



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Alejandro Esteban Pimentel Alarcon

Profesor:

Fundamentos de programación

Asignatura:

3

Grupo:

10

No de Práctica(s):

Oscar García García
Ana Cristina Romero Rojas

Integrante(s):

*No. de Equipo de
cómputo empleado:*

2712
1147

No. de Lista o Brigada:

2020-1

Semestre:

28/10/19

Fecha de entrega:

Observaciones: No muestran muchas evidencias de usar GDB,
corrigen los errores según las salidas del compilador

CALIFICACIÓN: 8

Objetivo: Aprender las técnicas básicas de depuración 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.

Ejemplo:

```
#include <stdio.h>

int main(int argc, char * argv[]) {

    // Asignamos variables
    int numero = 10;
    int lista[numero];
    char caracter = 'B';
    float numeroReal = 89.8;
    long int suma = 0;
    double promedio;

    // Mostramos texto y valores
    printf("Primero texto solo\n");
    printf("Luego podemos poner un entero: %i\n", numero);
    printf("También podemos poner un caracter: %c\n", caracter);
    printf("Un numero real: %.2f\n", numeroReal);

    // Podemos llenar la lista con valores
    for(int i = numero ; i >= numero ; i++){
        lista[i] = i;
    }

    // Y ahora podemos hacer calculos con la lista
    for(int i = numero ; i >= numero ; i++){
        suma += lista[i];
    }
    promedio = suma / numero;
    printf("La suma es: %li\n", suma);
    printf("El promedio es: %lf\n", promedio);

    return 0;
}
```

```
Last login: Mon Oct 14 09:18:43 on ttys000
Nigeria29:~ fp03alu15$ servidor
Nigeria29:~ fp03alu15$ ssh fp03alu15@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:jTgFsbvP7IaIpwchV27DaUa9i2pvAVVZwZzbIneOF8.
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.
fp03alu15@192.168.2.200's password:
```

SAMBA

```
-bash: aviso: setlocale: LC_CTYPE: no se puede cambiar el local (UTF-8)
[fp03alu15@samba ~]$ █
```

```

Escritorio actividad1.c actividad2.c actividad3.c ejemplo1.c
[fp03alu15@samba ~]$ gcc -std=c99 -g ejemplo1.c -o ejemplo1
[fp03alu15@samba ~]$ gdb ./ejemplo1
-bash: gdb./ejemplo1: No existe el fichero o el directorio
[fp03alu15@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/fp03alu15/ejemplo1...done.
(gdb) run

```

```

[Nigeria29:Downloads fp03alu15$ gcc ejemplo1.c -o main
[Nigeria29:Downloads fp03alu15$ ./main
Primero texto solo
Luego podemos poner un entero: 10
También podemos poner un caracter: B
Un numero real: 89.80
Segmentation fault: 11

```

```

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) list
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++){
(gdb)

```

Inferior 1 [process 21650] will be killed.

Quit anyway? (y or n) y

[No Source Available]

```

ejemplo1.c
1      #include <stdio.h>
2
3      int main(int argc, char * argv[]) {
4
5          // Asignamos variables
6      B+> int numero = 10;
7          int lista[numero];
8          char caracter = 'B';
9          float numeroReal = 89.8;
10         long int suma = 0;
11         double promedio;
12
13         // Mostramos texto y valores

```

child process 22198 In: main Line: 6 PC: 0x400542

```

(gdb) start
Temporary breakpoint 1 at 0x400542: file ejemplo1.c, line 6.
Starting program: /users/fp03/fp03alu48/Escritorio/ejemplo1

Temporary breakpoint 1, main (argc=1, argv=0x7fffffff378) at ejemplo1.c:6
Missing separate debuginfos, use: debuginfo-install glibc-2.15-37.fc17.x86_64
(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 22198 In: main Line: 20 PC: 0x40060f

```

(gdb) start
Temporary breakpoint 1 at 0x400542: file ejemplo1.c, line 6.
Starting program: /users/fp03/fp03alu48/Escritorio/ejemplo1

Temporary breakpoint 1, main (argc=1, argv=0x7fffffff378) at ejemplo1.c:6
Missing separate debuginfos, use: debuginfo-install glibc-2.15-37.fc17.x86_64
(gdb) █

```

```

12
13      // Mostramos texto y valores
3 14      printf("Primero texto solo\n");
15      printf("Luego podemos poner un entero: %i\n", numero);
5 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;
29      printf("La suma es: %li\n", suma);
19 30      printf("El promedio es: %lf\n", promedio);
Un numero real: 89.80 un caracter: B
child process 22945 In: main                               Line: 21   PC: 0x4005ff
The program is not being run.
Starting program: /users/fp03/fp03alu48/Escritorio/ejemplo1

Temporary breakpoint 1, main (argc=1, argv=0x7fffffffe378) at ejemplo1.c:6
Missing separate debuginfos, use: debuginfo-install glibc-2.15-37.fc17.x86_64
(gdb) p i
$1 = 10 i
(gdb) print lista
print lista
$2 = {-163754450, 0, 4195102, 0, -1, 0, -7568, 32767, -7552, 32767}
(gdb) █

```

```

3 14      printf("Primero texto solo\n");
15      printf("Luego podemos poner un entero: %i\n", numero);
5 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
>
>      // 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;
29      printf("La suma es: %li\n", suma);
19 30      printf("El promedio es: %lf\n", promedio);
Un numero real: 89.80 un caracter: B
child process 22945 In: main                               Line: 21   PC: 0x4005ff
The program is not being run.
Starting program: /users/fp03/fp03alu48/Escritorio/ejemplo1
Missing separate debuginfos, use: debuginfo-install glibc-2.15-37.fc17.x86_64
(gdb) print lista
print lista
$2 = {-163754450, 0, 4195102, 0, -1, 0, -7568, 32767, -7552, 32767}
(gdb) display i
display i
1: i = 10
(gdb) display lista
display lista
2: lista = {-163754450, 0, 4195102, 0, -1, 0, -7568, 32767, -7552, 32767}
(gdb) █

```

```

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     for(int i = numero ; i >= numero ; i++){
29         for(int i = numero ; i >= numero ; i++){
30             for(int i = numero ; i >= numero ; i++){
31                 lista[i] = i; s: %lf\n", promedio);

```

Un numero real: 89.88 un caracter: 0

child process 22945 In: main

Line: 21 PC: 0x4005ff

The program is not being run.

Starting program: /users/fp03/fp03alu48/Escritorio/ejemplo1

Missing separate debuginfos, use: debuginfo-install glibc-2.15-37.fc17.x86_64

(gdb) print lista

print lista

\$2 = {-163754450, 0, 4195102, 0, -1, 0, -7568, 32767, -7552, 32767}

(gdb) display i

display lista

(gdb)

1: i = 10

(gdb) n

n

2: lista = {-163754450, 0, 4195102, 0, -1, 0, -7568, 32767, -7552, 32767}

1: i = 11

(gdb) n

n

2: lista = {-163754450, 0, 4195102, 0, -1, 0, -7568, 32767, -7552, 32767}

1: i = 11

(gdb) █

```

20         for(int i = numero ; i >= numero ; i++){
21         for(int i = numero ; i >= numero ; i++){
22         for(int i = numero ; i >= numero ; i++){
23         for(int i = numero ; i >= numero ; i++){
24         for(int i = numero ; i >= numero ; i++){
25         for(int i = numero ; i >= numero ; i++){
26         for(int i = numero ; i >= numero ; i++){
27         lista[i] = i; scritorio/ejemplo1

```

Line: 21 PC: 0x4005ff

Missing separate debuginfos, use: debuginfo-install glibc-2.15-37.fc17.x86_64

(gdb) print lista

print lista

\$2 = {-163754450, 0, 4195102, 0, -1, 0, -7568, 32767, -7552, 32767}

(gdb) display i

display lista

(gdb)

1: i = 10

1: i = 11

(gdb) n

2: lista = {-163754450, 0, 4195102, 0, -1, 0, -7568, 32767, -7552, 32767}

(gdb) n

2: lista = {-163754450, 0, 4195102, 0, -1, 0, -7568, 32767, -7552, 32767}

1: i = 12

(gdb) n

2: lista = {-163754450, 0, 4195102, 0, -1, 0, -7568, 32767, -7552, 32767}

1: i = 13

(gdb) n

2: lista = {-163754450, 0, 4195102, 0, -1, 0, -7568, 32767, -7552, 32767}

1: i = 13

(gdb) █

```

((gdb) break 20
Breakpoint 1 at 0x4005f7: file ejemplo1.c, line 20.
((gdb) run
Starting program: /users/fp03/fp03alu48/Escritorio/ejemplo1
Primero texto solo
Luego podemos poner un entero: 10
También podemos poner un caracter: B
Un numero real: 89.80

Breakpoint 1, main (argc=1, argv=0x7fffffffe378) at ejemplo1.c:20
20         for(int i = numero ; i >= numero ; i++){
Missing separate debuginfos, use: debuginfo-install glibc-2.15-37.fc17.x86_64
(gdb) █

```

Actividad 1

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

En la actividad 1, al ponerlo en GDB lo primero que hace es detectarme el **error** por el cual no puede correr el programa

El error es: debe ser

int main En el primer programa no había error

En vez de

void main

```

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 }

```

input

Compilation failed due to following error(s).

```

main.cpp:3:11: error: '::main' must return 'int'
void main()
    ^

```

Tras corregirlo, ya es posible compilar y correr

```
1  #include <stdio.h>
2
3  int 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 }
```

input

Ingresa un número:

Ahora veamos la funcionalidad del programa

```
Ingresa un número: 3  
  
El resultado es: 4  
  
...Program finished with exit code 0  
Press ENTER to exit console.□
```

```
Ingresa un número: 4  
  
El resultado es: 4  
  
...Program finished with exit code 0  
Press ENTER to exit console.□
```

```
Ingresa un número: 5  
  
El resultado es: 9  
  
...Program finished with exit code 0  
Press ENTER to exit console.□
```

```
Ingresa un número: 6  
  
El resultado es: 9  
  
...Program finished with exit code 0  
Press ENTER to exit console.□
```

```
Ingresa un número: 7

El resultado es: 16

...Program finished with exit code 0
Press ENTER to exit console.[]
```

```
Ingresa un número: 9

El resultado es: 25

...Program finished with exit code 0
Press ENTER to exit console.[]
```

Tras ver varios resultados del programa, la funcionalidad del programa es recibir un número y en otra variable llamada AS se va guardando el nuevo resultado de la suma de AS más el contador que también es una variable que va aumentando de dos en dos y que empieza en el 1, esto se repite por un ciclo que tiene de condición que contador aumentará 2 unidades siempre y cuando sea menor o igual al valor que la persona introduce al principio, hasta que contador alcance un valor que no sobrepase o sea igual al número que se introdujo, cuando lo alcance se detendrá y mostrara el resultado total que se acumuló en AS.

Actividad 2:

Utilizar GBD para corregir el programa

```
1  #include <stdio.h>
2  #include <math.h>
3
4  void main()
5  {
6      int K, AP, N;
7      double X, AS;
8      printf("Ingrese cuántos términos calcular de la serie: X^K/K!");
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;
21     }
22     printf("Resultado=%le",AS);
23 }
```

Compilation

```
main.cpp:4:11: error: '::main' must return 'int'
void main()
    ^
main.cpp: In function 'int main()':
main.cpp:10:14: warning: format '%i' expects argument of type 'int*', but argument 2 has type 'int' [-Wformat=]
scanf("%i",N);
    ^
main.cpp:12:15: warning: format '%lf' expects argument of type 'double*', but argument 2 has type 'double' [-Wformat=]
scanf("%lf",X);
    ^
```

Podemos ver que los errores son que en vez de void main, tendrá que ser int main, y en las líneas 10 y 12 falta el símbolo & para las variables.

El programa corregido es:

```

1  #include <stdio.h>
2  #include <math.h>
3
4  int main()
5  {
6      int K, AP, N;
7      double X, AS;
8      printf("Ingrese cuántos términos calcular de la serie: X^K/K!");
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;
21     }
22     printf("Resultado=%le",AS);
23 }

```



Ingrese cuántos términos calcular de la serie: X^K/K!

N=4

X=5

Resultado=6.537500e+01

...Program finished with exit code 0

Press ENTER to exit console.

Actividad 3:

Utilizar GBD para corregir el programa

El programa no es incorrecto como tal pero al compilar y correr, no tiene sentido el resultado con lo que se propone el programa

```
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 }
20
```

input

Ingrese un número:
6
El factorial de -1 es 0.

...Program finished with exit code 0
Press ENTER to exit console.[]

Tras tratar de corregir el programa para que tuviera sentido, quedo asi:

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

Y estos son algunos resultados

```
Ingrese un número:
5
El factorial de 5 es 120.

...Program finished with exit code 0
Press ENTER to exit console.
```

```
Ingrese un número:
6
El factorial de 6 es 720.

...Program finished with exit code 0
Press ENTER to exit console.
```

Conclusión:

Es importante la depuración de programas para poder comprobar programas y verificar el error pues a veces uno puede no saber cuál es y gracias a esta práctica pudimos aprenderlo pero habrá que practicarlo más