Departamento de Ciencias Básicas

Turbo Pascal 7, FPS y sus herramientas de análisis de errores.

Ciencias de la Computación

Septiembre 09

TP7, FPS y sus herramientas de análisis de errores

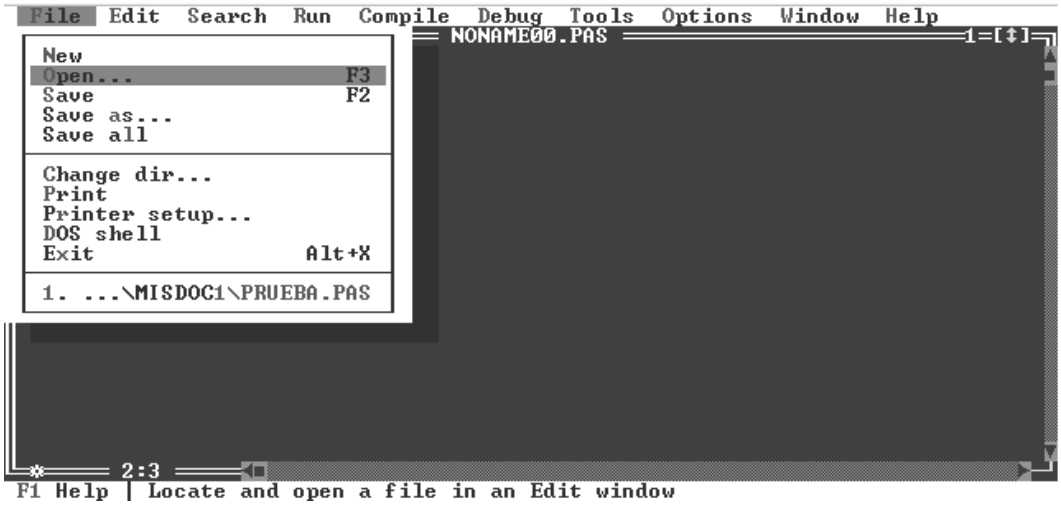
Índice

1. Advertencia preliminar

La siguiente guía de uso para TP7 y FPS es una guía básica para la utilización del software. Dicha Guía no pretende abarcar todo el potencial de estos compiladores, sino que busca transmitir las herramientas básicas para implementar los algoritmos que en la cátedra se plantean y diversas técnicas para evaluar el funcionamiento de los mismos.

2. Uso general del programa

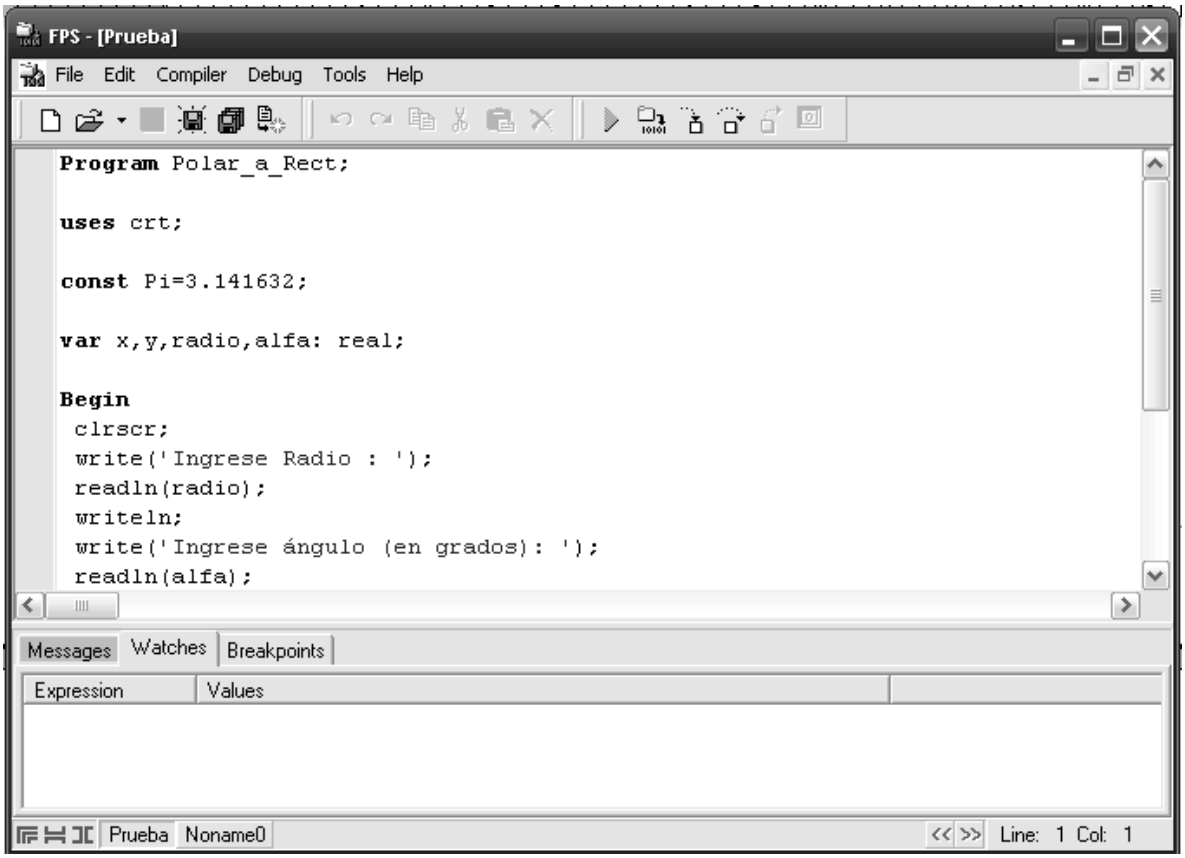
El software **TP7**, a pesar de trabajar bajo la plataforma de DOS tiene una interfaz para el usuario amigable similar a los entornos de Windows.



Existen algunas diferencias con los entornos convencionales de Windows, las funciones copiar, pegar y cortar no son CTRL+C, CTRL+V ni CTRL+X sino que las funciones son las siguientes

Copiar	**CTRL+Insert**
Pegar	**Shift+Insert**
Cortar	**Shift+Supr**
Eliminar selección	**CTRL+Supr**
Eliminar Línea de programación	**CTRL+Y**

El software FPS se presenta más simple que Turbo Pascal 7 y con un entorno de Windows ya que efectivamente trabaja bajo esta plataforma, no obstante reproduce bajo DOS.



Las funciones de copiar, cortar, pegar, etc. son similares a otros entornos de Windows como el paquete de Office o demás software bajo esta misma plataforma.

Existen una serie de palabras llamadas palabras reservadas que no se pueden utilizar como nombres de variables, constantes, nombres de procedimientos, etc. Dichas palabras reservadas se listan a continuación.

and	**asm**	**array**	**begin**	**case**	**const**
constructor	**destructor**	**div**	**do**	**downto**	**else**
end	**exports**	**file**	**for**	**function**	**goto**
if	**implementation**	**in**	**inherited**	**inline**	**interface**
label	**library**	**mod**	**nil**	**not**	**object**
of	**or**	**packed**	**procedure**	**program**	**record**
repeat	**set**	**shl**	**shr**	**string**	**then**
to	**Type**	**unit**	**until**	**uses**	**Var**
while	**wuit**	**xor**			

2.1 Compilación del programa y reproducción del programa (TP7 y FPS)

La compilación del programa es un procedimiento del software de programación que consiste en traducir su propio lenguaje de programación (Pascal, C++, Visual Basic, etc.) en otro lenguaje que interprete la PC para poder reproducir el algoritmo (generalmente lenguaje de máquina).

En TP7 se puede compilar el algoritmo desde la función MAKE que se ubica en (COMPILE → MAKE) o también presionar F9. Dicha función compila el algoritmo y genera un archivo ejecutable (ARCHIVO.EXE) para poder utilizarlo independientemente del software de programación, este archivo se guarda en la carpeta que está guardado el archivo .PAS del programa. La función MAKE solo compila, no ejecuta el programa. Para compilar y ejecutar se usa la función RUN ubicada en el menú RUN o bien presionar CTRL+F9.

De manera similar se puede compilar en FPS desde la función (COMPILER →COMPILE), o bien presionando CTRL+F9, generando también un archivo ejecutable (.EXE). Mientras que para reproducir se debe hacer desde la función (RUN→ RUN), o bien presionamos F9.

Tanto la función MAKE o COMPILE como RUN, aparte de generar el archivo ejecutable, busca errores de sintaxis, si existe alguno el algoritmo no se reproduce.

Recordemos que los software de programación no interpretan errores en el funcionamiento del algoritmo sino en la forma de escribirlo.

2.2 Complementos para mejorar la calidad del algoritmo (TP7 y FPS)

Cuando se ejecuta un programa se observa que en la ejecución en DOS, el programa sigue mostrando el contenido que estaba antes de la ejecución del mismo como se muestra la siguiente imagen.

```
    Microsoft   Windows   XP   [Versión
5.1.2600]
    <C>  Copyright  1985-2001  Microsoft
Corp.

    c:\cd Pascal
    c:\Pascal\Coord

    Ingrese radio : 12

    Ingrese Ángulo (en grados): 30
    ----------------------------
    Resolución
    La coordenada x es : 10.3923
    La coordenada y es : 6.00001

    c:\Pascal\Coord
```

En FPS puede suceder que no pase esto ya que cada vez que se inicia la reproducción del algoritmo FPS limpia automáticamente la pantalla. Sin embargo cuando se ejecute independientemente de FPS, el programa mostrara este error.

Para eliminar el problema se utiliza la función limpiar pantalla o Clear Screen. Se llama al inicio del programa o cada vez que se necesite limpiar la pantalla, escribiendo CLRSCR. Dicha función esta contenida en una librería de funciones de pantalla llamada CRT, entonces se escribe uses crt; en el encabezado del algoritmo.

Cuando mostramos una variable de tipo real en pantalla aparece en forma científica.

1.05678000E+05

{EJEMPLO}

```
Program Polar_a_Rect;

uses crt;

const Pi=3.141632;

var x,y,radio,alfa: real;

Begin
 clrscr;
 write('Ingrese Radio : ');
 readln(radio);
 writeln;
 write('Ingrese ángulo (en grados): ');
 readln(alfa);
 x:= radio*cos(alfa*Pi/180);
 y:= radio*sin(alfa*Pi/180);
 writeln('------------------------');
 writeln('Resolución');
 writeln('La coordenada x es: ',x:6:4);
 writeln('La coordenada y es: ',y:6:4);
 readkey;
End.
```

Para mejorar la visualización lo que se hace es colocar después de la variable la cantidad de cifras enteras que se deseen y luego los decimales

```
writeln (variable : Nº de enteros : N º de decimales);
```

Entonces se observa mejor el número, como se muestra a continuación:

105678.00.

En ocasiones cuando se termina el programa no se visualiza nada de los resultados. El problema es que el software no tiene una pausa en el final que permita dejarnos observar lo que sucede. Para solucionarlo se utiliza una pausa temporal que es la función DELAY, que se la llama de la siguiente manera:

```
Delay(1000);
```

Entre paréntesis va el tiempo en milisegundos que se desea que el algoritmo aguarde. Dicha función se coloca al final del programa o en cualquier lugar donde se desean tener pausas.

También se puede usar la función READKEY, que detiene el algoritmo hasta que el usuario presione una tecla para pasar a la siguiente sentencia. Para esta última función se necesita también, llamar a la librería CRT desde el comando uses, en el encabezado del algoritmo

*NOTA: Cuando se utilice la librería CRT, con el comando **uses** es posible que nos aparezca el siguiente error.*

Dicho error nos imposibilita de utilizar esta librería, debemos eliminar las sentencias de readkey y clrscr. Podemos solucionar el problema reemplazando el archivo TURBO.TPL ubicado en TP\BIN\ por el archivo actualizado en la página de la Materia.

En ocasiones es interesante resaltar con colores leyendas y/o resultados que se muestran para ello una sentencia antes de leyenda, llamamos al procedimiento textcolor y entre paréntesis colocamos el color en inglés. También se puede pude modificar el color de fondo del texto llamando a textbackground y entre paréntesis colocar el color deseado

```
textcolor(color);
textbackground(color);
```

2.2.1 Colores de Fondo y de Texto

Constante	Valor	Significado	De Fondo o de Texto
BLACK	0	Negro	Ambos
BLUE	1	Azul	Ambos
GREEN	2	Verde	Ambos
CYAN	3	Cían	Ambos
RED	4	Rojo	Ambos
MAGENTA	5	Magenta	Ambos
BROWN	6	Marrón	Ambos
LIGHTGRAY	7	Gris Claro	Ambos
DARKGRAY	8	Gris Oscuro	Sólo para texto
LIGHTBLUE	9	Azul Claro	Sólo para texto
LIGHTGREEN	10	Verde Claro	Sólo para texto
LIGHTCYAN	11	Cían Claro	Sólo para texto
LIGHTRED	12	Rojo Claro	Sólo para texto
LIGHTMAGENTA	13	Magenta Claro	Sólo para texto
YELLOW	14	Amarillo	Sólo para texto
WHITE	15	Blanco	Sólo para texto

3. Errores

3.1 Errores comunes de Sintaxis (TP7)

Los errores de sintaxis son aquellos que surgen del tipeo incorrecto de funciones o la omisión de caracteres en el transcurso del programa, el módulo de compilación de ambos softwares los detecta y nos muestra donde esta el código incorrecto.

A continuación se marcan los errores mas comunes.

3	Unknown identifier
26	Type mismatch

58	TO or DOWNTO expected

90	"=" expected
91	":=" expected

36	BEGIN expected
37	END expected
50	DO expected
54	OF expected
57	THEN expected

85	";" expected
86	":" expected
87	"," expected
88	"(" expected
89	")" expected

92	"[" or "(." expected
93	"]" or ".)" expected
94	"." expected
95	".." expected

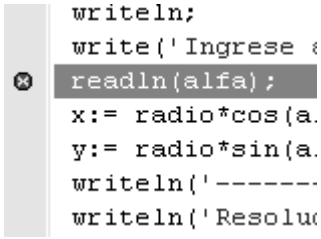
3.2 Detección de errores de Algoritmo (FPS y TP7)

Para la detección de error en los algoritmos se usan dos funciones que permiten observar el comportamiento del mismo, sentencia por sentencia (paso a paso) y otra que nos permite ver como varían las variables. Para hacer la reproducción paso por paso se usa la función STEP OVER o la tecla F8. Por cada vez que se utilice, el algoritmo reproducirá solo la sentencia marcada y pasará a la siguiente sentencia, según corresponda en el algoritmo.

En ocasiones solo se necesita evaluar solo una porción del algoritmo y no se desea reproducir todo paso a paso, para ello se usan los BREACKPOINTS. Los BREACKPOINTS son marcaciones en el algoritmo, cuando el programa llega a esta sentencia se detiene, dando la posibilidad de reproducir el resto del algoritmo de un paso o reproducirlo paso a paso.

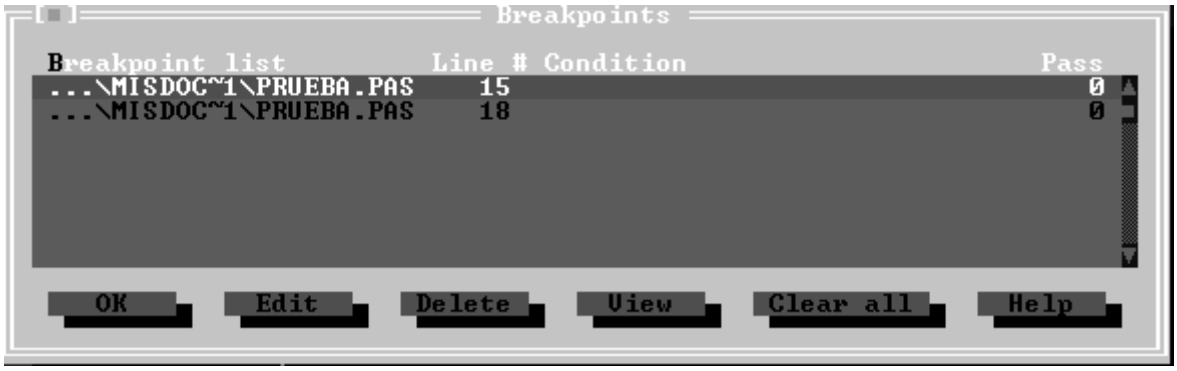
Para insertar BREACKPOINTS, en TP7 se posiciona en la sentencia que se desea detener; ingresando al menú (DEBUG → ADD BREACKPOINTS) o bien CTRL+F8, y aparecerá remarcado con rojo la sentencia de parada. Se pueden colocar más de un BREACKPOINTS en el mismo programa.

En FPS para insertar BREACKPOINTS, se debe posicionar en la sentencia que se desea detener; haciendo un clic derecho a la izquierda de la línea como muestra la figura, también ir al menú (DEBUG→BREACKPOINTS (LINE:##)) o bien presionar CTRL+F8 (ubicado el cursor en la sentencia que se desea detener).

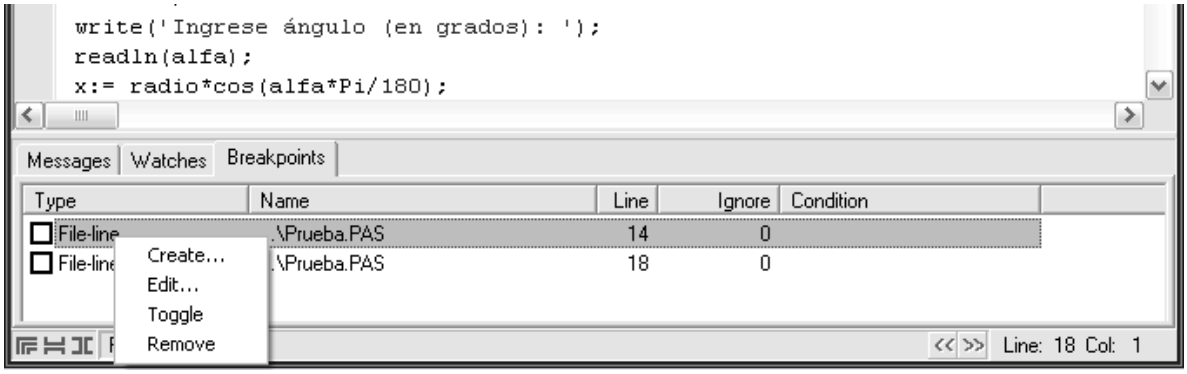


Una vez que se reproduce, el software detiene el algoritmo en la sentencia correspondiente (BREACKPOINT) y a partir de allí da la posibilidad de reproducir paso a paso (presionando F8 ó (RUN→STEP OVER)) ó reproduce completamente las sentencias posteriores (presionando CTRL+F9 ó (RUN→RUN)).

En TP7, si se desea editar o eliminar los BREACKPOINT, vamos a la opción (DEBUG→BREACKPOINTS) y se observa a siguiente ventana donde se puede desplazar los BREACKPOINTS o eliminarlos



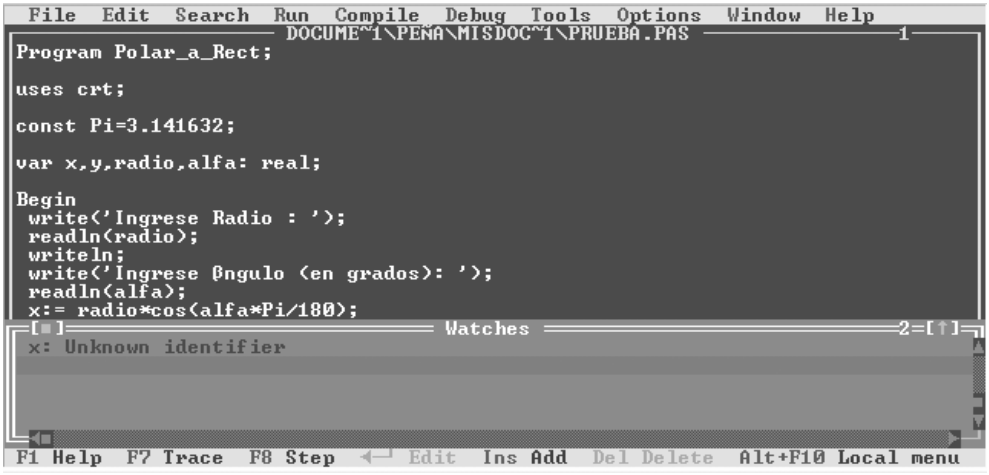
En FPS, si se desea editar los BREACKPOINTS en la parte inferior de la pantalla aparece una ventana con tres pestañas, desde la pestaña de BREACKPOINTS se los puede modificar y/o eliminar.



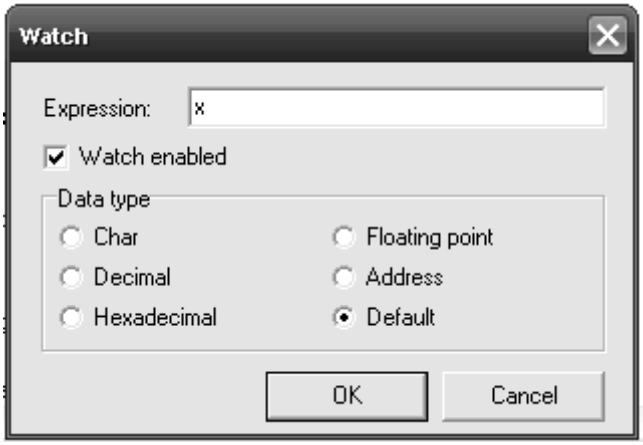
También otra potente herramienta para la detección de errores que se utiliza en forma conjunta con la función paso a paso (STEP OVER ó F8), es la función de WATCHES. Con dicha función una ventana muestra como varían las variables que se deseen observar.

En TP7, para agregar una variable a la ventana de Watches se debe ir a (DEBUG→ ADD WATCH…) y se abre la siguiente ventana.

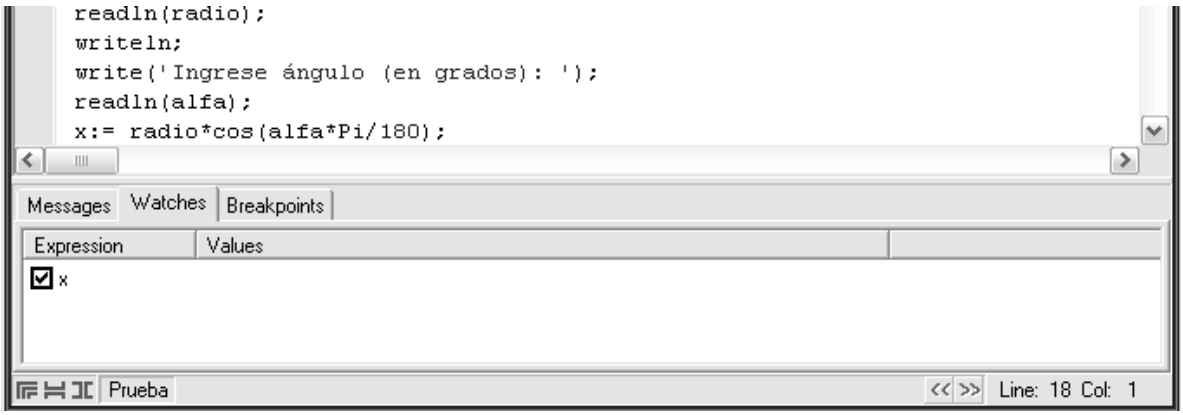
En la ventana anterior, donde dice “Watch expression” se coloca el nombre de la variable que se desea visualizar. Una vez agregada se abre la ventana de WATCHES desde (DEBUG→WATCH), generalmente el software lo hace automáticamente. Allí se observa como varían las variables, se recomienda poner BREACKPOINTS y reproducir paso a paso para observar fehacientemente la evolución de la variable a lo largo del algoritmo.



En FPS, para agregar una variable a la ventana de WATCHES vamos a (DEBUG→ WATCHES→Create). Allí aparece la siguiente ventana



En donde dice “Expression:” se coloca la variable que se desea observar y luego clic en OK. En la solapa de WATCHES ubicada en la parte inferior de la pantalla principal se observa que aparece la variable en cuestión.



Entonces ahora se ejecutará paso a paso el programa o con BREACKPOINTS y se puede observar en esta ventana como varían las variables elegidas.

4. Bibliografía

- Archivo de Ayuda FPS 0.6.4- Rimgaudas Laucius - (04/2002)
- Archivo de Ayuda Turbo Pascal 7.0 - Borland, Internacional - (1983-92)
- TURBO PASCAL 7.0 -

Cualquier duda o sugerencia comunicarse a:
alejandro_soraire@hotmail.com