Práctica 2 – Eficiencia de la Memoria Caché

1. Representar y analizar la tasa de fallos obtenida para dos tipos de funciones de correspondencia elegidas de entre correspondencia Directa, Asociativa por Conjuntos (definir el número de conjuntos) y Totalmente Asociativa.

Para ello, cada grupo de trabajo va a tener en cuenta lo siguiente:

- Considerar un sistema con un único procesador.

- Realizar las simulaciones con dos tamaños diferentes de memoria caché.

- Elegir para todas las simulaciones una misma política de reemplazo: Aleatoria, LRU, FIFO o LFU, para el caso de correspondencia asociativa.

- Realizar las simulaciones eligiendo dos trazas diferentes. Comentar la utilidad de las trazas elegidas.

Una vez hechas las simulaciones, recoger en una tabla los resultados obtenidos y representarlos en una gráfica. En esta gráfica se recogerá la tasa de fallos frente al tamaño de memoria caché para todas las funciones de correspondencia elegidas

Elegiremos en nuestro caso, comparar entre la correspondencia directa y la totalmente asociativa. Usaremos para ello, las trazas NASA7 y COMP, y el algoritmo LRU.

Tamaño de memoria caché de 32Kb:

* Correspondencia directa:

Configuraremos la memoria caché de la siguiente forma:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

1. Traza COMP:

Tabla

Descripción generada automáticamente

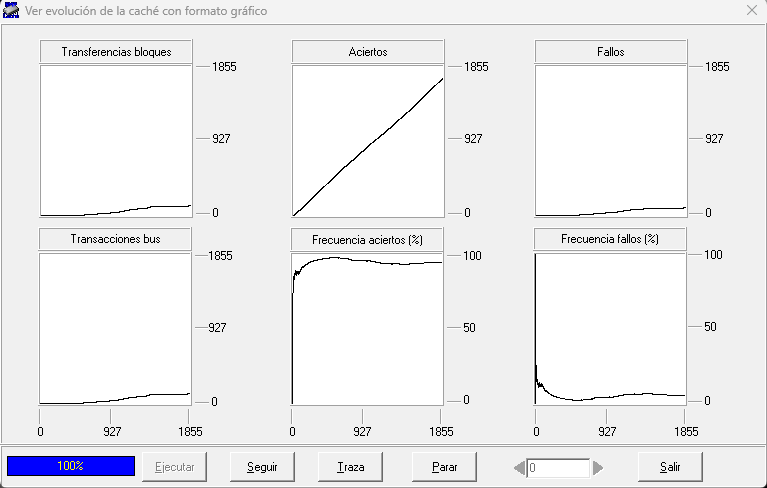
Imagen que contiene Gráfico

Descripción generada automáticamente

1. Traza NASA7:

Tabla

Descripción generada automáticamente



* Correspondencia totalmente asociativa:

Configuraremos la memoria caché de la siguiente forma:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

1. Traza COMP:

Tabla

Descripción generada automáticamente

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

1. Traza NASA7:

Tabla

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente

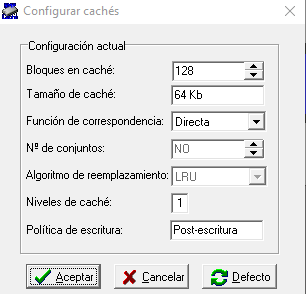
* Resultado final:

|  |  |  |
| --- | --- | --- |
|  | **COMP** | **NASA7** |
| **Directa** |  |  |
| **Totalmente Asociativa** | Medalla de oro premio de ubicación, medalla, cinta, medalla, naranja png |  PNGWingMedalla de oro premio de ubicación, medalla, cinta, medalla, naranja png |  PNGWing |  |

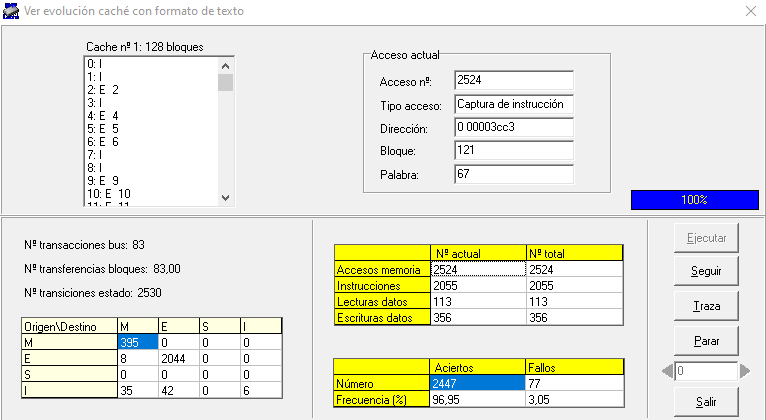
Tamaño de memoria caché de 64Kb:

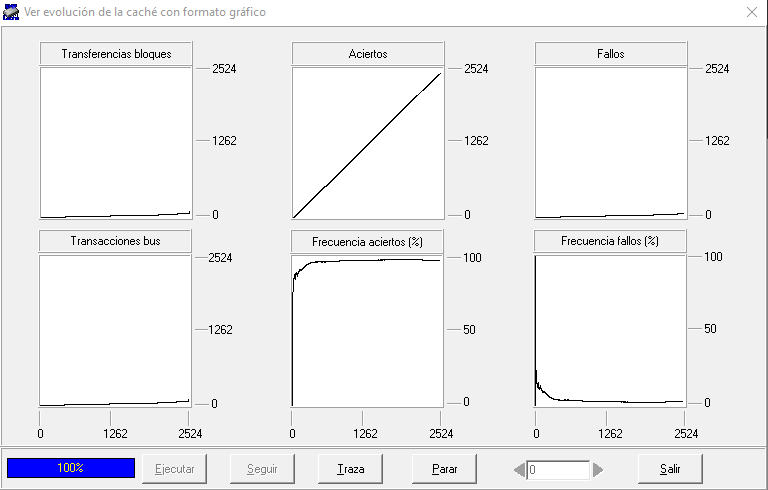
* Correspondencia directa:

Configuraremos la memoria caché de la siguiente forma:

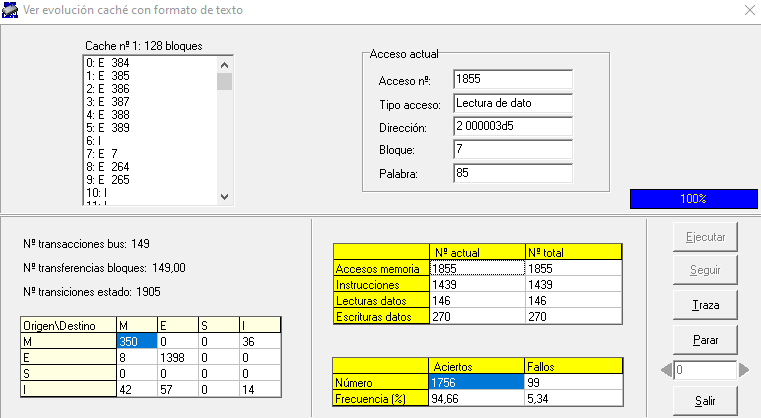


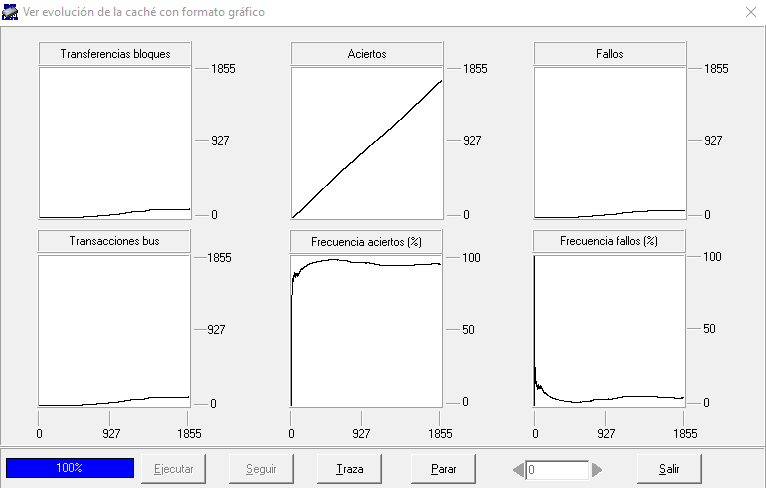
1. Traza COMP:





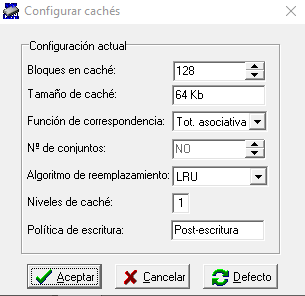
1. Traza NASA7:



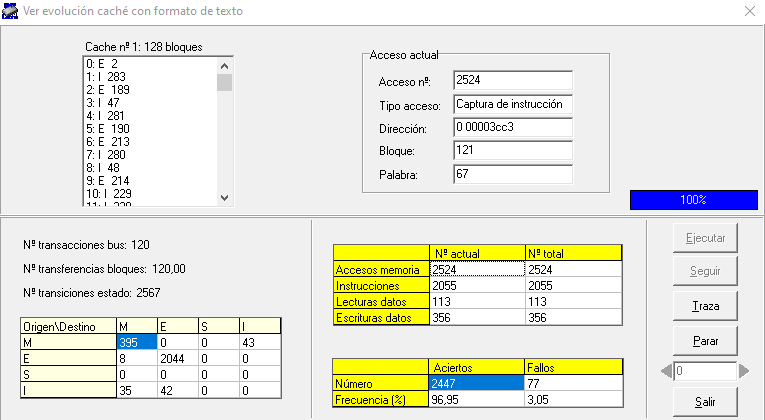


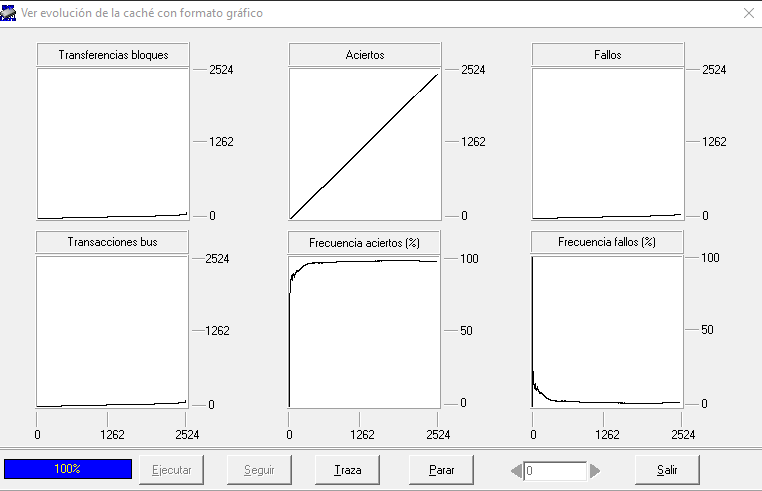
* Correspondencia totalmente asociativa:

Configuraremos la memoria caché de la siguiente forma:

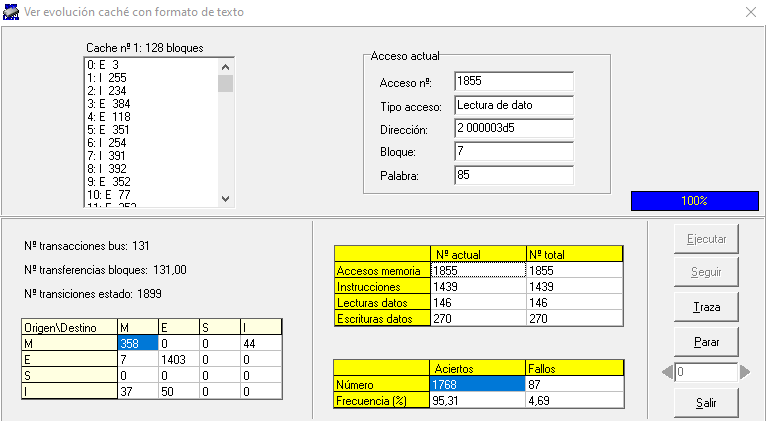


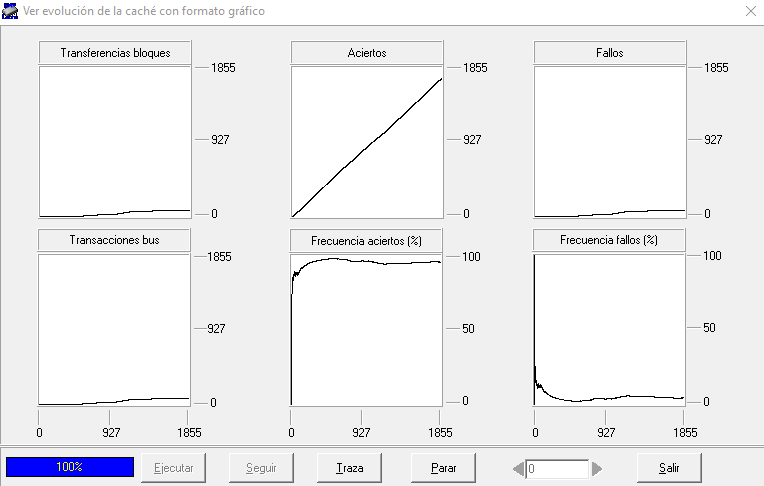
1. Traza COMP:





1. Traza NASA7:





* Resultado final:

|  |  |  |
| --- | --- | --- |
|  | **COMP** | **NASA7** |
| **Directa** | Signo de igual png imágenes | PNGWing |  |
| **Totalmente asociativa** |  | Medalla de oro premio de ubicación, medalla, cinta, medalla, naranja png |  PNGWing |

1. Realizar las simulaciones para cada algoritmo de reemplazo (Aleatorio, LRU, FIFO ó LFU) y responder a las siguientes cuestiones:

- ¿Qué algoritmo de reemplazo es el que obtiene mejores resultados?

- ¿Y el segundo mejor?

- Empleando la opción de visión de la evolución de la caché en modo gráfico, ¿se mantiene constante la tasa de fallos?, ¿por qué?

Para este apartado de la práctica, utilizaremos la configuración:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente



Y la traza NASA7.

1. **Aleatorio:**

**Tabla

Descripción generada automáticamente**

**Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente**

1. **LRU:**

**Tabla

Descripción generada automáticamente**

**Imagen que contiene Gráfico

Descripción generada automáticamente**

1. **FIFO:**

**Tabla

Descripción generada automáticamente**

**Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente**

1. **LFU:**

**Tabla

Descripción generada automáticamente**

**Imagen que contiene Gráfico

Descripción generada automáticamente**

* **Comparación:**

|  |  |
| --- | --- |
| Aleatorio | LRU |
|  |  |
| FIFO | LFU |
|  |  |

|  |  |
| --- | --- |
| Aleatorio | LRU |
|  |  |
| FIFO | LFU |
|  |  |

**Tras una alta tasa de fallos por  
cargar las primeras posiciones en  
memoria, esta se mantiene  
constante.**

1. En esta última parte de la práctica se van a diseñar tres ficheros de traza de memoria de forma que se pueda demostrar que dependiendo del programa que se esté ejecutando, una estructura de memoria caché puede dar un buen rendimiento con una configuración determinada y un mal rendimiento con otra; aunque de manera general y como media se esperase justo lo contrario.

Para ello se va a establecer lo siguiente:

- Número de procesadores: 1

- Protocolo de coherencia: MESI

- Arbitración del bus: Aleatoria.

Y cada grupo de trabajo va a definir lo siguiente:

- Tamaño de palabras.

- Tamaño de bloque.

- Tamaño de memoria.

- Tamaño de cache.

Diseñar tres ficheros de trazas (fichero1.prg, fichero2.prg y fichero3.prg), con 500 líneas cada uno de ellos como mínimo, que demuestren que:

1. fichero1.prg → Da mejor rendimiento para una función de correspondencia directa que para una función de correspondencia totalmente asociativa con el algoritmo de reemplazo elegido.

2. fichero2.prg → Da mejor rendimiento para un algoritmo de reemplazo que para otro (elegir los dos algoritmos de reemplazo), con una función de correspondencia asociativa por conjuntos.

3. fichero3.prg → Obtiene el mejor rendimiento justo al contrario que el del fichero2.prg.

Mostrar y explicar los patrones seguidos por cada uno de los ficheros de trazas diseñados para demostrar lo que se pide en el enunciado.

Mostrar y explicar los resultados que se obtienen al someterlos al proceso de simulación con.

NOTA: Los ficheros con trazas de memoria tendrán extensión .prg, y van a ser ficheros de texto. Constan de un conjunto de líneas, cada una de las cuales contiene dos números, separados por un único espacio en blanco: etiqueta valor

Donde:

etiqueta es un número decimal que identifica el tipo de operación de acceso a memoria demandado por la CPU en un momento dado:

- capturar una instrucción (0),

- leer un dato de memoria (2)

- escribir un dato en memoria (3).

valor es un número hexadecimal, que indica la dirección efectiva de la palabra de memoria a la que se va a acceder. Esta dirección será traducida por el simulador para localizar la palabra dentro de la estructura de bloques del sistema de memoria.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteInterfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Fichero1.prg**

**Patrón: B4, B9, B10, B15, B0**

**Correspondencia DIRECTA**

**Instante 1**

**MC**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| B4\* | B4 | B4 | B4 | B4 B0\* |
|  | B9\* | B9 | B9 | B9 |
|  |  | B10\* | B10 | B10 |
|  |  |  | B15\* | B15 |

**5 fallos, 0 aciertos**

**Instante 2**

**MC**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| B0 B4\* | B4 | B4 | B4 | B4 B0\* |
| B9 | B9 | B9 | B9 | B9 |
| B10 | B10 | B10 | B10 | B10 |
| B15 | B15 | B15 | B15 | B15 |

**2 fallos, 3 aciertos**

**Correspondencia TOTALMENTE ASOCIATIVA (FIFO)**

**Instante 1**

**MC**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| B4\* | B4 | B4 | B4 | B4 B0\* |
|  | B9\* | B9 | B9 | B9 |
|  |  | B10\* | B10 | B10 |
|  |  |  | B15\* | B15 |

**5 fallos, 0 aciertos**

**Instante 2**

**MC**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| B0 | B4 | B4 | B15\* | B15 |
| B4\* | B4 | B4 | B4 | B0\* |
| B10 | B9\* | B9 | B9 | B9 |
| B15 | B15 | B10\* | B10 | B10 |

**5 fallos, 0 aciertos**

En la **correspondencia directa** solo nos daría 5 fallos en el primer instante. Para el resto de instantes solo se producirían 2 fallos y 3 aciertos constantemente. Por lo tanto, estimamos que habrá una tasa de **40%** de fallos. Sin embargo, para la **correspondencia totalmente asociativa** con el algoritmo **FIFO** vemos que se producen 5 fallos constantemente, es decir, una tasa de **100%** de fallos.

Vamos a comprobarlo con el simulador **SMPCACHE**:

Tabla

Descripción generada automáticamente**Correspondencia DIRECTA**

**Correspondencia TOTALMENTE ASOCIATIVA (FIFO)**

Tabla

Descripción generada automáticamente

**Fichero2.prg**

**Patrón: B4, B9, B10, B15, B4, B9, B10, B33**

**Correspondencia ASOCIATIVA POR CONJUNTOS (FIFO)**

**MC**



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| B4\* | B4 | B4 | B4 | B4 | B4 |
| B10\* | B10 | B10 | B10 | B10 | B10 |
| B9\* | B33\* | B15\* | B9 | B15\* | B33\* |
| B15\* | B15 | B9\* | B33\* | B33 | B9\* |

**Instante 1 Instante 2 Instante 3**

**5 fallos y 3 aciertos en el 1er instante y 3 fallos y 5 aciertos en el resto**

**Correspondencia ASOCIATIVA POR CONJUNTOS (LRU)**

**MC**



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| B4\* | B4 | B4 | B4 | B4 | B4 |
| B10\* | B10 | B10 | B10 | B10 | B10 |
| B9\* | B9 | B9 | B9 | B9 | B9 |
| B15\* | B33\* | B15\* | B33\* | B15\* | B33\* |

**Instante 1 Instante 2 Instante 3**

**5 fallos y 3 aciertos en el 1er instante y para el resto 2 fallos y 6 aciertos**

Como podemos observar, el algoritmo **LRU** es más eficiente con una tasa de fallos de aproximadamente **25%** (2 fallos y 6 aciertos), mientras que la FIFO cuenta con una tasa de fallos de **38%** (3 fallos y 5 aciertos)

Vamos a comprobarlo con el simulador **SMPCACHE**:

**Correspondencia ASOCIATIVA POR CONJUNTOS (FIFO)**

Tabla

Descripción generada automáticamente

**Correspondencia ASOCIATIVA POR CONJUNTOS (LRU)**

Tabla

Descripción generada automáticamente

**Fichero3.prg**

**Patrón: B4, B9, B10, B15, B15, B10, B9, B4, B33, B15, B10, B9, B4**

**Correspondencia ASOCIATIVA POR CONJUNTOS (FIFO)**

**MC**



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| B4\* | B4 | B4 | B4 | B4 | B4 |
| B10\* | B10 | B10 | B10 | B10 | B10 |
| B9\* | B9 | B33\* | B15\* | B9\* | B9 |
| B15\* | B15 | B9\* | B9 | B33\* | B15\* |

**Instante 1 Instante 2 y 3 Instante 4 y 5**

**5 fallos y 4 aciertos en el 1er instante y 3 fallos y 7 aciertos en el resto cada 2 instantes**

**Correspondencia ASOCIATIVA POR CONJUNTOS (LRU)**

**Patrón: B4, B9, B10, B15, B15, B10, B9, B4 B33, B15, B10, B9, B4**

**MC**



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| B4\* | B4 | B4 | B4 | B4 | B4 |
| B10\* | B10 | B10 | B10 | B10 | B10 |
| B9\* | B9 | B15\* | B9\*\* | B15\* | B33\* |
| B15\* | B15 | B9\*\* | B15\* | B9\*\* | B9\* |

**Instante 1 Instante 2 y 3 Instante 4 y 5**

**6 fallos y 2 aciertos en el 1er instante y para el resto 6 fallos y 4 aciertos cada 2 instantes**

Como podemos observar, el algoritmo **FIFO** es más eficiente con una tasa de fallos de aproximadamente **30%** (3 fallos y 7 aciertos), mientras que la **LRU** cuenta con una tasa de fallos de **60%** (6 fallos y 4 aciertos)

Vamos a comprobarlo con el simulador **SMPCACHE**:

**Correspondencia ASOCIATIVA POR CONJUNTOS (FIFO)**

Tabla

Descripción generada automáticamente

**Correspondencia ASOCIATIVA POR CONJUNTOS (LRU)**

Tabla

Descripción generada automáticamente