    }

    printf("El factorial de %i es %li.\n", numero, resultado);

    return 0;
}

```

Esto es lo que sucedía cuando corríamos la actividad original:

```

julio@julio-VirtualBox ~/Descargas $ gcc -g actividad3.c -o actividad3
julio@julio-VirtualBox ~/Descargas $ gdb ./actividad3
GNU gdb (Ubuntu 8.1-0ubuntu3.1) 8.1.0.20180409-git
Copyright (C) 2018 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.
Para las instrucciones de informe de errores, vea:
<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"...
Leyendo símbolos desde ./actividad3...hecho.
(gdb) run
Starting program: /home/julio/Descargas/actividad3
Ingrese un número:
3
El factorial de -1 es 0.
[Inferior 1 (process 2217) exited normally]es %li.\n", numero, resultado);
(gdb) _

```



```
Peppermint Terminal
actividad3.c
8+> 4 {
5 // Práctica 10- Con int numero;
6
7 printf("Ingrese un número:\n");
8 scanf("%i",&numero);
9
10 long int resultado = 1;
11 while(numero>=0){
12     numero--;
13     resultado *= numero;
14 }
15 int main()
16 printf("El factorial de %i es %li.\n", numero, resultado);
17
18 return 0;

native process 2266 In: main L4 PC: 0x555555554772
(gdb) start
Punto de interrupción temporal 1 at 0x555555554772: file actividad3.c, line 4.
Starting program: /home/julio/Descargas/actividad3
Temporary breakpoint 1, main () at actividad3.c:4
(gdb) _
```

En este paso cambiamos el signo del while, el resultado fue que nos regresaba el número, pero el factorial no era el resultado esperado:

```
#include <stdio.h>

int main()
{
    int numero;

    printf("Ingrese un número:\n");
    scanf("%i",&numero);

    long int resultado = 1;
    while(numero<=0){
        numero--;
        resultado *= numero;
    }

    printf("El factorial de %i es %li.\n", numero, resultado);

    return 0;
}

julio@julio-VirtualBox ~/Descargas $ gcc -g act3.c -o act3
julio@julio-VirtualBox ~/Descargas $ ./act3
Ingrese un número:
3
El factorial de 3 es 1.
```

Por ultimo volvimos a poner el signo original y le quitamos el de igual para que ahora si nos diera el resultado, pero al correr el programa imprimía el número que disminuía, o sea daba "El factorial de 1 es ...", entonces para cambiar eso guardamos el número que entro en otra variable y la nueva variable fue la que imprimimos:

```

#include <stdio.h>
#include <math.h>
int main()
{
    int numero, numero2;
    long int resultado = 1;

    printf("Ingrese un número:\n");
    scanf("%i",&numero);
    numero2=numero;

    while(numero>1){
        resultado=resultado*numero;
        numero--;
    }

    printf("El factorial de %i es %li.\n", numero2, resultado);

    return 0;
}

```

```

julio@julio-VirtualBox ~/Descargas $ gdb ./activ
GNU gdb (Ubuntu 8.1-0ubuntu3.1) 8.1.0.20180409-git
Copyright (C) 2018 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.
Para las instrucciones de informe de errores, vea:
<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"...
Leyendo símbolos desde ./activ...(no se encontraron símbolos de depuración)hecho.
(gdb) run
Starting program: /home/julio/Descargas/activ
Ingrese un número: factorial de %i es %li.\n", numero2, resultado);
5
El factorial de 5 es 120.
[Inferior 1 (process 1885) exited normally]
(gdb) _

julio@julio-VirtualBox ~/Descargas $ gcc -w act3.c -o activ
julio@julio-VirtualBox ~/Descargas $ ./activ
Ingrese un número:
5
El factorial de 5 es 120.
julio@julio-VirtualBox ~/Descargas $ _

```

Conclusión: Concluimos que el uso del GDB es útil, pero en ciertos casos nos resultó mejor correr directamente el programa en la terminal, sin embargo, para nosotros es una herramienta nueva y probablemente por eso se nos complique un poco el uso de esta, pero aun así consideramos que es la opción más óptima.

