COMPUTACIÓN Y ALGORITMOS	**Tarea 0**
D. Alonso Ramírez Manzanares	Uso del debugger
Guillermo Arriaga García	25 de enero de 2011

USO DEL DEBUGGER

"Debugger" significa "depurador" y es un programa que sirve para identificar los errores de otro programa, para solucionarlos. Es posible compilar un programa que contenga errores de ejecución ajenos a la sintaxis; por lo que, en la ejecución del programa no llegaremos al resultado deseado debido a una asignación equívoca de algo. Depurar un programa es una herramienta valiosa para poder saber que es lo que está sucediendo en cualquier parte del programa y, así, identificar más fácilmente la asignación equivocada que tengamos, o haber intentado usar una instrucción no disponible en la versión alctual del CPU, o haber intentado tener acceso a memoria protegida o no disponible. El depurador depende de la arquitectura y sistema en el que se ejecute, por lo que sus funcionalidades cambian de un sistema a otro.

Además de poder eliminar errores de un programa, un debugger permite conocer el funcionamiento interno del programa examinado, por lo que se pueden saltar las restricciones comunes, las protecciones de copia y desarrollar la ingeniería inversa.

El depurador ejecuta el programa en análisis hasta algún punto de detención, para que analizar lo que está sucediendo ahí. Los puntos de detención pueden ser:

- Un punto determinado mediante un punto de ruptura (breakpoint).
- Un punto determinado bajo ciertas condiciones mediante un punto de ruptura condicional.
- Un momento determinado cuando se cumplan ciertas condiciones.
- Un momento determinado a petición del usuario.

Durante esa interrupción, el usuario puede:

- Examinar y modificar la memoria y las variables del programa.
- Examinar el contenido de los registros del procesador.
- Examinar la pila de llamadas que han desembocado en la situación actual.
- Cambiar el punto de ejecución, de manera que el programa continúe su ejecución en un punto diferente al punto en el que fue detenido.
- Ejecutar instrucción a instrucción.
- Ejecutar partes determinadas del código, como el interior de una función, o el resto de código antes de salir de una función.

Típicamente, los depuradores también ofrecen funciones más sofisticadas tales como correr un programa paso a paso (un paso o animación del programa), parar el programa (breacking), es decir, pausar el programa para examinar el estado actual en cierto evento o instrucción especificada por medio de un breakpoint, y el seguimiento de valores de algunas variables. Algunos depuradores tienen la capacidad de modificar el estado del programa mientras que está corriendo, en vez de simplemente observarlo. También es posible continuar la ejecución en una posición diferente en el programa bypaseando un estrellamiento o error lógico. Un "estrellamiento" sucede cuando el programa no puede continuar normalmente debido a un error de programación.

Las opciones que se describen a continuación, permiten al programador ejecutar en modo "manual" las partes del código donde se cree que existen errores lógicos que requieren un análisis detallado. Así como también, ejecutar de manera "automática" las partes del código que se presumen libres de errores.

BREAKPOINT	Punto de parada de la ejecución del programa. Son puntos gruesos en el margen de alguna(s) línea(s) del código.
TOGGLE BREAK POINT	Es para insertar un breakpoint o eliminar alguno al seleccionarlo.
REMOVE ALL BREAKPOINTS	Elimina todos los breakpoints agregados al código.
DEBUGGING WINDOWS	Es la ventana de depuración. En ella se puede consultar el valor de cualquier variable accesible desde ese punto de detención.

STEP INTO	Ejecuta el programa paso a paso (línea a línea del programa o de una función/procedimiento) sin necesidad de incluir breakpoints.
STEP OUT	Hace que se salga de una función o procedimiento que se está ejecutando, pasando a la detención en la sentencia inmediata siguiente a la llamada de dicha función o procedimiento.
RUN TO CURSOR	Ejecuta el programa hasta la línea en la que se encuentre posicionado el cursor.
CONTINUE	Continúa con la ejecución del programa hasta el siguiente breakpoint (si existe) o hasta el final del programa.
RESTART	Vuelve a comenzar la ejecución.
PAUSE	Detiene momentáneamente la ejecución.
NEXT LINE	Posibilita la ejecución paso a paso de cada línea. Si la línea contiene una llamada a función, ésta se ejecuta de una sola vez.
NEXT INSTRUCTION	Aparece descrita la siguiente instrucción a realizar y se reanuda la ejecución paso a paso o se ejecuta la siguiente sentencia. No es recomendable el uso de ésta última debido a que está diseñada para instrucciones en lenguaje ensamblador o lenguaje de máquina (programación de bajo nivel).
WATCHES	Se selecciona en "Debugging Windows" y habre una ventana con las variables locales declaradas en el código. Los watches son vistas permanentes a variables señaladas durante la ejecución. Además, se muestra cuándo dichas variables no son accesibles desde tal o cual procedimiento de ejecución.
EDIT WATCHES	El editor de variables observadas, ofrece la posibilidad de añadir/quitar y modificar el formato de representación de los datos contenidos durante la depuración.

FUENTES CONSULTADAS:

"Depurador Code::blocks" [doc][en linea]. Última consulta: 21 de enero de 2011. Disponible en: http://www.ing.unlp.edu.ar/progalg/apuntes/Depurador%20CodeBlocks.doc

Visual Basic 6.0, Aprenda Informática: "Utilización del debugger" [pdf][en línea]. Última consulta: 21 de enero de 2011. Disponible en: http://www.tecnun.es/asignaturas/Informat1/AyudaInf/aprendainf/VisualBasic6/vbasic60.pdf

Wikipedia: "Depurador" [en línea]. Última consulta: 21 de enero de 2011. Disponible en: http://es.wikipedia.org/wiki/Depurador