Laboratorios de computación salas A y B

Profesor:	Marco Antonio Martínez Quintana
Asignatura:	Fundamentos de programación
Grupo:	3 sección B
No de Práctica(s):	10
Integrante(s):	Frías Hernández Camille Emille Román
No. de Equipo de cómputo empleado:	No aplica
No. de Lista o Brigada:	15
Semestre:	Primer Semestre
Fecha de entrega:	/11/2020
Observaciones:	

CALIFICACIÓN:

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.

Desarrollo:

Debido a que nuestro entorno de trabajo está en gcc nos centraremos principalmente en su forma de depurar y haremos las actividades en dicho entorno.

Gcc es un compilador basado en Linux, el cual nos permite compilar en una terminal, de manera que podrá generar programas y por ende habrá que depurarlos de vez en cuando, para esto primero generaremos un ejecutable el cual será el archivo del cual el depurador hará su trabajo. Para esto utilizamos

gcc -g -o Nombre Nombre.c

Posterior a ello iniciamos el depurador con

gdb. / Nombre

Esto nos permitirá abrir el depurador e iniciar una línea de comandos como:

List o l: Permite listar diez líneas del código fuente del programa, si se desea visualizar todo el código fuente debe invocarse varias veces este comando para mostrar de diez en diez líneas. Se puede optar por colocar un número separado por un espacio para indicar a partir de qué línea desea mostrarse el programa. También es posible mostrar un rango de líneas introduciendo el comando y de qué línea a qué línea separadas por una coma. Ejemplo: lista 4,6

b: Establece un punto de ruptura para lo cual debe indicarse en qué línea se desea establecer o bien también acepta el nombre de la función donde se desea realizar dicho paso. Ejemplo: b5

d o delete: Elimina un punto de ruptura, indicando cuál es el que debe eliminarse usando el número de línea. Ejemplo: d 5

clear: Elimina todos los puntos de ruptura. Ejemplo: clear

info line: Permite mostrar información relativa a la línea que se indique después del comando. Ejemplo: infoline 8

run o r: Ejecuta el programa en cuestión. Si el programa tiene un punto de ruptura se ejecutará hasta dicho punto, de lo contrario se ejecutará todo el programa.

c: Continúa con la ejecución del programa después de un punto de ruptura.

s: Continúa con la siguiente instrucción después de un punto de ruptura.

n: Salta hasta la siguiente línea de código después de un punto de ruptura.

p o print: Muestra el valor de una variable, para ello debe escribirse el comando y el nombre de la variable separados por un espacio. Ejemplo: p suma_acumulada

ignore: Ignora un determinado punto de ruptura indicándolo con el número de línea de código.Ejemplo: ignore 5

q o quit: Termina la ejecución de GDB.

Actividad:

Para el siguiente código fuente, utilizar algún entorno de depuración para encontrar la utilidad del programa y la funcionalidad de los principales comandos de depuración, como puntos de ruptura, ejecución de siguiente línea o instrucción.

1.-

```
#include <stdio.h>

void main()

{

	Int N, CONT, AS;

	AS=0;

	CONT=1;

	Printf ("TECLEA UN NUMERO:");

	scanf ("%i",&N);

	while (CONT<=N)

	{

		AS=(AS+CONT);

		CONT=(CONT+2);

	}

	Printf ("\nEL RESULTADO ES %i\n",AS);

}
```

Funcionalidad del programa:

El programa guarda un número, y le suma un contador siempre que el número sea mayor que el contador y después le suma 2 al contador.

Función de los comandos básicos:

Los puntos de ruptura sirven para saber los valores de la variable en cada momento del código o en qué momento saber, la ejecución de la siguiente línea que sirve para seguir con la línea de comando siguiente.

2.-

```
#include<stdio.h>
Void main()
{
      int i,j;
      for(i=1;i<10;i++)
      {
            printf("\nTabla del%i\n",i);
            for(j=1;j==10;j++)
            {
                  printf("%i X %i = %i\n",i,j,i*j);
            }
      }
}
```

El programa funciona con 2 ciclos for los cuales hacen las tablas del 1-10 de forma ordenada el problema del programa se resuelve poniendo igual o menor que, de la siguiente forma:

3.-

```
#include<stdio.h>
void main()
{
      int i,j;
      for(i=1;i<=10;i++)
      {
            printf("\nTabla del%i \n",i);
            for(j=1;j<=10;j++)
            {
                  printf("%i X %i = %i\n",i,j,i*j);
            }
      }
}
```

El problema era que a la hora de pedir los valores no los guardaba por lo tanto el programa no se podía ejecutar, esto se resuelve usando el “&” al momento de guardar, quedando de la siguiente manera:

```
#include <stdio.h>

#include <math.h>

void main()

{

	int K,X,AP,N;

	float AS;

	printf("EL TERMINO GENERICO DE LA SERIE ES: X^K/K!");

	printf("\n N=");

	scanf("%d",&N);

	printf("X=");

	scanf("%d",&X);

	K=0;

	AP=1;

	AS=0;

	while(K<=N)

	{

		AS=AS+pow(X,K)/AP;

		K=K+1;AP=AP*K;

	}

	printf("SUM= %ld",AS);

}
```

Conclusiones:

La depuración de programas es fundamental para obtener un resultado rápido, eficiente y sin fallas, es por eso que los depuradores son un conocimiento esencial en nuestra carrera

Bibliografía

Gutiérrez Rodríguez, Javier Jesús. Primeros pasos con GDB. Consulta: octubre de 2016. Disponible en: http://www.lsi.us.es/~javierj/ssoo_ficheros/GuiaGDB.htm

Ferreira, Amelia. Depurador gdb. Consulta: octubre de 2016. Disponible en: http://learnassembler.com/gdbesp.html

Ferreira, Amelia. Depurador gdb -uso de la opción -g de gcc. Consulta: octubre de 2016. Disponible en: http://learnassembler.com/opc.html

Gutiérrez,Erik Marín. Depuración de programas DevC++. Consulta: octubre de 2016. Disponible en: http://programacionymetodos.blogspot.mx/2012/05/depuracion-de-programas-dev-c.html

González Cárdenas, Miguel Eduardo; Marín Lara, Claudia Lorena; Noguerón Pérez, Pedro. Apuntes De Computadoras Y Programación.Universidad Nacional Autónoma de México.

Pozo Coronado,Salvador. Primeros pasos con GDB. Consulta: octubre de 2016. Disponible en: http://www.c.conclase.net/devcpp/?cap=depurar