

Módulo 1: Diseño de un Benchmark para evaluación del rendimiento de un sistema computador

PRÁCTICA 1. Diseño de un Benchmark sintético

TEORÍA:

TEMA 1: ANÁLISIS DE PRESTACIONES EN LOS SISTEMAS COMPUTADORES

El usuario de computadores es el candidato perfecto para evaluar un nuevo computador comparando el tiempo de ejecución de su carga de trabajo con el computador antiguo. Sin embargo no siempre esto es posible, siendo necesario utilizar programas evaluadores. A continuación se comentan cuatro niveles de programas de evaluación:

- *Programas reales. Programas comerciales que se pueden utilizar con entradas y salidas reales.*
- *Núcleos. Partes centrales de programas reales. Livermore y Linpack son ejemplos.*
- *Benchmarks reducidos. Programas escritos por usuarios de unas 10-100 líneas, con una función predefinida. Quicksort y Puzzle son ejemplos.*
- *Benchmarks sintéticos. No realizan ninguna función conocida sino son solo un conjunto de operaciones. Se diferencia de los núcleos porque el código se crea artificialmente. Whetstone y Dhrystone son ejemplos.*

Con los tres últimos niveles existe el problema de que el constructor de computadores optimice específicamente su arquitectura para esos programas. En el caso extremo se podría construir un compilador que detectase los benchmarks y generase código optimizado para estos casos particulares.

SPEC (System Performance Evaluation Cooperative 1988) supuso un esfuerzo para armonizar la forma de medir rendimientos. Es posible gracias a los sistemas operativos portables (UNIX) y lenguajes de alto nivel. Consiste en ejecutar un conjunto de programas reales predefinidos con entradas reales. Las casas fundadoras fueron HP, DEC, MIPS y SUN. La versión 1.0 contenía programas con código entero (Gcc, Expreso, Li, etc) y flotante (Spice, Doduc, etc).

1.1 Realización de la Práctica

Realizar un Benchmark Sintético que permita comparar el rendimiento entre dos computadores y calcular el número de MIPS (Millones de Instrucciones Por Segundo) en ambos casos.

Para el diseño del Benchmark se hará uso de la interrupción 1Ch que genera el temporizador del sistema con una frecuencia de 18,2 veces por segundo. Por otra parte, se desarrollará un programa en código ensamblador formado por un número determinado de

instrucciones. Mediante una rutina que atienda a la interrupción 1Ch capaz de controlar el tiempo que transcurre, es posible saber el número de instrucciones que son ejecutadas en ese intervalo de tiempo.



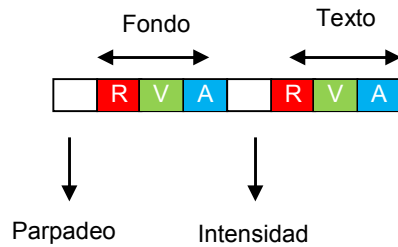
Para la realización de la práctica cada grupo de trabajo va a seguir los siguientes pasos:

1. Elegir una forma de escribir sobre la pantalla: *modo texto* o *modo gráfico*.
2. En modo texto, elegir la forma de mandar los caracteres a pantalla: escribiendo sobre la memoria de video (a partir de la dirección B8000:0000h).
3. En modo texto, elegir la cadena a imprimir en pantalla: cadena numérica, cadena no numérica, cadena alfanumérica y sus atributos si procede.
4. En modo gráfico, elegir los píxeles a colorear en pantalla y asignarle el color.

Información útil

- Direcciones de memoria principal:
 - o Memoria de video en modo gráfico: A000:0000h.
 - o Memoria de video en modo texto: B800:0000h.
- Organización de la pantalla en modo texto: 25 filas x 80 columnas.
- Organización de la pantalla en modo gráfico: 200 filas x 320 columnas.
- Interrupción software para establecer el modo de pantalla:
 - o AL = modo (texto = 03h, gráfico = 13h).
 - o AH = 0
 - o INT 10h

- Asignación de atributos a caracteres en memoria de video en modo texto:



- Colores en modo gráfico:

Color	Código
Negro	00h
Azul	01h
Verde	02h
Rojo	04h
Amarillo	0Eh
Blanco	0Fh

- Interrupción software para capturar una tecla pulsada en teclado:

- o AH = 0
- o INT 16h
- o Se almacena en AX el código de la tecla pulsada

- Proceso de ensamblado:

```
tasm /zi /l nombre_fichero.asm
```

- Proceso de linkado para que un programa sea de tipo .EXE:

```
tlink /v nombre_fichero.obj
```

- Proceso de linkado para que un programa sea de tipo .COM:

```
tlink /t nombre_fichero.obj
```

1.2 Evaluación

Al finalizar la práctica el alumno debe entender y dominar los siguientes conceptos:

- Técnicas para el análisis de rendimiento en sistemas computadores.
- Implementación de un Benchmark Sintético.
- Uso de la interrupción 1Ch para el cálculo de MIPS de un sistema computador.
- Descripción del proceso seguido para calcular los MIPS de un sistema computador.

1.3 Referencias bibliográficas

- Morgan C. L., Waite M. (1988). Introducción al microprocesador 8086/8088 (16 Bit), Madrid, McGraw-Hill.
- Charle Ojeda F. (2003). Programación en ensamblador, Madrid, Anaya.
- Departamento de Arquitectura y Tecnología de Computadores de la Universidad de Granada. El universo digital del IBM PC, AT y PS/2 <http://atc.ugr.es/docencia/udigital/>
- Listado de las principales interrupciones del μ P 8086:
<http://www.itescam.edu.mx/principal/sylabus/fpdb/recursos/r34562.PDF>