



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: ALEJANDRO ESTEBAN PIMENTEL ALARCON

Asignatura: FUNDAMENTOS DE PROGRAMACION

Grupo: 3

No de Práctica(s): 10

Integrante(s): HINOJOSA RUIZ DIANA LAURA

*No. de Equipo de
cómputo empleado:*

No. de Lista o Brigada: 6740

Semestre: PRIMER SEMESTRE

Fecha de entrega: 28 OCTUBRE 2019

Observaciones: Tarde entrega.

CALIFICACIÓN: 6

Depuración de programas

Objetivo:

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

Introducción:

Se analizaran distintos programas directamente de la terminal, la cual nos indicara si existen errores o no y donde es que se encuentran. Todo esto se logra con gdb que abre una ventana con distintas funciones que sirven para ejecutar el programa, para imprimirlo, o recorrerlo línea por línea.

Desarrollo:

La clase la comenzamos haciendo un ejemplo y trabajando con él desde la terminal. Exploramos por primera vez algunas funciones nuevas, incluyendo gdb, run, list o next/n. Al no poder llevar a cabo bien todo lo que queríamos tuvimos que abrir otro servidores que corresponde a Linux, por eso aparece "SAMBA".

```
Last login: Mon Nov  4 09:01:27 on console
[Mongolia22:~ fp03alu24$ servidor
[Mongolia22:~ fp03alu24$ ssh fp03alu24@192.168.2.200
The authenticity of host '192.168.2.200 (192.168.2.200)' can't be established.
RSA key fingerprint is SHA256:jTgFsbnpP7IaIpwchV27DaUa9i2pvAVVZwZzbIneOF8.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.2.200' (RSA) to the list of known hosts.
[fp03alu24@192.168.2.200's password:
Last login: Mon Oct 14 10:06:07 2019 from marruecos16.red

SAMBA

-bash: aviso: setlocale: LC_CTYPE: no se puede cambiar el local (UTF-8)
[[fp03alu24@samba ~]$ cd
.cache/  Escritorio/ .gnome2/  .mozilla/
[[fp03alu24@samba ~]$ ls
Escritorio  actividad1.c  actividad2.c  actividad3.c  ejemplo1  ejemplo1.c
[[fp03alu24@samba ~]$ gcc -std=c99 -g ejemplo1.c -o ejemplo1
[[fp03alu24@samba ~]$ ./ejemplo1
Primero texto solo
Luego podemos poner un entero: 10
También podemos poner un caracter: B
Un numero real: 89.80
Violación de segmento ('core' generado)
[[fp03alu24@samba ~]$ gdb ./ejemplo1
```

```

[fp03alu24@samba ~]$ gdb ./ejemplo1
GNU gdb (GDB) Fedora (7.4.50.20120120-42.fc17)
Copyright (C) 2012 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-redhat-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /users/fp03/fp03alu24/ejemplo1...done.
(gdb) run
Starting program: /users/fp03/fp03alu24/ejemplo1
Primero texto solo
Luego podemos poner un entero: 10
También podemos poner un caracter: B
Un numero real: 89.80

Program received signal SIGSEGV, Segmentation fault.
0x000000000040060c in main (argc=19, argv=0x1100000010) at ejemplo1.c:21
21          lista[i] = i;
Missing separate debuginfos, use: debuginfo-install glibc-2.15-37.fc17.x86_64
(gdb)

```

ejemplo1.c

```

ejemplo1.c
10      long int suma = 0;
11      double promedio;
12
13      // Mostramos texto y valores
14      printf("Primero texto solo\n");
15      printf("Luego podemos poner un entero: %i\n", numero);
16      printf("También podemos poner un caracter: %c\n", caracter);
17      printf("Un numero real: %.2f\n", numeroReal);
18
19      // Podemos llenar la lista con valores
20      for(int i = numero ; i >= numero ; i++){
> 21          lista[i] = i;
22      }
23
24      // Y ahora podemos hacer calculos con la lista
25      for(int i = numero ; i >= numero ; i++){
26          suma += lista[i];

```

child process 28847 In: main

Line: 21 PC: 0x40060c

(gdb)

Actividad 1:

Utilizar gdb para encontrar la utilidad del programa y describir su funcionalidad.

Recorrí línea por línea el programa para saber lo que se hace en cada paso.

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

```
Un numero real: 89.80

Program received signal SIGSEGV, Segmentation fault.
0x000000000040060c in main (argc=19, argv=0x1100000010) at ejemplo1.c:21
21          lista[i] = i;
Missing separate debuginfos, use: debuginfo-install glibc-2.15-37.fc17.x86_64
[fp03alu24@samba ~]$ ls
Escritorio  actividad1.c  actividad2.c  actividad3.c  ejemplo1  ejemplo1.c
[fp03alu24@samba ~]$ gcc -std=c99 -g actividad1.c -o actividad1
[fp03alu24@samba ~]$ ./actividad1
Ingresa un número: 6

El resultado es: 9
[fp03alu24@samba ~]$ gdb ./actividad1
GNU gdb (GDB) Fedora (7.4.50.20120120-42.fc17)
Copyright (C) 2012 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-redhat-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /users/fp03/fp03alu24/actividad1...done.
(gdb) run
Starting program: /users/fp03/fp03alu24/actividad1
Ingresa un número: 6

El resultado es: 9
[Inferior 1 (process 29006) exited with code 024]
Missing separate debuginfos, use: debuginfo-install glibc-2.15-37.fc17.x86_64
[fp03alu24@samba ~]$ gcc []
```

```
actividad1.c
1  #include <stdio.h>
2
3  void main()
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
19
Ingresa un número: 7
El resultado es: 16 In: main Line: 16 PC: 0x4005af
(gdb) start
Temporary breakpoint 1 at 0x400554: file actividad1.c, line 6.
Starting program: /users/fp03/fp03alu24/actividad1

Temporary breakpoint 1, main () at actividad1.c:6
(gdb) next n
next n
(gdb) █
```

Actividad 2:

Utilizar gdb para corregir el programa.

Para compilar el código de la actividad hay que ejecutar: `gcc -w actividad2.c -o actividad2 -lm`.

```
/home/programacion

ruus1@DESKTOP-1V77FA0 /home/programacion
$ gcc -w actividad2.c -o actividad2 -lm

ruus1@DESKTOP-1V77FA0 /home/programacion
$ ./actividad2
Ingrese cuántos términos calcular de la serie: X^K/K!
N=7

ruus1@DESKTOP-1V77FA0 /home/programacion
$ gcc -std=c99 -g actividad2.c -o actividad2

ruus1@DESKTOP-1V77FA0 /home/programacion
$ gdb ./actividad2
GNU gdb (GDB) (Cygwin 8.1.1-1) 8.1.1
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-pc-cygwin".
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 ./actividad2...done.
(gdb) run
Starting program: /home/programacion/actividad2
[New Thread 5852.0x346c]
[New Thread 5852.0x1640]
Ingrese cuántos términos calcular de la serie: X^K/K!
N=9
[Thread 5852.0x1640 exited with code 3221225477]
[Inferior 1 (process 5852) exited with code 03000000000005]
(gdb)
```

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

Recorrí el programa línea por línea y llegue a que le hacía falta el signo & en los scanf.

actividad2.c

```
10      scanf("%i",&N);
11      printf("X=");
12      scanf("%lf",&X);
13      K=0;
14      AP=1;
15      AS=0;
16      while(K<=N)
17      {
18          AS=AS+pow(X,K)/AP;
19          K=K+1;
20          AP=AP*K;
21      }
22      printf("Resultado=%le",AS);
23  }
24
```

exec No process in:

L?? PC: ??

```
[New Thread 7708.0x26bc]
[New Thread 7708.0x2b00]
[Thread 7708.0x3738 exited with code 3221225477]
[Thread 7708.0x26bc exited with code 3221225477]
[Inferior 1 (process 7708) exited with code 0300000000005]
(gdb) next
The program is not being run.
(gdb) |
```

#include <stdio.h>

#include <math.h>

void main()

```
{
    int K, AP, N;
    double X, AS;
    printf("Ingrese cuantos terminos 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);
}
```



```
/home/programacion
ruus1@DESKTOP-1V77FA0 /home/programacion
$ gdb ./actividad21
GNU gdb (GDB) (Cygwin 8.1.1-1) 8.1.1
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-pc-cygwin".
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 ./actividad21...done.
(gdb) run
Starting program: /home/programacion/actividad21
[New Thread 6404.0x2668]
[New Thread 6404.0x3b14]
Ingrese cuántos términos calcular de la serie: X^K/K!
N=3
X=4
Resultado=2.366667e+01[Thread 6404.0x3b14 exited with code 22]
[Inferior 1 (process 6404) exited with code 026]
(gdb) quit
```

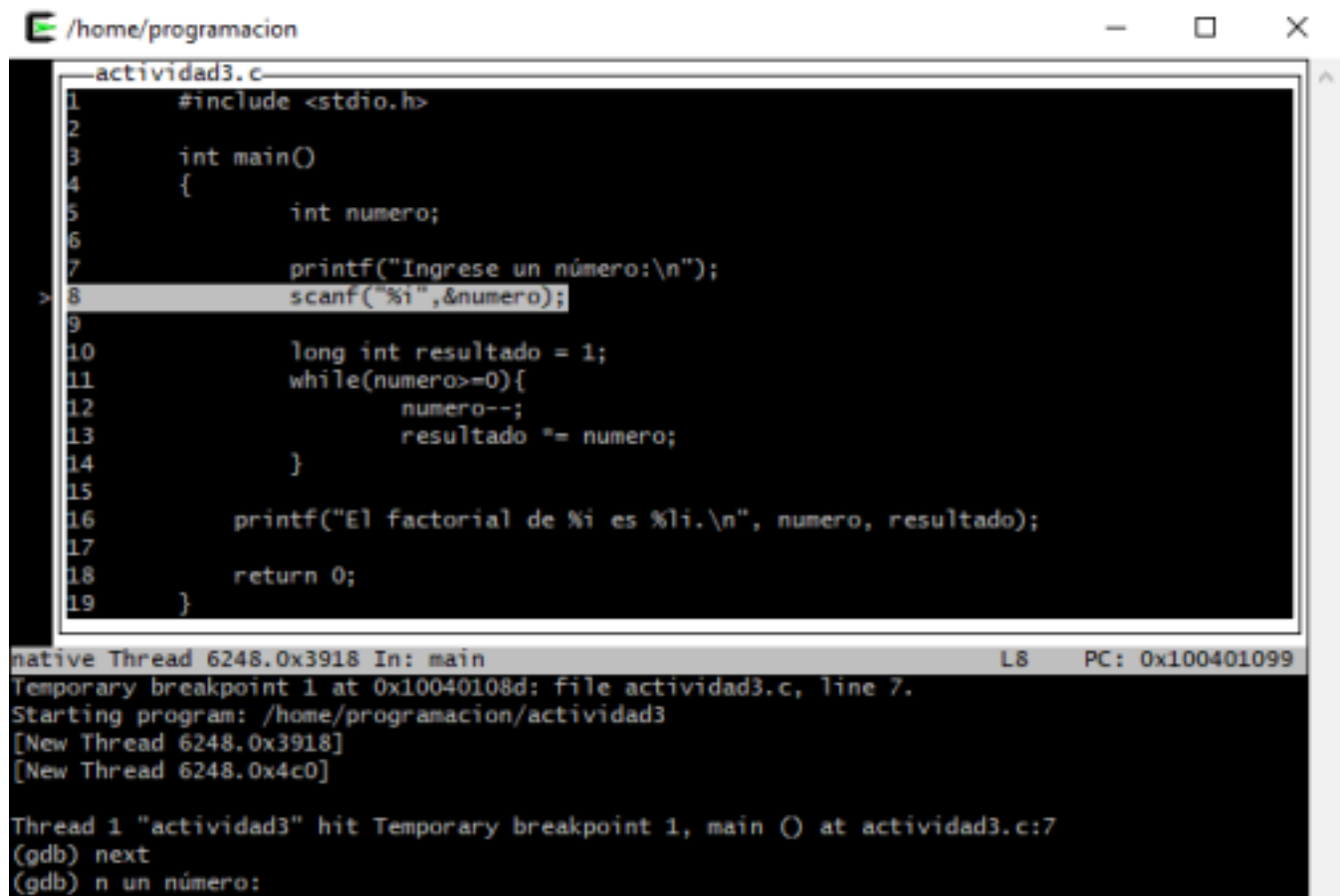
Actividad 3:

Utilizar GDB para corregir el programa.

Esto es lo que pasa cuando se corre la actividad 3.

```
/home/programacion
ruus1@DESKTOP-1V77FA0 /home/programacion
$ gdb ./actividad3
GNU gdb (GDB) (Cygwin 8.1.1-1) 8.1.1
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-pc-cygwin".
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 ./actividad3...done.
(gdb) run
Starting program: /home/programacion/actividad3
[New Thread 6360.0xc10]
[New Thread 6360.0x2258]
Ingrese un número:
7
El factorial de -1 es 0.
[Thread 6360.0x2258 exited with code 0]
[Inferior 1 (process 6360) exited normally]
(gdb)
```


Se recorre el programa línea por línea y se detecta el problema.



```

/home/programacion
-actividad3.c
1  #include <stdio.h>
2
3  int main()
4  {
5      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
16     printf("El factorial de %i es %li.\n", numero, resultado);
17
18     return 0;
19 }

native Thread 6248.0x3918 In: main L8 PC: 0x100401099
Temporary breakpoint 1 at 0x10040108d: file actividad3.c, line 7.
Starting program: /home/programacion/actividad3
[New Thread 6248.0x3918]
[New Thread 6248.0x4c0]

Thread 1 "actividad3" hit Temporary breakpoint 1, main () at actividad3.c:7
(gdb) next
(gdb) n un número:
```

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;
}

```

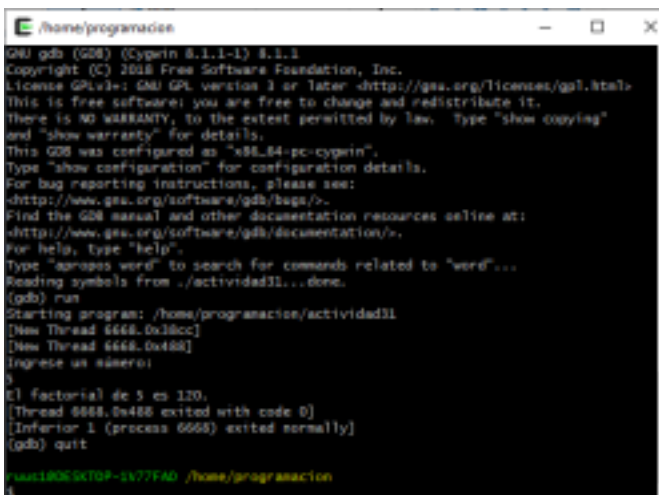
```

ruus1@DESKTOP-1V77FA0 /home/programacion
$ gcc -g actividad31.c -o actividad31 -lm

ruus1@DESKTOP-1V77FA0 /home/programacion
$ ./actividad31
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:



```

/home/programacion
GNU gdb (GDB) (Cygwin 8.1.1-1) 8.1.1
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-pc-cygwin".
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 ./actividad31...done.
(gdb) run
Starting program: /home/programacion/actividad31
[New Thread 6668.0x38cc]
[New Thread 6668.0x488]
Ingrese un número:
3
El factorial de 3 es 120.
[Thread 6668.0x488 exited with code 0]
[Inferior 1 (process 6668) exited normally]
(gdb) quit

ruus1@DESKTOP-1V77FA0 /home/programacion

```

```

1  #include <stdio.h>
2  #include <math.h>
3
4  int main()
5  {
6      int numero, numero2;
7      long int resultado = 1;
8
9      printf("Ingrese un número:\n");
10     scanf("%i",&numero);
11     numero2=numero;
12
13     while(numero>1){
14         resultado = resultado*numero;
15         numero--;
16     }
17
18
19     printf("El factorial de %i es %li.\n", numero2, resultado);
20
21     return 0;
22 }

```

Conclusión:

GDB es una herramienta muy útil para analizar a detalle el funcionamiento de un programa, lo que facilita su comprensión, así como el encontrar posibles fallas en el mismo, 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.