

Sistemas Operativos 1

Simulación de la Gestión de Memoria

Edwin Salvador

07 de enero de 2016

Índice

1. Instrucciones Iniciales	2
2. Familiarizándose con el simulador	2
3. Correspondencia entre memoria virtual y física	4
3.1. Memoria virtual	5
3.2. Paginación	5
3.3. Asignación de memoria virtual con MOSS Simulador	5
3.4. Archivos del Simulador MOSS	5
3.4.1. Archivo de configuración	5
3.4.2. Archivos de comandos	7
3.4.3. Archivo de salida	7
4. Experimentos	8
5. Referencias	9

El simulador de gestión de memoria ilustra el comportamiento de los fallos de página en un sistema de memoria virtual paginada. El programa lee el estado inicial de la tabla de páginas y una secuencia de instrucciones de memoria virtual y escribe un registro de seguimiento que indica el efecto de cada instrucción. Incluye una interfaz gráfica de usuario para que los estudiantes puedan observar como funcionan los algoritmos de sustitución de páginas.

1. Instrucciones Iniciales

Las instrucciones de este laboratorio son para un equipo con Windows, pero los simuladores funcionan en cualquier ordenador que pueda ejecutar Java.

1. Ir a <http://www.ontko.com/moss/#memory>
2. Se puede leer guía de instalación pero aquí también se incluye algunas de las instrucciones de instalación.
3. Asegurarse de que se ha instalado Java en su máquina. Una forma de comprobar esto es abrir un símbolo del sistema y escribir `java -version`. Si necesita instalar Java, ir a <http://java.com/en/download/manual.jsp>, descargar e instalar el entorno de ejecución de Java.
4. Crear una carpeta en su computadora llamada MOSS. En esta carpeta, cree una carpeta llamada memoria.
5. Descargar el archivo ZIP autoextraíble desde el sitio web de referencia en el paso 1.
6. La mejor manera de hacerlo es hacer clic derecho sobre el enlace y seleccionar Guardar enlace como.
7. Extraiga los archivos del ZIP haciendo doble clic en el archivo memory.exe.
8. Entre los archivos extraídos, está una guía llamada user_guide.html. Leer esta guía para familiarizarse con el simulador.
9. Cuando se ejecuta cada ejercicio, debe registrar los resultados en archivos de registro separados. Colocar todos los archivos de registro en un solo archivo Word junto con su análisis indicando el número y el enunciado del ejercicio. Este archivo es el que deberá ser subido al Tunitin.
10. Inicie una ventana de comandos de MS-DOS.
11. Vaya al directorio donde se encuentra el simulador de memoria, utilizando el comando `cd`.

2. Familiarizándose con el simulador

El simulador de gestión de memoria se utiliza para simular el comportamiento de los fallos de página en un sistema de memoria virtual paginada. El programa traza un conjunto de instrucciones que son especificados en un archivo de comandos. El estado inicial de la tabla de páginas se especifica en un archivo de configuración. El simulador se puede ejecutar desde el símbolo del sistema de MS-DOS ejecutando:

```
java MemoryManagement commands memory.conf
```

Vamos a hablar de los archivos de comandos y de configuración más adelante.

La figura 1 ilustra como se ve la interfaz gráfica de usuario después de una corrida del simulador.

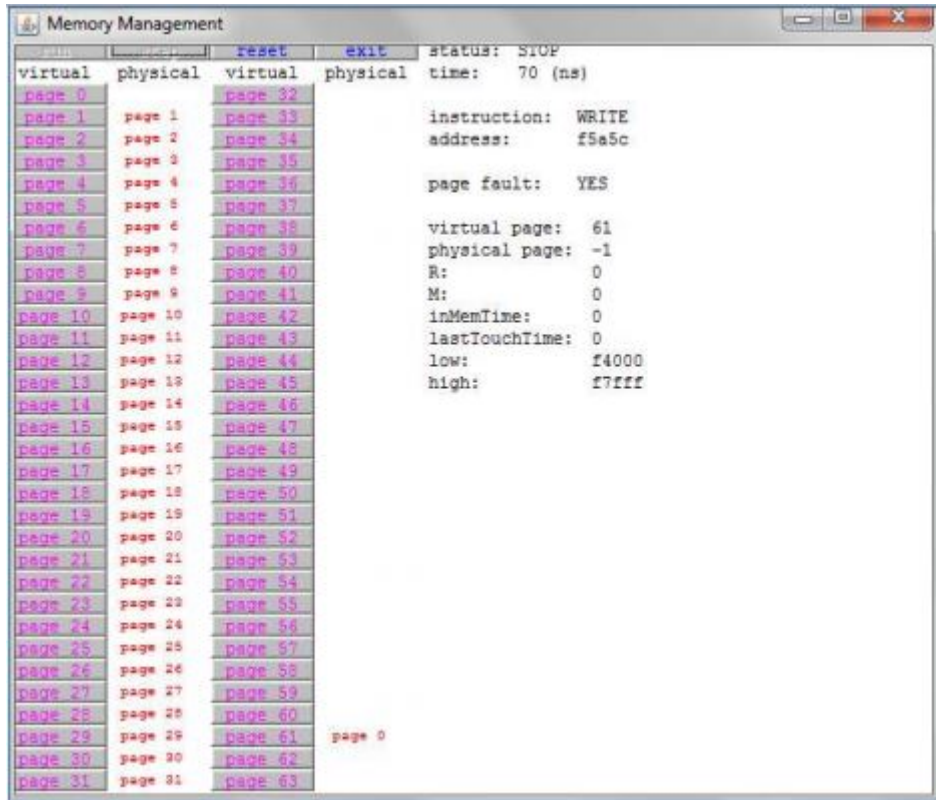


Figura 1: Interface del simulador

Echemos un vistazo a lo que muestra la interfaz gráfica de usuario. En la parte superior de la pantalla hay 4 botones, Ejecutar, Paso, Reiniciar y Salir. También hay una serie de botones etiquetados de Page0 a Page63. La Tabla 1 proporciona una descripción de cada una.

Tabla 1: Descripción de los botones

Botón	Descripción
Run	Ejecuta la simulación de todo, la actualización de la pantalla de cada paso del camino
Step	Ejecuta sólo un paso de la simulación a la vez y actualiza la pantalla. Cada vez que se pulsa el botón, se ejecuta el siguiente paso en la simulación.
Reset	Inicializa el simulador
Exit	Sale el simulador
Page n	Muestra información sobre la página virtual en el área de visualización

A la derecha de la pantalla hay una sección informativa. La Tabla 2 describe cada campo en esta área.

Tabla 2: Descripción de la sección informativa

Campo	Descripción
status	Indica si estamos corriendo una carrera completa, paso, o que todo ha terminado
time	nanosegundos que han transcurrido desde el inicio de la simulación. Si el estado se detiene, esto indica el tiempo total transcurrido para el simulación.
instruction	Operación que se realiza actualmente (ya sea de lectura o escritura)
address	Dirección de memoria virtual de la última operación realizada
Page fault	Muestra Yes, si la última operación provocó un fallo de página
virtual page	Todos los campos restantes aplican a la página virtual que se muestra aquí. Esta es la última página virtual visitada por el simulador o el número de página del botón pulsado.
physical page	La página física de la página virtual que se muestra arriba. Si es -1, no hay página física asociada a la página virtual.
R	Se establece en 1 si la página se ha sido leída, 0 si no ha sido.
M	Se establece en 1 si la página se ha modificado, 0 si no ha sido.
inMemTime	Número de nanosegundos desde que la página física se asignó a la página virtual.
lastTouchTime	Número de nanosegundos desde que la página física fue modificada.
low	El límite inferior de la dirección de memoria virtual de la página virtual
high	El límite superior de la dirección de memoria virtual de la página virtual

Realicemos una corrida de muestra:

Escribir en el CMD: `java MemoryManagement commands memory.conf`.

Ahora debería ver una pantalla similar a la Figura 1. Sólo por la práctica, pulse el botón step varias veces. Anote sus resultados para cada uno de los cuatro pasos en la Tabla 3.

Tabla 3: Resultados

Campo	Observación 1	Observación 2	Observación 3	Observación 4
Time				
Instruction				
Address				
Page fault				
Virtual Page				
Physical page				
R				
M				
InMemTime				
LastTouchTime				
Low				
High				

3. Correspondencia entre memoria virtual y física

A continuación tenemos una revisión de los conceptos de memoria virtual con el fin de ayudar a entender mejor cómo utilizar el simulador.

3.1. Memoria virtual

La memoria virtual hace que la memoria principal y secundaria (disco) parezcan como una unidad contigua de memoria a un programa de usuario. La memoria virtual se utiliza para permitir que los programas se ejecuten incluso cuando no hay suficiente memoria física dentro de un sistema y también para favorecer la compartición eficiente de espacio de memoria entre varios usuarios.

Cuando se realiza una solicitud de una dirección de memoria, la unidad de gestión de memoria (MMU) traduce la dirección de memoria virtual en una dirección de memoria física real o determina que la dirección de la memoria virtual requerida no está actualmente en la memoria física. En este último caso, el sistema operativo debe entonces intercambiar algo de memoria física con la memoria en el almacenamiento secundario de modo que la memoria virtual deseada puede ser ahora accedida. A esto se lo conoce como un fallo de página (page fault).

3.2. Paginación

Aunque hay muchas técnicas para implementar la memoria virtual, una de las más populares es la paginación. Esto es lo que vamos a examinar en el simulador. La memoria se divide en segmentos de tamaño fijo, llamados marcos de página. Cada proceso es capaz de acceder a la memoria en un segmento de tamaño fijo llamado página. Algunas de estas páginas se mantienen dentro de memoria física en marcos de página mientras que algunas páginas se almacenan en memoria secundaria, como una unidad de disco. La MMU es capaz de realizar la traducción entre la memoria física y la memoria virtual mediante el uso de una tabla de páginas, que normalmente se almacena en memoria física. Para entender mejor como funciona la paginación, leer la sección 3.3.1 y 3.3.2 páginas de la 189 a la 194 del libro Sistemas Operativos Modernos de Andrew Tanenbaum (disponible en el repo).

3.3. Asignación de memoria virtual con MOSS Simulador

Para simular la memoria virtual mediante una tabla de páginas, tendremos que configurar el simulador utilizando un archivo de comandos, que contiene una secuencia de instrucciones de memoria para realizar y un archivo de configuración, que especifica el contenido inicial del mapa de memoria virtual. Vamos a hablar de esto en más detalle.

3.4. Archivos del Simulador MOSS

Como se mencionó anteriormente, hay 2 archivos que son utilizados por el simulador para cada ejecución. También hay un archivo de salida, donde se registran los resultados.

3.4.1. Archivo de configuración

El archivo de configuración se llama `memory.conf`, y se utiliza para especificar el contenido inicial del mapa de memoria virtual. También se utiliza para proporcionar otra información tal como si la operación debe ser registrada. El comando `memset` se utiliza para inicializar cada entrada en el mapa de páginas de memoria virtual. Esta es seguida por seis números enteros, como se muestra en la Tabla 4.

Los dos primeros parámetros se utilizan para especificar la asignación de memoria entre la página física y la página virtual. Los últimos cuatro parámetros son valores que pueden ser utilizados por un algoritmo de sustitución de página. Veamos un ejemplo:

Tabla 4: Descripción de los números del comando `memset`

Posición del Entero	Significado
1	número de página virtual para inicializar
2	número de página física asociado a la página virtual. Esto se establece en -1 si no hay página asignada
3	Cero si la página no ha sido leída y uno si la página ha sido leída
4	Cero si la página no se ha modificado y uno si lo ha sido.
5	Número de nanosegundos que la página ha estado en la memoria
6	La última vez que la página ha sido modificada, en nanosegundos

```
memset 45 34 0 0 0 0
```

Esto le dice al simulador que la página virtual 45 está mapeada a la página física 34. La página no ha sido leída o modificada.

Al especificar el `memset`, tenemos que tener en cuenta que cada página física debe ser mapeada a una sola página virtual. Si no especificamos una página virtual en algún comando `memset` en el archivo de configuración, entonces se asume que no se asigna a una página física. Además, tenga en cuenta que hay 64 páginas virtuales y 64 páginas físicas.

Algunas otras opciones disponibles para el archivo de configuración:

1. **enable_logging** Esto se puede establecer en verdadero o falso. De forma predeterminada, se establece en falso. Si está habilitada, entonces un mensaje de una línea es escrito para cada operación de lectura o escritura.
2. **Log_file** El usuario puede especificar el nombre del archivo de registro si el registro está habilitado (ver arriba). Si el registro está habilitado y no se especifica ningún archivo de registro, la salida se escribe en la salida estándar.
3. **pagesize** Especifica el tamaño de una página en potencias de dos. Uno puede utilizar un número (1, 2, 4, 8, 16, etc.) o utilizar la palabra clave de potencia (es decir `power 2` es igual a 4.) El máximo tamaño de página es 67108864 o `power 26`. Este es también el tamaño de página por defecto si ninguno ha especificado.
4. **address_radix** El radix predeterminado en el que se muestran los números. Es 2 por defecto, que es binario. También se puede especificar radix 8 (octal), 10 (decimal) o 16 (hexadecimal).
5. **numpages** El número de páginas de memoria virtual y física. De forma predeterminada, esto es 64.

Echemos un vistazo a un archivo de configuración de ejemplo:

```
memset 0 0 0 0 0 0
memset 1 1 0 0 0 0
memset 2 2 0 0 0 0
memset 3 3 0 0 0 0
Enable_logging true
log_file tfile.out
pagesize 16384
addressradix 16
numpages 64
```

3.4.2. Archivos de comandos

El archivo de comandos especifica una secuencia de operaciones de lectura o escritura que el simulador puede realizar. El comando incluye la operación (lectura o escritura), así como la dirección de memoria virtual. Si la página virtual de la dirección está presente en la memoria física, la operación tendrá éxito, de lo contrario se producirá un fallo de página.

Los comandos tienen el formato siguiente:

```
read address
read random
write address
write random
```

La dirección es una dirección de memoria virtual, que puede tener un radix en frente de ella. Si no hay radix, el simulador asume números decimales. La palabra clave **random** genera una dirección de memoria virtual al azar, esto es útil si usted quiere experimentar rápidamente sin necesidad especificar una dirección.

Echemos un vistazo a un archivo de comandos de ejemplo:

```
READ bin 110
READ 20
WRITE hex CA32
READ bin 100000000101
READ bin 111000000001
WRITE bin 11000000110
WRITE random
```

Pasos:

1. Leer la dirección de memoria 6
2. Leer la dirección de memoria 20
3. Escribir en la dirección de memoria 51762
4. Leer la dirección de memoria 2053
5. Leer la dirección de memoria 3585
6. Escribir en la dirección de memoria 1542
7. Escribir a una dirección aleatoria

3.4.3. Archivo de salida

El archivo de salida sólo contiene una línea para cada operación y una indicación de si la operación estaba bien o provocó un fallo de página. Un archivo de salida de ejemplo podría tener el siguiente aspecto:

```
READ 10 ...okay
```

```
READ 23 ...okay
WRITE 3ac2...okay
WRITE 1000002...page fault
```

4. Experimentos

1. Mantenga `memory.conf` como inicialmente. Modifique el archivo de comandos, escriba la siguiente secuencia de comandos:

```
// Enter READ/WRITE commands into this file
// READ <OPTIONAL number type: bin/hex/oct> <virtual memory address or
random>
// WRITE <OPTIONAL number type: bin/hex/oct> <virtual memory address or
random>
READ bin 110
READ bin 111
WRITE hex CB33
WRITE hex FB12
WRITE hex B4A2B
READ bin 100000100100000
READ bin 110000010000110
WRITE bin 1100111000000000
WRITE random
```

Ahora, intente ejecutar el simulador (`java MemoryManagement commands memory.conf`). Pulse el botón de reinicio y luego el botón `run`. Echa un vistazo al archivo de registro. ¿Hay fallos de página? Si es así, ¿dónde éstos se producen, y por qué?

2. Modificar el archivo de comandos de nuevo e ingrese la siguiente secuencia de comandos:

```
READ bin 100
READ bin 010
READ bin 111
WRITE hex cc12
WRITE hex bc35
WRITE random
READ bin 111110100000
WRITE 6001
WRITE hex 7563e
```

Ahora, intente ejecutar el simulador (`java MemoryManagement commands memory.conf`). Pulse el botón de reinicio y luego el botón `run`. Echa un vistazo al archivo de registro. ¿Hay fallos de página? Si es así, ¿dónde éstos se producen, y por qué?

3. Considere un sistema de memoria virtual con un tamaño de página de 1024. Hay ocho páginas virtuales y cuatro marcos físicos. La tabla páginas se muestra a continuación:

Número de página Virtual	Número de marco de página
0	3
1	1
2	-
3	-
4	2
5	-
6	0
7	-

Guarde una copia del archivo `memory.config` original. Modifique el archivo `memory.config` de tal manera que refleje la tabla de página anterior. (Sugerencia: debe especificar númpáginas como 9.) Modifique el archivo de comandos para probar las siguientes operaciones:

```

READ 750
WRITE 1301
READ 2560
READ 4018
WRITE 4495
READ 5180
READ 6437
READ 7263

```

¿Cuál de estas direcciones virtuales causan fallos de página? ¿Por qué?

4. Modificar una copia del archivo `memory.config` original para mapear 8 páginas de memoria física cualquiera a las primeras 8 páginas de memoria virtual. Modificar una copia del archivo original de comandos para leer de una dirección de memoria virtual en cada uno de los 64 páginas virtuales. Ejecutar el simulador en el modo de un solo paso. ¿Qué direcciones de memoria virtual causan fallos de página?

Pregunta:

Basado en lo que has visto con los experimentos, ¿qué algoritmo de sustitución de página está siendo utilizado por el simulador de gestión de memoria MOSS?

5. Referencias

- Learn.saylor.org,. (2016). CS401: Unit 6 Activities. Retrieved 7 January 2016, from <https://learn.saylor.org/mod/page/view.php?id=927>