

Simuladores de Jerarquía de Memoria en el Contexto de un Proceso de Investigación-Acción

L. Moreno, E.J. González, B. Popescu, J. Torres, J. Toledo, C. González

Dpto. de Ingeniería de Sistemas y Automática y Arquitectura y Tecnología de Computadores
Universidad de La Laguna

Facultad de Física y Matemáticas, 38207 Tenerife
ejgonzal@ull.es

Resumen

Presentamos en esta ponencia dos simuladores para la docencia de la jerarquía de memoria en varias asignaturas, productos de un proceso de investigación-acción en el cual también se ha originado un mapa conceptual para una mejor coordinación a los profesores de las asignaturas involucradas. En el diseño de ambos simuladores se ha perseguido abarcar todos los conceptos de estudio sin perder la facilidad de uso ni el objetivo pedagógico.

1. Motivación

Los profesores de las asignaturas de Estructura de Computadores, Arquitectura de Computadores y Sistemas Operativos de la titulación de Ingeniero en Informática hemos iniciado un proceso de diagnóstico de nuestras asignaturas que pretende formular estrategias para resolver los problemas en la impartición de algunos de los conceptos comunes a estas disciplinas. Se trata de establecer una adecuada relación entre la teoría y la práctica educativa, aplicando las aportaciones que provienen de las teorías educativas y del conocimiento práctico de los profesores.

Para llevar a cabo este proceso, hemos elegido inicialmente una materia recurrente en las tres disciplinas mencionadas: la Jerarquía de Memoria, considerada en su doble aspecto (Memoria Virtual y Memoria Cache). En este punto, debemos recordar que precisamente la enseñanza de la cache fue el Tema Estratégico de las JENUI celebradas en Palma de Mallorca en 2001 y objeto de estudios posteriores [2], lo que demuestra la importancia de esta materia en la enseñanza universitaria de la Informática.

Para el análisis de los conceptos involucrados en la Jerarquía de Memoria, hemos realizado un proceso de investigación-acción, es decir, un proceso secuenciado que parte de la búsqueda y

diagnóstico de problemas, continúa con la formulación de estrategias para solucionarlos, los pone en práctica, y los evalúa. Con todo ello se plantea una nueva formulación y descripción del problema. En la primera fase del proceso, nos hemos reunidos todos los profesores involucrados, y hemos realizado el mapa conceptual de todos los conceptos que deberán ser impartidos en las tres asignaturas mencionadas

Los mapas conceptuales, creados por Novak en los años 60, son unas redes de conceptos, donde los nodos representan los conceptos y los enlaces las relaciones entre dichos conceptos. Es una herramienta que se constituye en un instrumento eficaz para ayudar a los alumnos a reflexionar sobre la estructura y el proceso de producción del conocimiento. Los mapas conceptuales representan una estructura que va de los conceptos más generales a los más específicos, facilitando la ordenación y secuenciación jerárquica de los contenidos de enseñanza, de forma que ofrecen estímulos adecuados al alumno [6]. Recordemos que en este sentido la motivación en Arquitectura de Computadores ya ha sido objeto de estudio en diversos trabajos como en [2].

En el tema de la Jerarquía de memoria, el uso del mapa conceptual (consultable en la dirección <http://www.isaatc.ull.es/portal/proyectos/mneme/jerarquia.jpg>) nos permitirá explorar estrategias de mejora en la enseñanza de la misma. En este proceso hemos analizado cuáles son los conceptos a abordar, llegando a las siguientes conclusiones:

- Surgen una gran cantidad de conceptos que definen a su vez una gran cantidad de parámetros a dimensionar: políticas locales/globales, procesos, colas de espera/ejecución, algoritmos de reemplazamiento, tamaño de los buses entre los diferentes niveles de cache, tamaño de las caches, tamaño de la página en memoria

virtual y de las líneas en las caches, write allocate/non write allocate, tipos de tablas de página, tamaño de la TLB, etc.

- Debemos contemplar todas las características que nos lleven a situaciones de máquinas existentes: niveles de cache, implementaciones de la tabla de página, algoritmos de reemplazamiento, etc.

El segundo paso en el proceso de investigación-acción ha sido la creación o utilización de simuladores que faciliten la comprensión de los conceptos. El uso de simuladores como recurso didáctico se encuentra muy extendido, ya que ofrece entre otras las siguientes ventajas [5]:

- El alto número de alumnos matriculados en determinadas asignaturas conlleva un aumento de carga docente en prácticas. Los simuladores contribuyen a atenuar este efecto sin que la calidad de la docencia disminuya.
- El desarrollo de plataformas educativas *on-line* puede facilitar su uso y su control mediante un programa de prácticas y tutoriales adecuado. En este caso emplearemos además el mencionado mapa conceptual, el cual los alumnos han colaborado a complementar asociando a cada concepto la teoría correspondiente y los recursos relacionados.

No vamos a entrar en la discusión sobre si deben emplearse simuladores sencillos o simuladores complejos [2]. Diremos que un simulador no puede hacer una simplificación muy grande de la realidad con objeto de hacerlo sencillo. Si la realidad es compleja con multitud de parámetros y con necesidad de mostrar el funcionamiento de una gran variedad de conceptos, entonces el simulador resultará complejo y habrá que dotarlo de mecanismos para hacerlo útil pedagógicamente.

Proponemos, como conclusión primera de este segundo paso, un simulador realizado en nuestra Universidad, que reúne en la Jerarquía de Memoria tanto la memoria virtual como la cache multinivel. Dicho simulador se denomina SIJEM, concebido como una aplicación de escritorio, realizado en C++. Asimismo y también como conclusión de esta etapa hemos decidido crear un nuevo simulador que permita a los alumnos el

aprendizaje en situaciones más complejas como son las que se presentan en las asignaturas de Sistemas Operativos y Arquitectura de Computadores. Este simulador lo hemos denominado MNEME. Ambos recursos pueden ser descargados de manera gratuita de la página web de nuestro Departamento [4][7].

Como tercer paso del proceso de investigación-acción hemos sometido ambos simuladores a la validación de profesores y alumnos de 5º curso de la titulación de Ingeniero Informático. Fruto de esta evaluación surgió la cuarta fase de la estrategia consistente en una reunión conjunta de profesores y alumnos. Esta reunión dio lugar a un conjunto de ideas para mejorar las prestaciones del simulador MNEME, haciéndolo más accesible a los alumnos e incorporando los sistemas multiprocesador y multicore. Ello ha dado lugar a la definición de un nuevo simulador paramétrico y reconfigurable en función de los conocimientos de los alumnos, con lo que hemos llegado a una nueva formulación del problema investigado en el proceso de investigación-acción.

2. Características del Simulador SIJEM

SIJEM [3],[7] es un simulador de jerarquía de memoria cuyo objetivo es servir de apoyo al aprendizaje de los conceptos relacionados con memoria virtual y memorias cache. A través de una interfaz altamente visual SIJEM trata de ilustrar los conceptos de:

- Memoria Virtual: Traducción de direcciones virtuales a direcciones reales; Paginación; Uso de tabla de páginas y TLBs (Translation Lookaside Buffer); Estrategias de búsqueda, colocación y reemplazamiento (FIFO, LRU, Clock, LFU, NUR, etc.) en memoria principal.
- Niveles de memoria: Memoria secundaria; Memoria principal; Memorias caché multinivel (Nivel 3, Nivel 2 y Nivel 1 Conjunta o Separada en datos e instrucciones); Estrategias de colocación, reemplazamiento (FIFO, LRU, Clock, LFU, NUR, etc.) y coherencia entre los diferentes niveles.

El simulador posee una gran cantidad de parámetros configurables, tales como tamaños de las diferentes memorias, bloques y páginas,

tiempos de acceso, caches habilitadas o deshabilitadas, algoritmos de colocación, reemplazamiento y coherencia, etc., disponiendo de un asistente que guía al alumno en él. Este asistente esta dividido en tres fases:

1. *Configurar*, a través de una serie de pasos, guía al usuario en el proceso de configurar todos los parámetros del simulador.
2. *Cargar fichero*. Aquí el usuario elige el fichero de traza entre los múltiples ejemplos incluidos en el simulador o bien los propios ficheros creados por él mismo.
3. *Simular*. En este punto, el usuario elige el tipo de simulación a realizar.

Además se incluye el botón *Resumen configuración*, que muestra en cualquier momento un resumen de la configuración elegida durante el proceso de configuración.

2.1. Configurar

Para guiar al usuario en el proceso de configurar el simulador, los parámetros fueron agrupados según la siguiente secuencia:

1. Memoria virtual / secundaria.
2. Memoria principal.
3. Tamaño de página.
4. Mecanismo de traducción de direcciones.
5. Parámetros de paginación.
6. Tamaño del bloque de caché.
7. Caché de nivel 3 (Tamaño, organización, número de vías, política de reemplazamiento).
8. Caché de nivel 2 (Tamaño, organización, número de vías, política de reemplazamiento).
9. Caché de nivel 1 (Tamaño, organización, número de vías, política de reemplazamiento).

Otra facilidad del simulador en este proceso es la opción de cargar ficheros de configuración creados a mano por el usuario o incluidos como ejemplo, con todos los parámetros necesarios para la simulación. Los ficheros de configuración son ficheros de texto con extensión *.cfg*. Estos ficheros pueden cargarse al inicio del asistente, de manera que los parámetros de las diferentes pantallas son introducidos automáticamente, aunque el usuario tiene la posibilidad de cambiarlos.

2.2. Cargar Fichero

Para realizar una buena simulación es necesario tener ficheros de traza que se correspondan con programas reales. La solución adoptada fue obtenerlos a través del portal de distribución de trazas del PEL (*Performance Evaluation Laboratory*) de la Bringham Young University en Washington. Dado que se trataba de efectuar simulaciones que mostraran las características de diferentes tipos de programas se optó por incluir dentro del simulador las trazas correspondientes a 6 tests del benchmark SPEC2002.

2.3. Simulación

Una vez realizada la configuración y elegido el fichero de traza es el momento de comenzar la simulación. La simulación se divide en dos partes: Traducción de direcciones y Búsqueda de páginas.

Traducción de direcciones

Dependiendo de la configuración elegida, se pueden mostrar tres interfaces diferentes, uno para cada tipo de mecanismo de traducción: a) Traducción de direcciones por transformación directa; b) Traducción de direcciones por transformación asociativa-directa con TLB y c) Traducción de direcciones por transformación asociativa-directa con TLB dividida en datos e instrucciones

Todos ellos comparten los siguientes elementos: Dirección virtual (tanto en binario como en decimal, dividida en número de página y desplazamiento dentro de la página); Dirección real (tanto en binario como en decimal), Dirección base de la tabla de páginas y Tabla de páginas.

Búsqueda de páginas

Dependiendo de la configuración elegida puede llegar a tener los siguientes componentes: Memoria secundaria, Memoria principal, Caché de nivel 3, Caché de nivel 2, Caché de nivel 1 conjunta o separada en datos e instrucciones y CPU.

La simulación se divide en una serie de pasos que muestran el estado de la memoria en ese instante, destacando las posiciones afectadas, la acción realizada o el suceso ocurrido, y una serie de estadísticas. Una de las características

innovadoras de este simulador es que hemos incluido la posibilidad de ir hacia delante y hacia atrás en el proceso de la simulación. Este método permite que los usuarios, en el caso de que den un paso hacia adelante y no comprendan lo sucedido, puedan retroceder uno o varios pasos para tener una visión global del suceso.

Finalmente se han incluido colores como elemento de ayuda al alumno para facilitar la comprensión de lo que está ocurriendo en cada momento. El color verde indica “éxito” mientras que el color rojo indica “fallo” en la búsqueda de páginas en cada uno de los niveles de la jerarquía. Esta funcionalidad ha sido añadida a requerimiento de los alumnos que han evaluado la herramienta.

3. Simulador MNEME

Como dijimos anteriormente, para las asignaturas de Sistemas Operativos y Arquitectura de Computadores hemos diseñado un simulador que trata de superar las capacidades de SIMJE, permitiendo experimentar en todos y cada uno de los conceptos que dichas asignaturas estudian en lo referente a jerarquía de memoria. Este simulador se denomina MNEME [4] y se ha implementado en Java Swing. A continuación analizaremos cuáles son las prestaciones de este nuevo simulador en función de la asignatura a la que va destinado:

En Sistemas Operativos:

- Definición de procesos asociados a un número de unidades de tiempo de ejecución e implementación de las colas de espera y de ejecución.
- Algoritmos de reemplazamiento basados en políticas globales, haciendo referencia a todos los procesos: working set (page aging) y Page fault frequency (PFF), basado en dos parámetros max PFF y min PFF. Cuando un proceso es superior a maxPFF, entonces el mecanismo memory allocation toma, de forma aleatoria, una página de un proceso que tenga su page fault frequency más bajo que minPFF.
- Algoritmos de reemplazamiento basados en políticas locales, haciendo referencia exclusivamente a páginas del propio proceso: LRU, FIFO, Clock, LFU, MIN u OPT, NUR, NFU, MRU.

- Tabla de página con las implementaciones Top-down, Bottom-up e inversa.

En Arquitectura de Computadores:

- Niveles de cache: L1, L2 y L3 independientes, con un tamaño de línea específico para cada nivel.
- Dimensión de los buses.
- Write Allocate/non write allocate.
- Paso del paso: procedimiento gráfico de descomposición en pasos más elementales del proceso de búsqueda de una instrucción/dato.

3.1. Prácticas con el simulador MNEME

En este punto será tratada una secuencia de prácticas tipo para la asignatura de Sistemas Operativos de Segundo Curso de Ingeniería Técnica en Informática utilizando las capacidades docentes que aporta el simulador MNEME. Las prácticas versarán sobre los distintos algoritmos de paginación que pueden ser utilizados en un sistema operativo, poniendo de manifiesto sus particularidades a través del simulador.

Familiarización con el simulador

La primera práctica básica tiene como objetivo principal que el alumno aprenda a utilizar el simulador MNEME. Para ello se planteará el caso más sencillo de paginación, donde el alumno empezará a utilizar el programa en un caso muy conocido para él, fácil de seguir y predecir. En este primer caso se planteará un sistema de paginación jerárquica de dos niveles y un único proceso que utilice esta memoria. El código del proceso consistirá en una serie de accesos a memoria, algunos produciendo fallos de página y otros accediendo a páginas ya disponibles en memoria principal.

Tras esta primera sesión de prácticas el alumno debería haber aprendido a manejar el simulador, entender todas las tablas y gráficos, además de ser capaz de seguir una traza de accesos a memoria desde el código fuente del proceso al funcionamiento del simulador.

Paginación Multiproceso

En la segunda práctica se presenta un sistema multiproceso con el mismo esquema jerárquico de

paginación, consiguiendo introducir al alumno en el simulador cuando entran en escena varios procesos accediendo simultáneamente a memoria. Para esto hay que definir el quantum de tiempo que tiene cada proceso y realizar la traza de los accesos a memoria de cada uno de los procesos. Para seguir esta traza hay que tener en cuenta que un proceso será expropiado de la CPU en el caso de que se produzca un fallo de página o se acabe su quantum de tiempo de ejecución.

Distintas estrategias de la gestión de páginas

A continuación, una vez que el simulador de memoria y la estructura de procesos sean bien conocidos, se buscará profundizar en los distintos algoritmos de gestión de páginas. Para ello, se verá cómo cambia el comportamiento del sistema con los distintos algoritmos de sustitución de página. En este cometido se utilizará un código con un gran número de fallos de página y se definirá un tamaño de memoria principal pequeño para forzar a que el simulador envíe a memoria secundaria las páginas seleccionadas. Se probarán tanto estrategias de intercambio locales como globales explotando las posibilidades del simulador.

Sistemas de paginación avanzada

La última práctica se centrará en algoritmos avanzados de paginación como la "tabla de páginas inversa", el envejecimiento de páginas y la tabla de páginas lineal en memoria virtual, facilitando al alumno la comprensión de estas técnicas de gestión de memoria que son nuevas para él. También se utilizará la TLB y se comprobará el efecto de la variación del tamaño de ésta para la eficiencia del sistema.

Por último y de manera meramente informativa se realizará una simulación utilizando todas las características del programa, haciendo el seguimiento del dato desde la memoria. Se comprobará la funcionalidad de los distintos niveles de cache de la CPU y el funcionamiento global del sistema, abarcando desde la visión del sistema operativo hasta la visión de la estructura de la máquina.

Esta sesión de prácticas se ha centrado en el estudio de la paginación, obviando otras funcionalidades que aporta el simulador (que

serán utilizadas en otras asignaturas como el funcionamiento de las caches multinivel en Arquitectura de Computadores). Un simulador que permite la simulación de una amplia gama de técnicas permitirá aumentar la visión global del alumno en el funcionamiento de Arquitectura de Computadores desde el hardware hasta el sistema operativo.

3.2. Proceso de Validación

El Simulador SIMJE fue realizado en el año 2004, año en el que fue validado por los alumnos de ese curso académico, y se hicieron propuestas de mejora dirigidas fundamentalmente a la interfaz con el usuario. Desde entonces y una vez mejorado, ha sido utilizado ininterrumpidamente con resultados satisfactorios en los sucesivos cursos hasta la actualidad.

Sin embargo, en este apartado nos centraremos en el simulador MNEME. Para validarlo se dedicaron 4 grupos de alumnos de 5º curso de la Ingeniería Informática, con amplia experiencia en la utilización de simuladores y en programación. Realizaron una validación de tipo colaborativo por grupo, aunque cada miembro del grupo tuvo que completar una encuesta realizada de forma individual. En total hemos analizado un conjunto de 10 encuestas. Los resultados se comentan a continuación.

El 100% de los alumnos afirmó que al menos probaron el programa MNEME entre 3 y 6 horas antes de contestar el cuestionario. Dicho cuestionario constaba de una serie de preguntas referentes tanto a aspectos educativos y técnicos como de funcionalidad del software. También se les ofreció la oportunidad de que indicaran cuáles eran las mejoras que a su juicio deberían ser implementadas en el programa. Sobre este punto es importante destacar que el 90% afirmó que no conocían software alguno que fuera similar al nuestro.

En el aspecto educativo los alumnos afirmaron en un 90% que el software ayuda a comprender mejor los contenidos, mientras que el 70% dijeron que además es capaz de mantener el interés. Sin embargo también nos confirmaron que es necesario pulir la interfaz de usuario, pues el 100% de los usuarios consideró que el esquema de máquinas no se entendía fácilmente, a la vez que el 90% afirmó que la ayuda suministrada no era lo suficientemente clara. Es más, desde el punto de

vista de la funcionalidad del software, el 40% de los alumnos dijo que el contenido de la memoria no se muestra con suficiente detalle y claridad. Sin embargo es importante destacar que la inmensa mayoría de los encuestados consideraron que el programa tiene las opciones necesarias para ser lo suficientemente flexible de cara a diseñar nuevos ejercicios. En lo que respecta al aspecto técnico hay un consenso generalizado en que el programa funciona correctamente. Es decir, el 100% de los usuarios afirma que el programa no se cuelga y se ejecuta perfectamente en distintos ordenadores.

Finalmente se les pidió a los alumnos que indicaran aquellos aspectos que eran mejorables y propusieran algunas modificaciones. La mayor parte de los encuestados afirmaron que el software era interesante y que tiene un gran potencial de cara a su formación; al mismo tiempo que indicaron la necesidad de mejorar la experiencia del usuario. En este sentido los alumnos se inclinaron por mejorar la ayuda mediante la inclusión de un asistente que iniciara al usuario en la utilización del programa y la utilización de globos emergentes que informaran acerca de lo que está pasando en cada instante de la simulación.

Como consecuencia de esta evaluación realizamos una reunión conjunta los profesores implicados con el simulador y los alumnos que realizaron la encuesta. Fruto de esta reunión han surgido una serie de ideas innovadoras encaminadas a:

- mejorar la interfaz.
- crear un tutorial de utilización del simulador.
- crear una herramienta que permita de forma gráfica ensayar diferentes estrategias de caché multinivel, e incorporarlas al sistema como metadatos XML.
- incorporar en cada nivel de cache su inclusividad/exclusividad.
- incorporar al simulador la capacidad multiprocesadora y multicore.
- crear una herramienta asociada con el simulador que posibilite al profesor ocultar, dependiendo de los conocimientos del alumno, determinadas funcionalidades del programa.

Estas ideas ya se están abordando en estos momentos, con lo que se inicia nuevamente el proceso de investigación-acción.

4. Conclusión

A través de un proceso de investigación-acción hemos tratado de analizar rigurosamente cuáles son algunos de los problemas que nos encontramos en la impartición de las asignaturas de Estructura de Computadores, Sistemas Operativos y Arquitectura de Computadores, centrándonos en el tema de la Jerarquía de memoria. Hemos realizado una estrategia que ha consistido en 4 fases: análisis de los conceptos involucrados y creación de un mapa conceptual, una segunda fase de creación de un simulador MNEME que permita trabajar con todos los conceptos involucrados con la Jerarquía de memoria y que permita al alumno un aprendizaje constructivista. Hemos incorporado al proceso otro simulador sencillo y amigable, denominado SIMJE. El proceso se ha completado con dos fases más de validación del programa MNEME por parte de los alumnos a través de la realización de un cuestionario y una reunión mantenida entre profesores y alumnos que han permitido definir en el simulador nuevas características referidas a la interfaz, a la configuración de las caches multinivel, tanto en un medio monoprocesador, como multiprocesador y/o multicore.

Referencias

- [1] Almisas, R.; Paz, R.; Linares, A.; Amaya, C. y Sevillano, J.L. "Un simulador de memorias cache multinivel". Actas de las VII Jornadas de Enseñanza Universitaria de Informática, Jenui 2001, 429-433.
- [2] Anguita, M.; Fernández, A.; Linares, A.F.; Díaz, A.; Cañas A. y Prieto, A. "Práctica de optimización para asignaturas de Escritura de los Computadores". Actas de las X Jornadas de Enseñanza Universitaria de Informática, Jenui 2004.
- [3] González C., Alesanco F., Castilla I., Moreno L. "SIJEM: Una herramienta didáctica para la enseñanza de la Jerarquía de Memoria". 4ta. Conferencia Iberoamericana en Sistemas, Cibernética e Informática. CISCI 2005. Volumen III. Julio. 2005.

- [4] MNEME puede descargarse de: <http://www.isaatc.ull.es/portal/proyectos/MNEME>.
- [5] Molero, X.; Rodas, A.; Pont, A., Sahuquillo, J. y Valiente, J.M. "Diseño de experiencias prácticas sobre memoria caché" Actas de las VII Jornadas de Enseñanza Universitaria de Informática, Jenui 2001, 3-8.
- [6] Novak, J.D. Clarify with concept maps: A tool for students and teachers alike. *The Science Teacher*. 1991, 58(7), 45-49.
- [7] SIJEM puede descargarse de : <http://www.isaatc.ull.es/portal/proyectos/sijem>.
- [8] Vega Rodríguez, M.A.; Gómez Pulido J.A. y Sánchez Pérez, J.M. "Enseñanza de arquitecturas de memorias caché mediante simuladores". Actas de las VII Jornadas de Enseñanza Universitaria de Informática, Jenui 2001, 9-14.

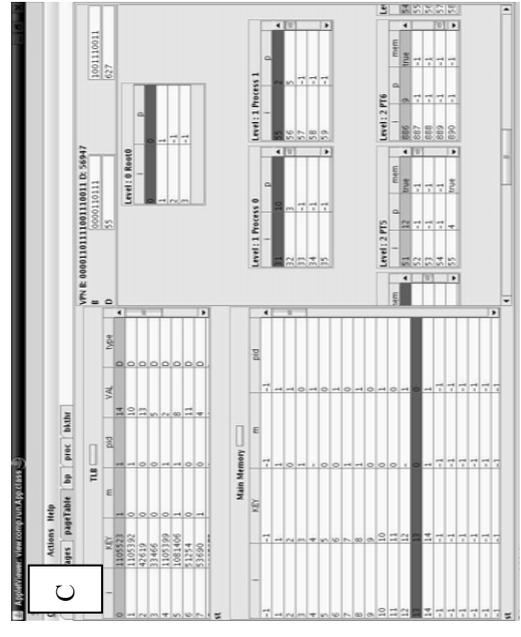
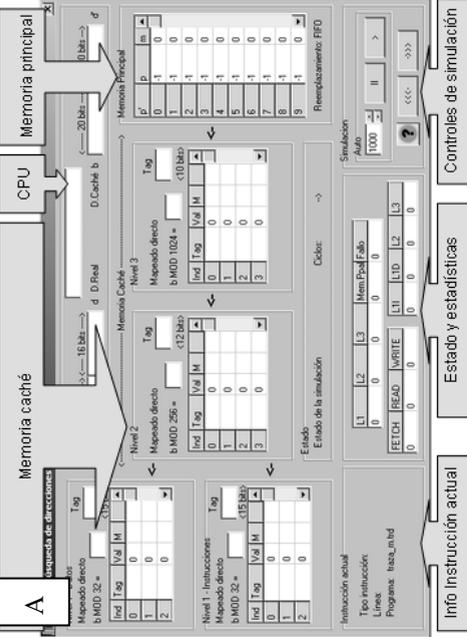
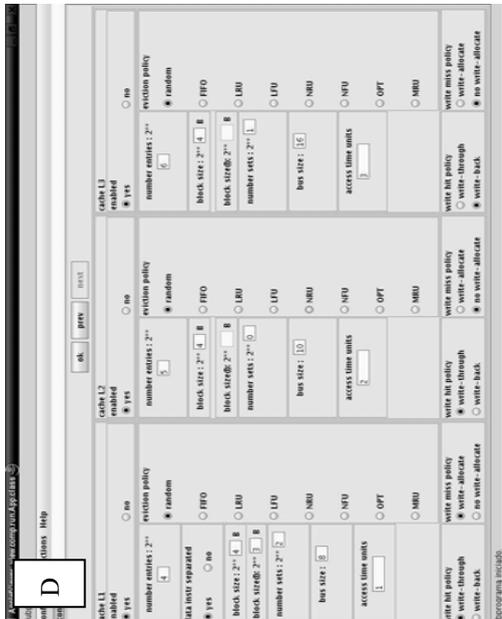
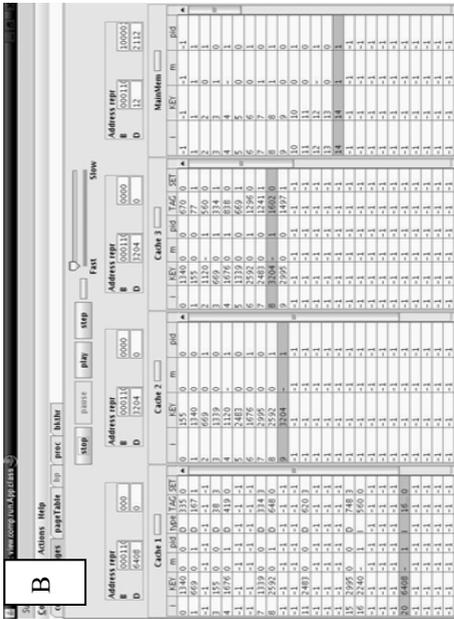


Figura 1. Diversas capturas de pantalla para los simuladores SIJEM (A) y MNEME: caches (B), tabla de páginas (C) y configuración de páginas (D)