	
Facultad de Ingeniería	Laboratorio de docencia

Laboratorios de computación

Profesor: Saavedraa Hernández Honorato

Asignatura: Fundamentos de programación

Grupo: 01

No de Práctica(s): 10 "Depuración de programas"

Integrante(s): Luis Salinas Ludwig

Semestre: 2018-1

Fecha de entrega: 06/11/2017

Observaciones:

CALIFICACIÓN: _____

salas A y B

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.

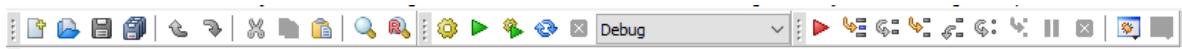
Desarrollo

Comenzamos esta práctica por analizar que es y de que se trata la depuración, a lo cual se vio que la depuración de un programa significa someter nuestro programa a una ejecución controlada por medio de herramientas creadas para tal razón. Lo que nos permite revisar a fondo nuestro programa y detectar si existe algún error por el cual no nos permite compilarlo y por estas razones no correrá adecuadamente.

Para el mejor entendimiento de la práctica se vio adecuadamente lo que era un error, defecto y falla.

Nos concentramos en la depuración de programas escritos en C con Code::Blocks, donde para poder depurar cualquier programa es necesario seleccionar en el asistente Create “Debug” Configuration y Create “Release”. Existen dos carpetas asociadas que son /bin/debug y /bin/release donde se crearán los ejecutables de depuración y su final. Este se encuentra en la parte izquierda a un costado del nombre de nuestro proyecto.

Cuando se lleva a cabo el desarrollo de nuestro programa es importante hacer notar que nos encontremos en el modo de depuración, el cual se cumple en la barra de herramientas.



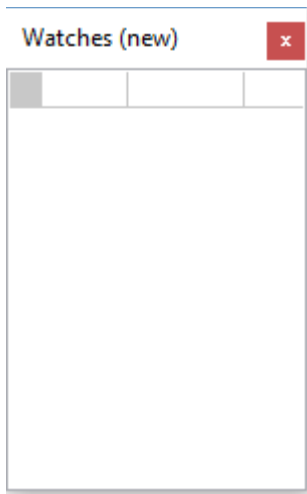
El menú que tiene la opción Debug y Release, siempre debe estar seleccionada la primera opción hasta no haber terminado nuestro programa, posteriormente cambiaremos la segunda opción y de esta manera utilizaremos el ejecutable que se encuentra en la /bin/release.

Del lado izquierdo del menú, observamos las opciones de compilación y ejecución del programa en el modo seleccionado. Del lado derecho observaremos las opciones más importantes para esta práctica que son las opciones de depuración.

Debug/Continue nos permite ejecutar el programa en el modo de depuración y reanudar la ejecución después del punto de ruptura. Stop debugger, está nos permite detener la depuración y continuar la edición de nuestro programa.

Para poder ver el cambio que se va dando en nuestras variables conforme el flujo del programa sigue es posible visualizarlo dándole clic sobre ella con el depurador corriendo, dándole clic en Watch “variable”, donde observaremos un pequeño

cuando con una tabla que contiene las variables, las cuales deseamos visualizar y ver el cambio que existe en su valor siguiendo el flujo.



Para agregar un punto de ruptura break point (punto de ruptura) tenemos que hacer clic derecho sobre algún número de línea de código y agregarlo, donde aparecerá un punto roco, de igual manera para quitar el punto de ruptura es necesario hacer clic derecho y este desaparecerá.

Podremos hacer correr nuestro programa en modo de depuración dándole clic izquierdo en Debug/Continue y podremos ver paso a paso cuando se corre el programa con una flecha amarilla que podremos ir corriendo paso a paso dándole clic en next line.

Por último, analizamos los ejercicios propuestos donde se vieron todos los puntos teóricos aplicados y de esta manera entender lo que es la depuración.

Conclusiones

La depuración en programas desde mi punto de vista es muy importante ya que gracias a esta podremos localizar nuestros errores en los programas, que por culpa de ellos nuestro programa no podrá compilar ni correr. Entender la depuración es una gran herramienta, ya que de esta manera podremos trabajar mejor y ser más eficaces. Así mismo el desarrollo de esta práctica nos ayudó para recordar cómo deben ser el cuerpo de nuestro programa dependiendo de su aplicación, ya que al olvidar la sintaxis es muy común tener errores, pero gracias a la depuración estos pueden ser solucionados.