



Laboratorios de computación Salas A y B.

<i>Profesor:</i>	Alejandro Esteban Pimentel Alarcon
<i>Asignatura :</i>	Fundamentos de programación
<i>Grupo:</i>	3
<i>No de Práctica(s):</i>	10
<i>Integrante(s):</i>	Flores Rodriguez Ricardo Manuel 317189795 Valencia Moya José Alejandro 317098543
<i>No. de Equipo de cómputo empleado:</i>	51 y 52
<i>No. de Lista o Brigada:</i>	12 y 51
<i>Semestre:</i>	Primero 2019-2020
<i>Fecha de entrega:</i>	28 de Octubre del 2019
<i>Observaciones:</i>	

CALIFICACIÓN: _____

Introducción

El alumno entrará en Gdb para poder interactuar con este programa y pueda localizar ,analizar y corregir errores en los problemas.

GDB (Gnu Project Debugger) es una herramienta que permite entre otras cosas, correr el programa con la posibilidad de detenerlo cuando se cumple cierta condición, avanzar paso a paso, analizar qué ha pasado cuando un programa se detiene o cambiar algunas cosas del programa como el valor de las variables.

GDB es una herramienta muy poderosa que nos ayudará a encontrar esos errores difíciles, por ejemplo cuando los punteros no apuntan a dónde estamos pensando.

Objetivo

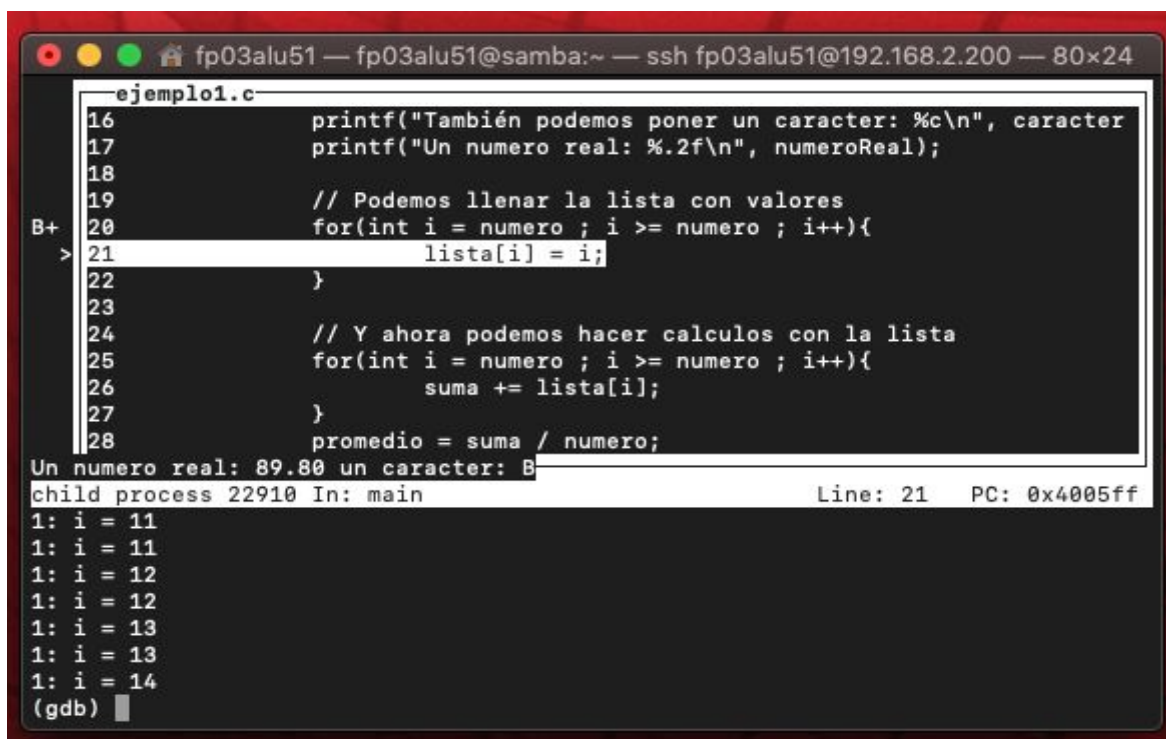
Aprender las técnicas básicas de depuración de programas en C para revisar de manera precisa el flujo de ejecución de un programa y el valor de las variables; en su caso, corregir posibles errores.

Actividad 1

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

-En esta actividad el profesor nos fue indicando todos los pasos para poder entrar a gdb y nos dio una serie de pasos para utilizarlo de manera correcta.

- En donde entramos al programa y pasamos de renglón en renglón hasta encontrar error.



```
fp03alu51 — fp03alu51@samba:~ — ssh fp03alu51@192.168.2.200 — 80x24
ejemplo1.c
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];
27      }
28      promedio = suma / numero;
Un numero real: 89.80 un caracter: B
Child process 22910 In: main                               Line: 21   PC: 0x4005ff
1: i = 11
1: i = 11
1: i = 12
1: i = 12
1: i = 13
1: i = 13
1: i = 14
(gdb)
```

fp03alu51 — fp03alu51@samba:~ — ssh fp03alu51@192.168.2.200 — 80x24

```
ejemplo1.c
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
B+> 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];
27         }
28         promedio = suma / numero;
Un numero real: 89.80 un caracter: B
child process 22910 In: main                               Line: 20   PC: 0x4005f7
```

```
Breakpoint 2, main (argc=1, argv=0x7ffffffe398) at ejemplo1.c:20
(gdb) print list
No symbol "list" in current context.
(gdb) print lista
$1 = {-163754450, 0, 4195102, 0, -1, 0, -7536, 32767, -7520, 32767}
(gdb) display
(gdb)
```

```
ejemplo1.c
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
9  11         // 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];
27         }
Un numero real: 89.80 un caracter: B
child process 22230 In: main                               Line: 21   PC: 0x4005ff
```

```
(gdb) start
Temporary breakpoint 1 at 0x400542: file ejemplo1.c, line 6.
Starting program: /users/fp03/fp03alu51/ejemplo1

Temporary breakpoint 1, main (argc=1, argv=0x7ffffffe398) at ejemplo1.c:6
Missing separate debuginfos, use: debuginfo-install glibc-2.15-37.fc17.x86_64
(gdb)
```

Actividad 2

Utilizar GDB para corregir el programa.

-Al ejecutar el programa muestra que el programa tiene un problema y al revisar se muestran estos en la línea 8.

```
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/fp03alu23/act...done.
(gdb) start
Temporary breakpoint 1 at 0x400664: file actividad2.c, line 8.
Starting program: /users/fp03/fp03alu23/act
Temporary breakpoint 1, main () at actividad2.c:8
```

```
actividad2.c
1  #include <stdio.h>
2  #include <math.h>
3
4  void main()
5  {
6      int K, AP, N;
7      double X, AS;
8  B-> 8  printf("Ingrese cuántos términos calcular de la serie: X"
9      printf("\nN=");
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;
    }
}

child process 23537 In: main Line: 8 PC: 0x400664
(gdb) start
Temporary breakpoint 1 at 0x400664: file actividad2.c, line 8.
Starting program: /users/fp03/fp03alu23/act
Temporary breakpoint 1, main () at actividad2.c:8
Missing separate debuginfos, use: debuginfo-install glibc-2.15-37.fc17.x86_64
(gdb)
```


-Una vez ya corregido los errores debemos corroborar que el programa ya no tenga más errores y compilarlo.

```
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/fp03alu23/act22...done.
(gdb) run
Starting program: /users/fp03/fp03alu23/act22
Ingrese cuantos terminos calcular de la serie: X*X/X!
N=4
X=7
Resultado= 1.897883e+02[Inferior 1 (process 26846) exited normally]
Missing separate debuginfos, use: debuginfo-install glibc-2.15-37.fc17.x86_64
(gdb) quit
[fp03alu23@samba ~]$
```

Actividad 3

Utilizar GDB para corregir el programa.

-Aquí nos dice que el programa tiene problemas en la línea 7 y debemos de corregir para que el programa corra al 100.

```
GNU gdb (Ubuntu 8.2-0ubuntu1) 8.2
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.
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"...
Leyendo simbolos desde ./act3...hecho.
(gdb) break 11
Punto de interrupción 1 at 0x11a8: file actividad3.c, line 11.
(gdb) run
Starting program: /home/vanessa/Escritorio/act3
Ingrese un número:
6

Breakpoint 1, main () at actividad3.c:11
11      while(numero>=0){
(gdb)
```

```
actividad3.c
B+> 4 {
5     int numero;
6
7     printf("Ingrese un número:\n");
8     scanf("%l",&numero);
9
10    long int resultado = 1;
11    while(numero>=0){
12        numero--;
13        resultado *= numero;
14    }
15
16    printf("El factorial de %l es %ll.\n", numero, resultado);
17
18    return 0;
19 }
20
21
22
23
24
25

native process 2261 In: main
(gdb) start
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Punto de interrupción temporal 2 at 0x5555555516d: file actividad3.c, line 4.
Starting program: /home/vanessa/Escritorio/act3

Temporary breakpoint 2, main () at actividad3.c:4
(gdb) print numero
$1 = 21845
(gdb) display numero
1: numero = 21845
(gdb) 
```

-Ya corregido los errores tenemos como resultado esto:

```
Ingrese un número:
3
El factorial de 3 es 6
vanessa@Titan:~/Escritorio$ ./ac3
Ingrese un número:
5
El factorial de 5 es 120
vanessa@Titan:~/Escritorio$ ./ac3
Ingrese un número:
26
El factorial de 26 es -1569523520172457984
vanessa@Titan:~/Escritorio$ ./ac3
Ingrese un número:
16
El factorial de 16 es 20922789888000
vanessa@Titan:~/Escritorio$ 
```

Conclusión

Es muy importante saber manejar gdb por que te ayuda a encontrar los errores de los programas y te dice con claridad en dónde fue dónde te equivocaste o hay que corregir. Me parece muy importante el uso de esta herramienta para los programa en c.