Carátula para entrega de prácticas	
Facultad de Ingeniería	Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Alejandro Esteban Pimentel Alarcón

Asignatura: Fundamentos de Programación

Grupo: 03

No de Práctica(s): 10

Integrante(s): Cureño Arvizu Ameyalli Jocelyn

No. de Equipo de cómputo empleado:

No. de Lista o 0779

Semestre: 2020 - 1

Fecha de entrega: 04 de noviembre de 2019

Observaciones:

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.

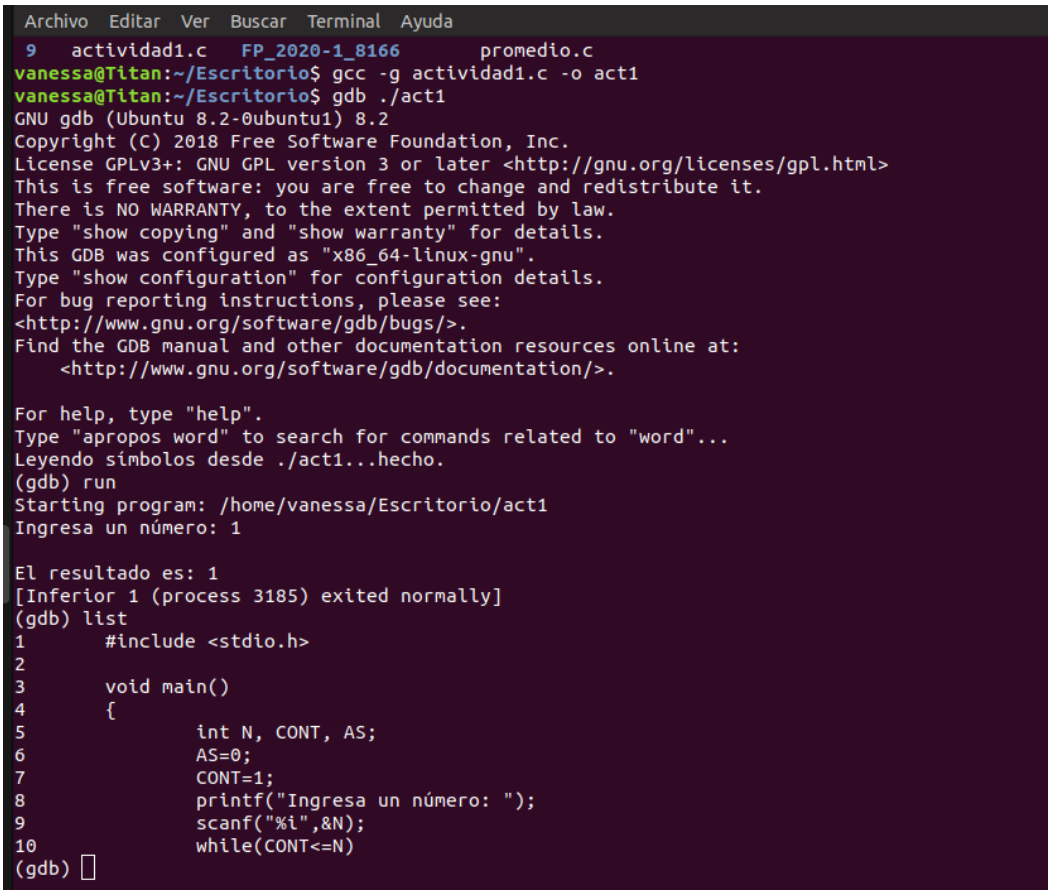
Introducción:

Depurar un programa significa someterlo a un ambiente de ejecución controlado por medio de herramientas dedicadas a ello. Este ambiente permite conocer exactamente el flujo de ejecución del programa, el valor que las variables adquieren, la pila de llamadas a funciones, entre otros aspectos. Es importante poder compilar el programa sin errores antes de depurarlo.

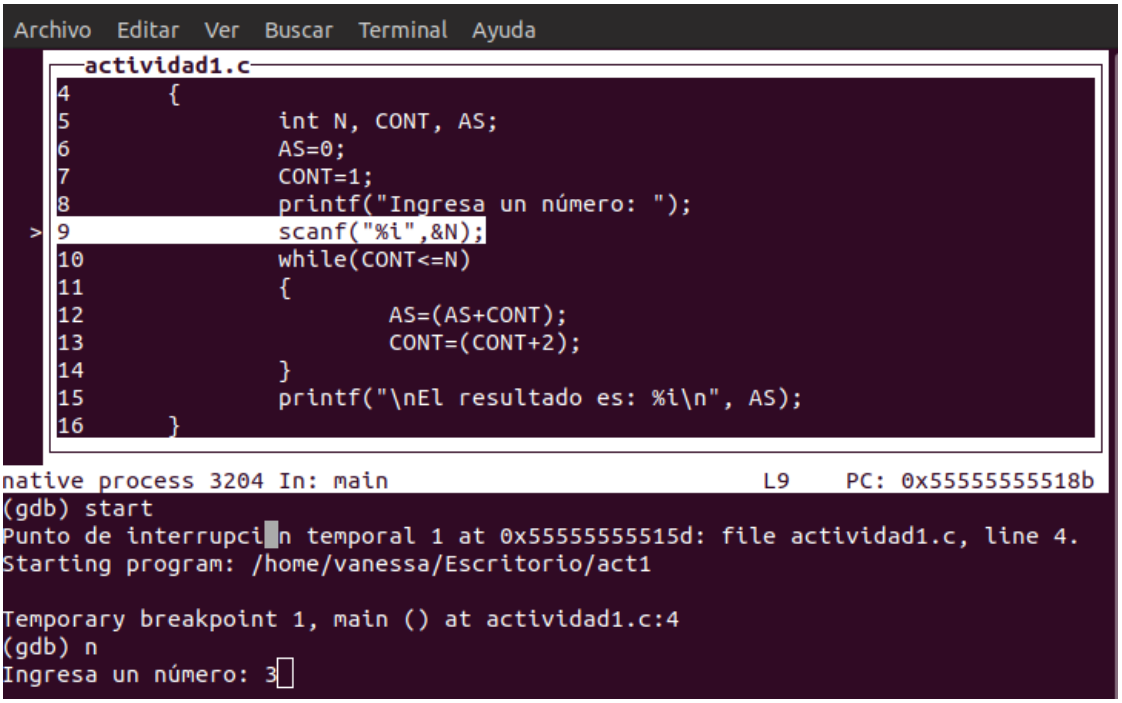
Actividad 1:

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

Para empezar a ver cómo funcionaba, lo compilamos de manera que pudiéramos utilizar gdb, como lo hicimos anteriormente y así ver con detalle su ejecución.



Y después hay que correr el programa igual con GDB para poder analizar con mayor precisión.



Después de que analizamos con gdb, al ir saltando renglón por renglón, se ve que la funcionalidad del programa es recibir un número, después en otra variable llamada *AS* se va guardando el nuevo resultado de la **suma** de *AS* más el *contador* que igual es una variable que aumentará de dos en dos e inicia desde 1 y, ésto se repite por un ciclo que tiene como condición que, *contador* aumentará 2 unidades siempre y cuando sea **menor o igual** al valor que se introduce al inicio, por tanto, hasta que *contador* alcance un valor que no sobrepase o sea igual al numero que se introduce, seguirá haciendo los acumulados de la suma tanto en *AS* como en *contador*. Cuando lo alcance, entonces ahí se detiene el ciclo y muestra el resultado total que se fue acumulando en *AS.*

Actividad 2:

Utilizar GDB para corregir el programa, para compilar el código de la actividad, ejecutar:

$ gcc -w actividad2.c -o actividad2 -lm

En esta actividad vamos a seguir los mismos pasos que en la primera, solo que con diferente compilación ya que con la primera opción, este programa no se puede compilar.

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
1876    in vfscanf.c
vanessa@Titan:~/Escritorio$ gcc -g -w actividad2.c -o a2 -lm
vanessa@Titan:~/Escritorio$ gdb ./a2
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 símbolos desde ./a2...hecho.
(gdb) run
Starting program: /home/vanessa/Escritorio/a2
Ingrese cuántos términos calcular de la serie: X^K/K!
N=4
X=5
Resultado=6.537500e+01[Inferior 1 (process 2427) exited normally]
(gdb) list
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);
(gdb)
```

```
actividad2.c
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      }
24
25
26
native process 2380 In: main
    errp=errp@entry=0x0) at vfscanf.c:1881
(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 0x55555555515d: file actividad2.c, line 8.
Starting program: /home/vanessa/Escritorio/ac2

Temporary breakpoint 2, main () at actividad2.c:8
(gdb) n
(gdb) n
(gdb) n
N=
```

Y cuando se corrige se muestra asi:

numero.c | tablas.c | promedio.c | actividad2.c

```c
#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\n",AS);
}

```

Actividad 3:

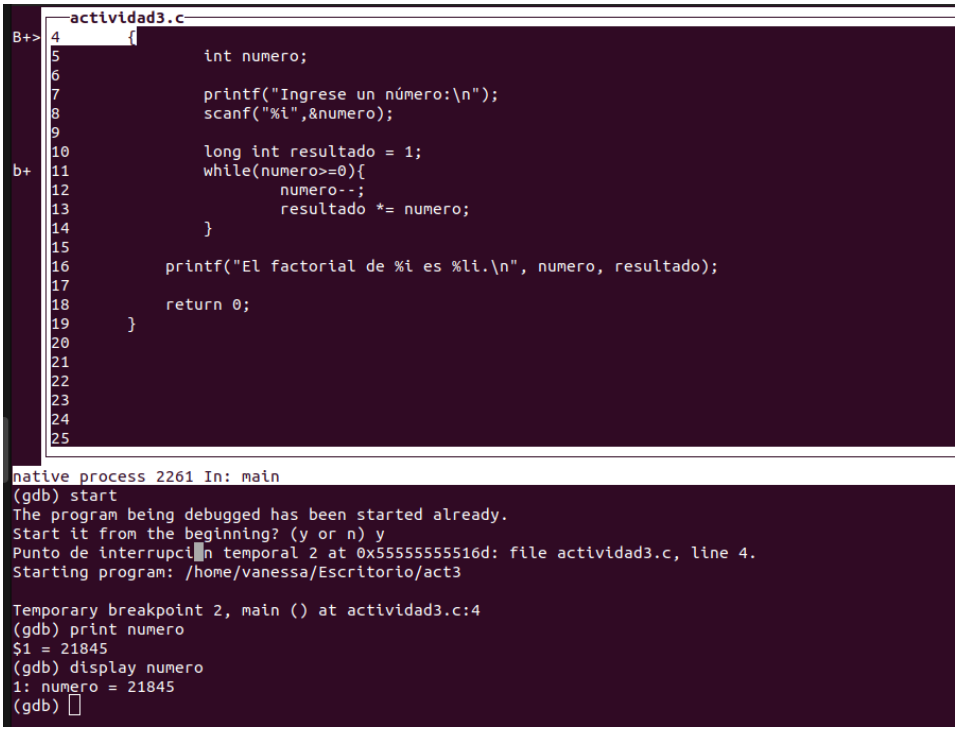
Utilizar GDB para corregir el programa.

Archivo Editar Ver Buscar Terminal Ayuda

```
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;
vanessa@Titan:~/Escritorio$ gdb ./act3
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 símbolos 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)
```



Conclusión:

Es importante la depuración de programas a nivel profesional, un programa se puede probar con casos específicos del usuario, sin embargo, puede que exista un fallo para una situación en particular que no previó el usuario, para esto sirve la depuración del programa que nos permite probar un programa para cualquier caso posible.

También nos permite encontrar el fallo dentro del código a través de varias funciones, nos permite imprimir funciones y mostrar un listado de una parte del código fuente, nos ayuda a probar casos específicos dentro de la terminal y nos permite saltar líneas de código para el funcionamiento correcto de un programa.