Universidad de los Andes
Ingeniería de Sistemas y Computación
ISIS 2203 Infraestructura Computacional
Semestre 2022-10

Caso de Estudio 2 – Memoria Virtual

Objetivos

- Entender la importancia de contar con una administración "apropiada" de la memoria: considerando número de procesos en ejecución, demanda del recurso por proceso, decisiones para adicionar y/o remover marcos de página asignados.
- Construir un prototipo a escala del sistema de administración de memoria virtual que permita simular y evaluar el comportamiento de un proceso de acuerdo con los recursos disponibles.

Problemática:

La memoria es un recurso limitado que debe administrarse con cuidado para garantizar el avance en la ejecución de los procesos creados. En este contexto surge el concepto de memoria virtual, el cual ofrece varias ventajas: independencia de direcciones físicas, posibilidad de compartir memoria y de correr programas más grandes que la memoria física que se les asigna. Queremos comprender un poco mejor cómo varía el comportamiento de un proceso de acuerdo con la memoria RAM que el sistema le asigna. Su tarea en este caso es escribir un programa en Java que simule el sistema de paginación utilizando el algoritmo de envejecimiento. Tanebaum explica este algoritmo en su libro Sistemas Operativos Modernos, capítulo 3 – sección "Simulación de LRU en software" (el libro está disponible en la biblioteca de la universidad).

Tareas:

Para simplificar el problema del manejo de memoria (y la solución) nos concentraremos en el manejo de las solicitudes de páginas para un solo proceso que suma matrices. El proceso puede ejecutar cualquiera de los siguientes métodos:

```
public void recorrido1() {
    for(int i = 0; i < num_filas; i++){
        for(int j = 0; j < num_cols; j++){
            mat3[i][j] = mat1[i][j] + mat2[i][j];
        }
    }
}

public void recorrido2() {
    for(int j = 0; j < num_cols; j++){
        for(int i = 0; i < num_filas; i++){
            mat3[i][j] = mat1[i][j] + mat2[i][j];
        }
    }
}
```

El programa debe simular el comportamiento del sistema de paginación mientras el proceso corre, incluyendo tomar decisiones de reemplazo con base en el algoritmo de envejecimiento. Además, el programa debe llevar el conteo de número de fallas de página generadas para indicarlo al final de la ejecución. No es necesario actualizar el contenido de los marcos de página y el swap, solo llevar registro del cambio de estado de las páginas.

El programa debe tener dos opciones para ejecución:

- La opción 1 debe pedir los parámetros de configuración y generar un archivo. El archivo debe tener: los parámetros de configuración, el número de páginas virtuales necesarias para almacenar las 3 matrices (NP) y las referencias de página que el proceso generará en ejecución. Los parámetros son: Tamaño de una página (TP), Tamaño de un número entero (TE), Número de filas de las matrices (NF), Número de columnas de las matrices (NC) y Tipo de recorrido (TR, 1 para recorrido1 o 2 para recorrido 2). El Anexo B, al final del documento, muestra un ejemplo del archivo que se debe generar.
- La opción 2 debe ejecutar dos threads de forma concurrente:
    - Un thread se encargará de ir actualizando el estado de la tabla de páginas y del registro de acceso a los marcos de página en memoria real de acuerdo con las referencias generadas por el proceso y el número de marcos de página asignados. Este thread debe correr cada dos milisegundos (en vez de pulsos de reloj usaremos milisegundos).

- El otro thread se encargará de ejecutar el algoritmo de envejecimiento (con base en el esquema presentado por Tanenbaum). Este thread debe correr cada milisegundo (en vez de pulsos de reloj usaremos milisegundos).
- Tenga en cuenta que los dos threads necesitarán compartir una o varias estructuras de datos por eso será necesario usar sincronización en algunos métodos (usted debe identificar cuáles).
- Esta opción debe pedir el nombre del archivo de referencias y el número de marcos de página para la simulación (MP). Al final se debe generar un mensaje en consola que indique la configuración, los marcos de página y el número de fallas de página.

Escriba un informe que incluya:

- Descripción del algoritmo usado para generar las referencias de página (opción 1)
- Descripción de las estructuras de datos usadas para simular el comportamiento del sistema de paginación y cómo usa dichas estructuras (cuándo se actualizan, con base en qué y en qué consiste la actualización).
- Esquema de sincronización usado. Justifique brevemente dónde es necesario usar sincronización y por qué.
- Una tabla con los datos recopilados (número de fallas de página por cada caso simulado).
- Una serie de gráficas que ilustren el comportamiento del sistema. Para eso cree gráficas en las que fije: tamaño de la matriz, tamaño de entero y tipo de recorrido, y grafique: tamaño de página vs. número de marcos de página vs. número de fallas de página. La gráfica en el Anexo A, al final del documento, ilustra el tipo de gráfica que buscamos.
- Entre los casos considerados incluya: Tamaño de página 8, 16 y 32; Tamaño de entero 1 y 2; Tamaño de matriz 4x4 y Marcos asignados 3, 6, 18 y 36. Además, considere otras configuraciones que le permitan entender cómo funciona la memoria virtual y el impacto del número de marcos de página en RAM asignados por el sistema.
- Escriba su interpretación de los resultados: ¿tienen sentido? Justifique su respuesta.

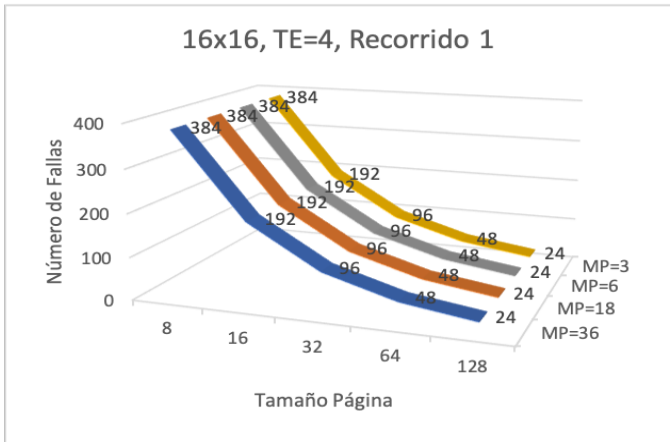
Entrega:

- Cada grupo debe entregar un zip de un proyecto Java que implemente el programa descrito (opciones 1 y 2). El proyecto Java debe incluir un subdirectorio docs con el informe en formato Word o pdf. **Al comienzo del informe, deben estar los nombres y carnés de los integrantes del grupo**. Si un integrante no aparece en el documento entregado, el grupo podrá informarlo posteriormente. Sin embargo, habrá una penalización: la calificación asignada será distribuida (dividida de forma equitativa) entre los integrantes del grupo.
- El trabajo se realiza en grupos de **2** personas. No debe haber consultas entre grupos.
- El grupo responde solidariamente por el contenido de todo el trabajo, y lo elabora conjuntamente (no es trabajo en grupo repartirse puntos o trabajos diferentes). Se puede solicitar una sustentación a cualquier miembro del grupo sobre cualquier parte del trabajo. Dicha sustentación será parte de la calificación de todos los miembros.
- El proyecto debe ser entregado en bloqueneon por uno solo de los integrantes del grupo.
- **La fecha límite de entrega 4 de abril, 2022 a las 23:50 p.m.**

Referencias:

- *Notas del curso.*
- *Sistemas Operativos Modernos. Andrew Tanenbaum. Editorial Pearson. Edición 3, año 2009.*

Anexo A



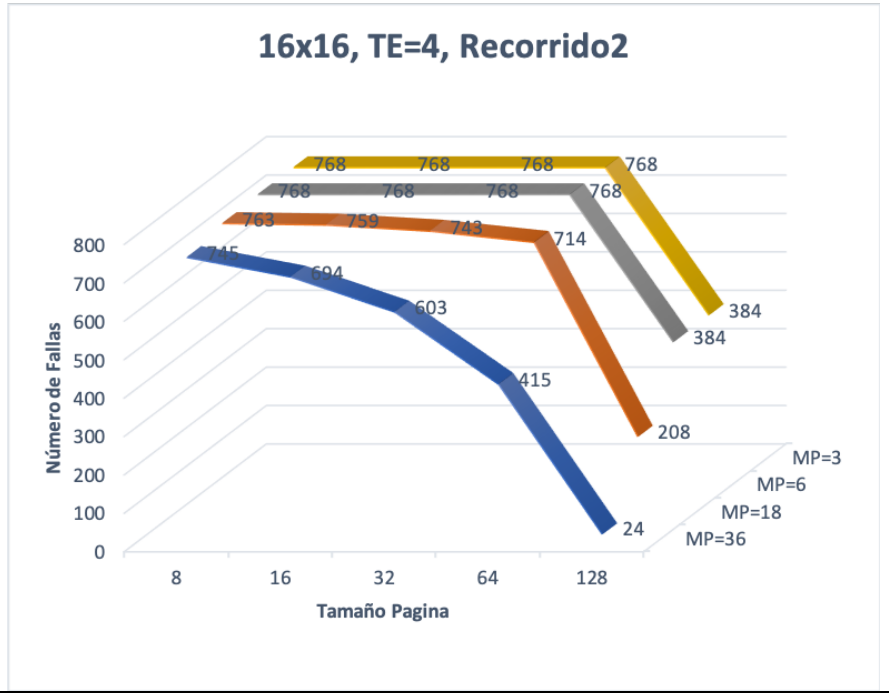


Figura 1. Tamaño de página vs. Marcos de página vs. Número de fallas. Configuración: matrices de 16x16 enteros, tamaño de entero 4B. A la izquierda recorrido tipo 1, a la derecha recorrido tipo 2.

Anexo B

En la tabla siguiente se describe el archivo de referencias generado por la opción 1. La columna de la izquierda muestra el contenido del archivo y la columna de la derecha describe el significado/propósito de cada línea del archivo.

Archivo	Descripción
TP=8	Tamaño de página – no es necesario para la simulación, se incluye por claridad
TE=2	Tamaño de entero – no es necesario para la simulación, se incluye por claridad
NF=4	Número de filas – no es necesario para la simulación, se incluye por claridad
NC=4	Número de columnas – no es necesario para la simulación, se incluye por claridad
TR=1	Tipo de recorrido – no es necesario para la simulación, se incluye por claridad
NP=12	Número de páginas
NR=48	Número de referencias – no es necesario para la simulación, se incluye por claridad

A:[0-0],0,0 B:[0-0],4,0 C:[0-0],8,0 A:[0-1],0,2 B:[0-1],4,2 C:[0-1],8,2 A:[0-2],0,4 B:[0-2],4,4 C:[0-2],8,4 A:[0-3],0,6 B:[0-3],4,6 C:[0-3],8,6 A:[1-0],1,0 B:[1-0],5,0 C:[1-0],9,0 A:[1-1],1,2 B:[1-1],5,2 C:[1-1],9,2 A:[1-2],1,4 B:[1-2],5,4 C:[1-2],9,4 A:[1-3],1,6 B:[1-3],5,6 C:[1-3],9,6 A:[2-0],2,0 B:[2-0],6,0 C:[2-0],10,0 A:[2-1],2,2 B:[2-1],6,2 C:[2-1],10,2 A:[2-2],2,4 B:[2-2],6,4 C:[2-2],10,4 A:[2-3],2,6 B:[2-3],6,6 C:[2-3],10,6 A:[3-0],3,0 B:[3-0],7,0 C:[3-0],11,0 A:[3-1],3,2 B:[3-1],7,2 C:[3-1],11,2 A:[3-2],3,4 B:[3-2],7,4 C:[3-2],11,4 A:[3-3],3,6 B:[3-3],7,6 C:[3-3],11,6	Cada una de las referencias generadas incluye cuatro valores: 1. A, B y C representan las matrices 2. Los números entre corchetes cuadrados indican la celda de la matriz: p.e. [0-0] representa la fila 0, columna 0 3. A continuación aparece la página de memoria virtual y el desplazamiento donde se encuentra almacenada la celda correspondiente. Suponemos que las tres matrices están almacenadas en orden y seguidas, empezando en la página 0: p.e. 0,0 representa la página 0, desplazamiento 0. El único valor que es estrictamente necesario para calcular el número de fallas de página es el tercer valor (que corresponde al número de la página virtual), pero los otros valores dan claridad para verificar que el resultado está bien.