

Win8086: Una herramienta para el análisis de la frecuencia de utilización de las instrucciones 8086

José L. Poza Luján¹, Álvaro Doménech Pujol², Pedro López Rodríguez³

¹*Escuela Universitaria de Informática. U. P. de València*
e-mail: jopolu@doctor.upv.es

²*Escola Universitària d'Informàtica. U. P. de València*
e-mail: alvaro@disca.upv.es

³*Facultad de Informática. U. P. de Valencia*
e-mail: plopez@disca.upv.es

Resumen: En el presente artículo se presenta una herramienta desarrollada para la obtención y posterior análisis de la frecuencia de utilización de las instrucciones Intel 8086. Esta herramienta permite generar las trazas en sistemas basados en DOS y Windows, y permite analizarlas gráficamente de forma cómoda.

1.- INTRODUCCIÓN.

a) Motivación

En las asignaturas de arquitectura de los computadores, sean genéricas o específicas del PC, es interesante disponer de una herramienta que permita estudiar el uso que los programas hacen del juego de instrucciones de la familia Intel. Si con esta herramienta se pueden monitorizar tanto los programas propios del usuario, escritos en un lenguaje de alto nivel, como los programas ya compilados disponibles, su ámbito de aplicación se extiende hacia las asignaturas de compiladores o de algorítmica.

b) Herramientas actuales

Actualmente existen métodos de monitorización en los que predomina el aislamiento sobre el sistema a analizar [Egg 90].

También existen herramientas software que monitorizan al propio sistema en el que se ejecutan, una de ellas es la herramienta Etch para sistemas Intel basados en Windows32 [Rom98], en la que se aprovechan las facilidades de Windows NT para la monitorización. Las herramientas que funcionan bajo Windows NT tienen la dificultad de no encontrar fácilmente este sistema entre el alumnado.

También se puede encontrar otra herramienta, IDTrace [Pier 94] que no monitoriza, sino simula la actuación del sistema. Estas herramientas tienen el problema de que el alumno/a no interactúa directamente en el sistema, esa interactividad es muy importante en el proceso de enseñanza-aprendizaje.

Otro de los mayores problemas con los que nos encontramos, es el que está en los archivos de trazas generados, en ocasiones el tamaño de los archivos los hace inviables para su análisis, por lo que se debe reducir su tamaño en gran medida [JoHa 92].

b) La herramienta Win8086

La herramienta desarrollada se divide en dos bloques principales, la generación de trazas, y el análisis de las mismas.

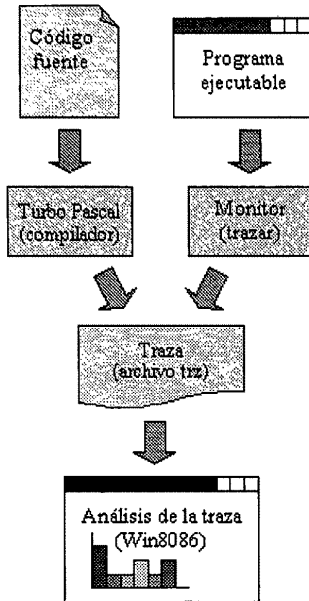


Ilustración 1 Esquema de bloques y operación

Una cuenta de Win8086 es un estadillo en el que se contabilizan el número total de instrucciones procesadas, las frecuencias de los códigos de operación ejecutados, la frecuencia de cada modo de direccionamiento aplicado, la distribución de direcciones accedidas, la tasa de aciertos de las instrucciones de salto condicional y la distribución de distancias cubiertas por estas últimas. Hay dos métodos de obtener una traza. El primero se aplica a casos en que se dispone de un código fuente en alto nivel y consiste en la inserción de llamadas a procedimientos de monitorización en el texto del programa. El segundo es la ejecución de un programa ejecutable activando previamente el sistema de monitorización.

El análisis de las trazas generadas se hace por medio de la aplicación *Win8086* que proporciona vistas de informes o de gráficos. Todo el proceso de trabajo se puede ver en la Ilustración 1.

2.- GENERACIÓN DE TRAZAS.

El esquema empleado para la monitorización se puede apreciar en la siguiente secuencia. En la Ilustración 2 se aprecia la ejecución normal de una instrucción, desde el punto de vista del programador y sin entrar en los detalles de cómo la UCP procesa la misma.

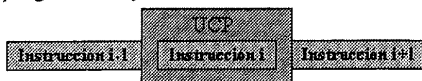


Ilustración 2 Secuencia habitual de ejecución

En la Ilustración 3 se aprecia el método de monitorización empleado: modificando un bit de la palabra de estado del procesador, éste provocará la interrupción de traza en cada ciclo de instrucción. Previamente se habrá instalado en la tabla de vectores la rutina de servicio asociada, que se encarga de actualizar los contadores necesarios.

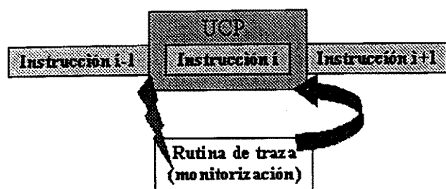


Ilustración 3 Ejecución monitorizada

La rutina de servicio activada accede a la instrucción a través de la pila, como se puede ver en la Ilustración 4.

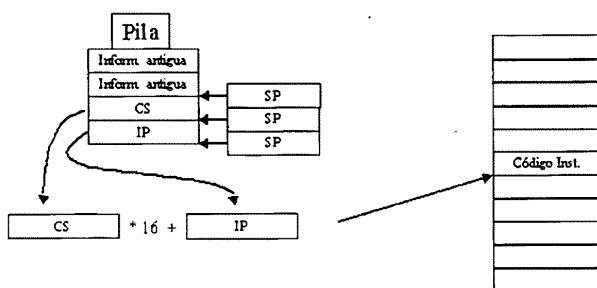


Ilustración 4 Acceso a la información de la instrucción procesada

a) Monitorización de código fuente.

La monitorización del código fuente se realiza por medio de la importación de una unidad de Turbo Pascal: *trazas*.

Esta unidad proporciona los siguientes servicios:

Servicios de tiempo. Para poder obtener los tiempos empleados en la monitorización.

Servicios de traza. Permiten configurar, comenzar y concluir la traza.

Servicios de resultados. Permiten guardar los resultados en un archivo de texto ASCII que será leído por la herramienta de análisis.

La referencia a las funciones de los servicios se hace insertando directamente en el código. Considérese el siguiente ejemplo en que el programa principal llama al procedimiento *whetstone*.

```

Begin
  { Lee número de iteraciones de whetstone }
  write('Número de iteraciones...');
  Readln(iteraciones);
  {Llamada al procedimiento}
  whetstone(iteraciones);
end;
```

Para analizar la ejecución de *whetstone* bastará con insertar referencias a los servicios de traza definidos en *trazas* antes y después de la llamada:

```

uses trazas;
```

```

...
begin
  { Lee número de iteraciones de whetstone }
  write('Número de iteraciones...');
  Readln(iteraciones);
  hola;
  pontraza;
  {Llamada al procedimiento}
  whetstone(iteraciones);
  quitatraya;
  adios;
  guardatraya('whet.trz','Traza de whetstone');
end.

```

El procedimiento *hola*, se encarga de insertar en la tabla de vectores de interrupción las correspondientes funciones de monitorizado

Los procedimientos *pontraza* y *quitatraya*, determinan el comienzo y finalización del monitorizado. Y el procedimiento *adios*, se encarga de restaurar la tabla de vectores de interrupción y dejar el sistema con la misma configuración que antes de la monitorización.

La combinación en la inserción de los procedimientos en el código, puede dar lugar a monitorizar exclusivamente zonas de código determinadas.

La modificación del código se puede realizar directamente en el editor de pascal que corresponda o bien por medio de una ventana de edición de la aplicación Win8086 y llamando al compilador en línea instalado, de una forma transparente al usuario/a.

b) Monitorización de programas ya compilados

Para la monitorización de programas ejecutables, se proporciona un sencillo programa *trazar* cuya labor es activar los procedimientos de monitorización y a continuación lanzar el programa a monitorizar.

Esta llamada se puede realizar desde una sesión en línea de comandos de DOS, la configuración de la llamada se debe hacer por medio de los parámetros del programa, como habitualmente se hace en cualquier aplicación de consola:

```
C:\>trazar /s2 /m100000,20000 c:\command.com
```

En la línea de comandos anterior se monitoriza una sesión de DOS, con rutina de monitorización en ensamblador (s2) y con intervalos de análisis de memoria de la dirección 100000 decimal a la posición 20000 decimal (/m).

También desde una ventana de la aplicación *Win8086* en la que se escoge la aplicación a monitorizar, con los parámetros de monitorización configurables en la correspondiente caja de diálogo.

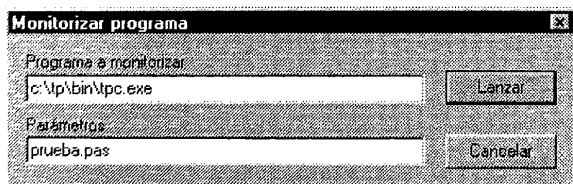


Ilustración 5 Ventana de monitorización de ejecutables.

3. ANÁLISIS

Una vez generada la traza, por cualquiera de los dos caminos posibles, se crea un archivo en el que está toda la información obtenida de la ejecución.

Cabe destacar que el programa puede abrir varias trazas diferentes al mismo tiempo, y varios gráficos a la vez, tanto de una misma traza como de varias.

Esto último es muy útil desde el punto de vista pedagógico, ya que facilita la comparación de resultados.

El archivo es de texto (ASCII), con la información dividida en líneas, para facilitar la forma de acceso al mismo y la interpretación por otro tipo de programas que no sea el propuesto en este proyecto.

Debido al preproceso de la monitorización, el archivo no ocupa más de un Kbyte, por lo que no plantea los problemas clásicos de los archivos de trazas [JoHa 92].

Ese archivo es abierto y procesado por la aplicación Win8086. La información que del mismo se obtiene es la siguiente:

- Instrucciones procesadas.
- Bifurcaciones realizadas.
- Direcciones de memoria accedidas.
- Modos de direccionamiento empleados.

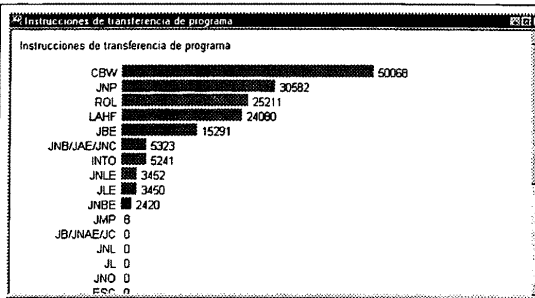
A continuación se expone un ejemplo de cada uno de los análisis.

a) Instrucciones procesadas

Las instrucciones procesadas, se pueden visualizar en modo gráfico de barras o listado de instrucciones. Se permite ordenar o filtrar las mismas por medio de la misma clasificación que Intel realiza [Intel 97]:

- Transferencia de datos.
- Operaciones aritméticas.
- Manejo de bits.
- Manejo de cadenas.
- Transferencia de programa.
- Control del procesador.
- Coma flotante.
- MMX.

Gráficos de instrucciones



Muestra las instrucciones que se han procesado, con un gráfico de barras

Nombre Instrucción	Ocurrencias	% Tipo	% Total
JZ	36398	24.4208	5.1143
JNZ	15689	13.2143	2.7664
LOOP	15136	10.8303	2.2673
RET	14222	9.5456	1.9593
CALL	13219	8.8724	1.8574
JMP	11617	7.7972	1.6323
JBE	8168	5.4822	1.1477
JNL	7983	5.2691	1.1217
JLE	7034	4.7211	0.9894
JNLE	5750	3.8680	0.8093
JBU/AE/JC	4654	3.1237	0.6539
JBU/AE/JNC	2004	1.3451	0.2816
JCXZ/JECXZ	1004	0.6739	0.1411
JNS	1000	0.6712	0.1405
INT	103	0.0091	0.0145
JND	0	0.0000	0.0000
JNP	0	0.0000	0.0000
JNBE	0	0.0000	0.0000
ENTER	0	0.0000	0.0000
JO	0	0.0000	0.0000

Muestra un listado de instrucciones que se muestran con los tantos por cien comparativos de las ocurrencias de las instrucciones.

b) Bifurcaciones

Las bifurcaciones se pueden analizar por medio de un gráfico de barras en los que se pueden ver cuántas instrucciones y tanto por cien han ocurrido, y cuántas de cada sentido (adelante o atrás), han resultado exitosas y cuántas fallidas.

La precisión del gráfico puede variarse como se ve a continuación:

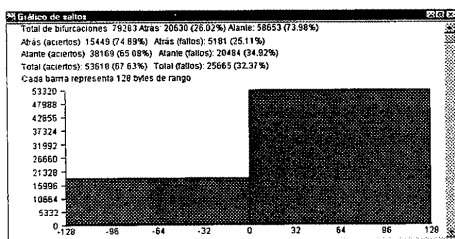


Ilustración 6 Rango de representación = 2

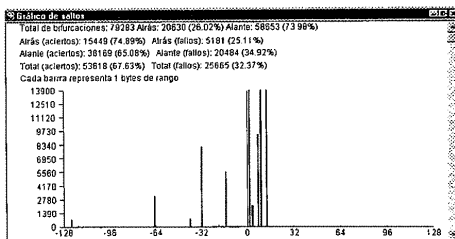


Ilustración 7 Rango de representación = 256

2

256

c) Rango de direcciones de memoria

Las direcciones de memoria accedidas por la aplicación, se pueden monitorizar por rangos y especificando la precisión del mismo, en la visualización se puede ver el número de accesos en cada rango.

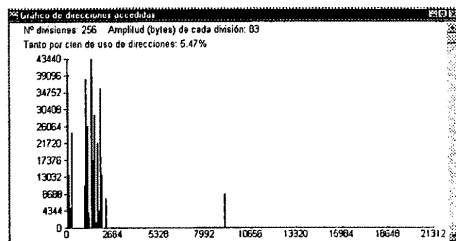


Ilustración 8 Rango de representación = 512

d) Modos de direccionamiento empleados

Se pueden visualizar los modos de direccionamiento empleados, en un gráfico de barras en los que se muestran los mismos, clasificados bajo el criterio de Intel.

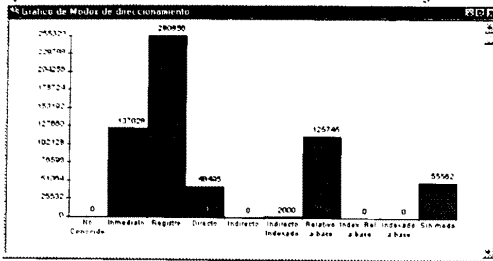


Ilustración 9 Gráfico de modos de direccionamiento.

4.- CONCLUSIONES.

Con esta aplicación, se ha facilitado el estudio de la arquitectura de un sistema, del compilador empleado y el comportamiento de los mismos ante un programa. Aparte, se tiene desarrollada una herramienta fácil de usar y que muestra los resultados de forma agradable.

Con ella se puede analizar fácilmente, a coste casi nulo, el comportamiento del juego de instrucciones de la familia de procesadores Intel 80x86.

Este juego crece para cada nueva versión. Sin embargo, desde el primitivo 8086, el empleo del juego de instrucciones es escaso. Es decir, el 90% de las instrucciones empleadas, suponen tan solo, el 10% de instrucciones disponibles.

Así mismo, por medio de la aplicación se justifica que el estudio de mejoras en los saltos, el empleo de la memoria caché, y la optimización en los modos de direccionamiento suponen un buen avance en el rendimiento del sistema.

Por medio de la herramienta, el alumno/a puede crear por sí mismo/a la monitorización, y llevar a cabo el análisis, con las ventajas que ello implica no sólo en el aprendizaje del comportamiento del procesador, sino en la implicación del alumnado en la práctica, al disponer de una herramienta fácil de emplear.

5.- REFERENCIAS.

- [Rom 98] **Ted Romer y otros**. "Instrumentation and optimization of Win32/Intel Executables Using Etch" University of Washington 1998.
- [Pier 94] **Jim Pierce and Trevor Mudge**, "IDtrace—A Tracing Tool for i486 Simulation". Analysis and Simulation of Computer and Telecommunications Systems (MASCOTS), Enero 1994
- [JoHa 92] **Eric E. Johnson y Jiheng Ha**, "PDATS. Lossless Address Trace Compression For Reducing File Size And Access Time".
- [Egg 90] **S. Eggers, D. Keppel, E. Koldinger y H. Levy**. "Techniques for Efficient Inline Tracing on a Shared-Memory Multiprocessor" Proc. 1990 ACM Sigmetrics Conference on Measurement and Modeling of Computer Systems, Boulder, CO May 1990, pp. 37-47
- [Intel 97] "Intel architecture software developer's manual volume #2 (instruction set reference)" Intel Corporation, 1997 order #243191-001
- [Dom 98] **Domenech Pujol, Álvaro, Real Sáez, Jorge Vicente**. "Organización del PC: del hardware a la aplicación". SPUPV 1998.
- [Poza 99] **José Luis Poza, Álvaro Doménech, Pedro López**. Obtención de trazas. Estudio, y desarrollo de una herramienta de análisis. P .F .C. DISCA – UPV, 1999.