Universidad de los Andes
Ingeniería de Sistemas y Computación
ISIS 2203 Infraestructura Computacional
Semestre 2024-10

Caso de Estudio 2 – Memoria Virtual

Objetivos

- Entender la importancia de contar con una administración "apropiada" de la memoria: considerando infraestructura de soporte, número de procesos en ejecución, demanda del recurso por proceso y 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 que el sistema le asigna.

Su tarea en este caso es escribir un programa en Java que simule el sistema de paginación usando el algoritmo de reemplazo "Paginas no usadas recientemente". Tanebaum explica este algoritmo en su libro Sistemas Operativos Modernos, capítulo 3 – sección 3.4.2 "El algoritmo de reemplazo de páginas: no usadas recientemente" (disponible en versión electrónica en la biblioteca).

Tareas:

1. Escribir un programa que simule la administración de memoria y calcule el número de fallas de página y el porcentaje de *hits* de datos en RAM (un hit es la búsqueda de una página que sí está en la RAM y un miss es la búsqueda de una página que genera una falla de página).
2. Además, calcule el tiempo con hits y *misses* vs el tiempo si todas las referencias estuvieran en RAM vs el tiempo si todas las referencias condujeran a fallas de página.
3. Analice los resultados y escriba el informe correspondiente.

Contexto:

Para simplificar el problema del manejo de memoria, y así poder concentrarnos en entender los retos asociados, estudiaremos las referencias de páginas generadas por un solo proceso que aplica un filtro a una matriz de datos.

La siguiente figura ilustra el cálculo a tener en cuenta.

```
for (int i=1; i<nf-1; i++) {
    for (int j=0; j<nc-1; j++) {
        // Recorrer los vecinos y aplicar el filtro
        // mat1: matriz de datos
        // mat2: matriz con el filtro (usaremos un filtro de 3x3 para resaltar bordes)
        // mat3: matriz resultante
        int acum = 0;
        for (int a=-1;a<=1;a++) {
            for (int b=-1;b<=1;b++) {
                int i2 = i+a;
                int j2 = j+b;
                int i3 = 1+a;
                int j3 = 1+b;
                acum += (mat2[i3][j3]*mat1[i2][j2]);
            }
        }
        if (acum>=0 && acum <=255)
            mat3[i][j] = acum;
        else if (acum<0)
            mat3[i][j] = 0;
        else
            mat3[i][j] = 255;
    }
}
// se asigna un valor predefinido a los bordes
for (int i=0;i<nc;i++ ) {
    mat3[0][i]=0;
    mat3[nf-1][i]=255;
}
for (int i=1;i<nf-1;i++ ) {
    mat3[i][0]=0;
    mat3[i][nc-1]=255;
}
```

Tenga en cuenta:

- El filtro siempre será una matriz de 3x3.
- Los datos de entrada pueden cambiar entre ejecuciones. Es decir, el tamaño de las matrices puede cambiar (número de filas y número de columnas).

Para calcular los tiempos de interés usaremos los siguientes tiempos de acceso:

- Tiempo de lectura para datos que están en la RAM: 30 ns
- Tiempo de lectura para datos que están en SWAP (resolución de una falla de página): 10 ms

El programa debe tener un menú alfanumérico con dos opciones:

- Generación de las referencias.
    - Esta opción recibe como parámetros (por consola) tamaño de página y tamaño de la matriz de datos y genera un archivo con las referencias correspondientes (las referencias que el método de multiplicación generaría). La lista de referencias debe mostrar las referencias de página que el proceso generaría en ejecución.
    - Tenga en cuenta que la matriz de datos y la matriz de resultado tienen el mismo tamaño. El filtro siempre será de 3x3.
    - Supondremos que las matrices se almacenan por filas (esto se conoce como row-major order), desde la página virtual 0 y en el siguiente orden: filtro, datos, resultado. Tenga en cuenta que, dependiendo del tamaño de las matrices y las páginas, podemos tener varias matrices en una misma página o parte de una matriz en una página y parte en otra. Tomaremos 4 bytes como el tamaño de un entero.
    - El archivo de referencias que se genere en esta opción debe incluir los siguientes datos: TP: tamaño de página, NF:#filas matriz de datos, NC: #columnas matriz de datos, NR:#referencias en el archivo y NP:# páginas virtuales del proceso (considerando solo lo necesario para almacenar las tres matrices). Después de estos datos deben estar las referencias generadas.

- Calcular datos: número de fallas de página, porcentaje de hits, tiempos.
    - Esta opción recibe como parámetros (por consola): número de marcos de página y nombre del archivo de referencias y calcula los datos resultantes.
    - En este caso el programa debe simular: (1) el comportamiento del proceso – es decir, cargar las referencias una por una y (2) el comportamiento del sistema de paginación, incluyendo identificar

cuándo hay falla de página y tomar decisiones de reemplazo con base en el algoritmo de envejecimiento.

- Además, el programa debe llevar el conteo de número de hits y fallas de página a medida que ocurren (solamente para lecturas y escrituras de datos - no consideraremos código, heap, ni stack). Observe que este modo recibe como entrada el archivo generado por la opción anterior.

Adicionalmente, en la segunda opción, el programa debe correr dos threads de forma concurrente:

- Un thread se encargará de ir actualizando el estado de la tabla de páginas y los marcos de página en RAM, de acuerdo con las referencias generadas por el proceso y el número de marcos de página asignados. Este thread debe correr cada milisegundo (en vez de pulsos de reloj usaremos milisegundos).
- El otro thread se encargará de ejecutar el algoritmo de actualización del bit R (con base en el esquema presentado por Tanenbaum). Este thread debe correr cada cuatro milisegundos (en vez de pulsos de reloj usaremos milisegundos).
- Para reducir la variación en los resultados no seleccione la página a reemplazar de forma aleatoria, busque en orden desde la primera entrada en su estructura de datos.
- 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).

Escriba un informe que incluya:

- Descripción del algoritmo usado para generar las referencias de página (modo uno)
- 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 (porcentaje de *hits* y *misses* por cada caso simulado).
- Una serie de gráficas que ilustren el comportamiento del sistema. Para eso muestre gráficas donde fije tamaño de página y grafique tamaño de matriz vs. Marcos vs. Porcentaje de hits. La gráfica al final del enunciado ilustra el tipo de gráfica que buscamos. Genere gráficas que muestren los datos para diferentes tamaños de matrices.
- Entre los casos considerados incluya tamaño de matriz 4x4, 8x8, 16x16, 32x32 variando tamaño de página y marcos asignados por el sistema. Además, considere otras configuraciones que le permitan entender cómo afecta la memoria virtual el desempeño del programa.
- Adicione las gráficas de tiempo.
- Escriba su interpretación de los resultados: ¿tienen sentido? Justifique su respuesta.

Entrega:

- Cada grupo debe entregar un zip de un proyecto Java con los archivos Java con la implementación correspondiente. En el subdirectorio docs debe estar 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 los grupos asignados para el caso 2. 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 8 de abril, 2024 a las 23:50 p.m.**

Información adicional:

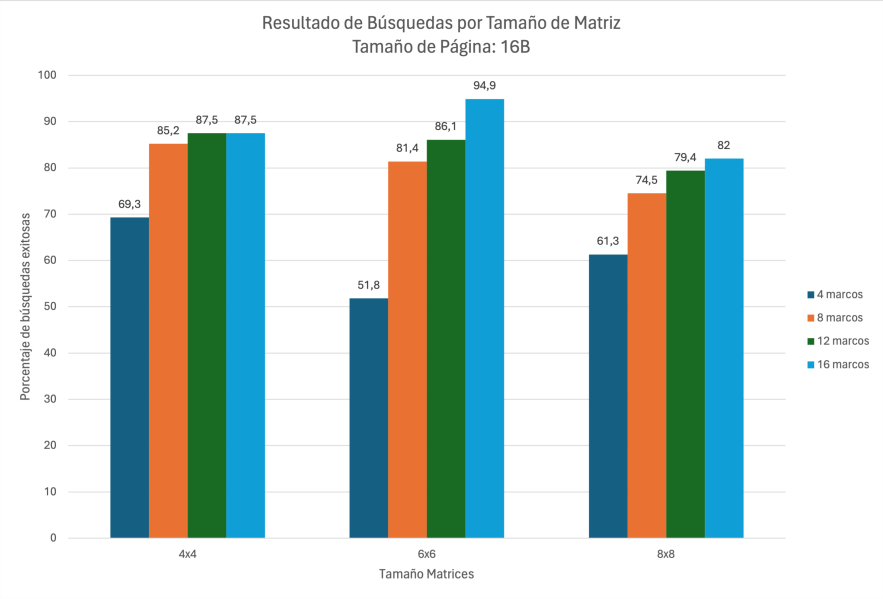
- **Anexo A.**

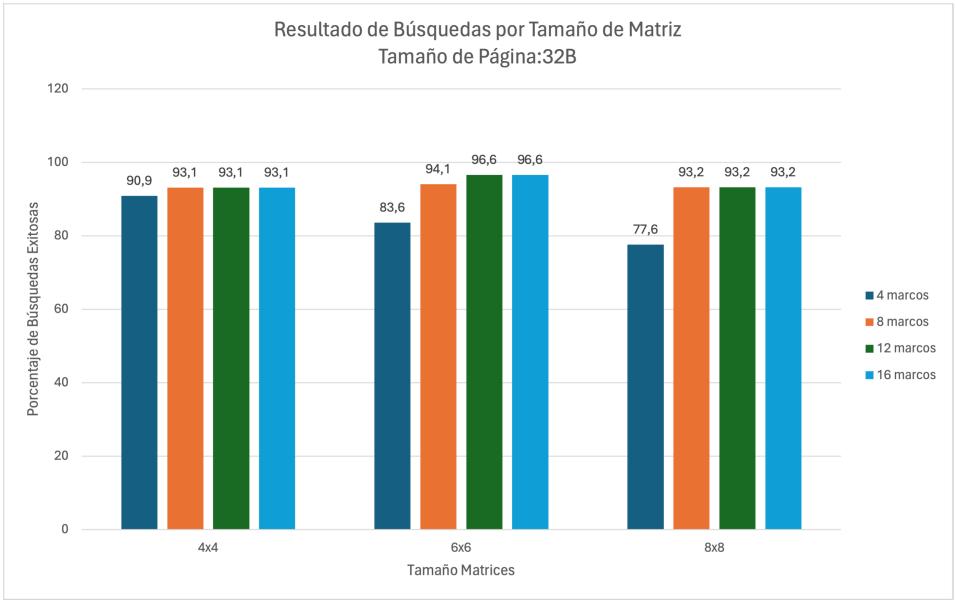
A continuación, se muestra el formato del archivo que debe generar el modo 1 como salida.

Datos archivo	Comentario
TP=16 NF=4 NC=4 NF_NC_Filtro=3 NR=88 NP=11 M[0][0],2,4,R F[0][0],0,0,R M[0][1],2,8,R F[0][1],0,4,R M[0][2],2,12,R F[0][2],0,8,R M[1][0],3,4,R F[1][0],0,12,R M[1][1],3,8,R F[1][1],1,0,R M[1][2],3,12,R F[1][2],1,4,R M[2][0],4,4,R F[2][0],1,8,R M[2][1],4,8,R F[2][1],1,12,R M[2][2],4,12,R F[2][2],2,0,R R[1][1],7,8,W M[0][1],2,8,R ...	Tamaño de página NF y NC: Número de filas y columnas de la matriz de datos y La matriz de resultado. Tamaño del filtro Número de registros Número de páginas (necesarias para almacenar el filtro y las 2 matrices) M: Matriz de datos F: Filtro R: Matriz de resultados Cada referencia tiene 4 campos (separados por ",”): Campo 1: matriz y posición. Esta información no es estrictamente necesaria para calcular hits y fallas, pero la incluimos por claridad. Campo 2: página virtual correspondiente Campo 3: desplazamiento Campo 4: bit de acción (R: lectura, W: escritura)

Anexo B.

A continuación, se presenta un ejemplo del tipo de gráfica que debe incluir en el informe, tenga en cuenta que esto es un ejemplo y el informe debe considerar más casos. También se presenta la tabla de datos, considere que, por el manejo de concurrencia, sus resultados pueden variar un poco con respecto a los valores presentados, pero debería estar en el rango +/- 10% (el conteo, no el porcentaje).





Páginas de 16B, matriz 4x4			
Marcos Asignados	Total referencias	Hits	Fallas
4	88	61	27
8	88	75	13
12	88	77	11
16	88	77	11
Páginas de 16B, matriz 6x6			
Marcos Asignados	Total referencias	Hits	Fallas
4	324	168	156
8	324	264	60
12	324	279	45
16	314	298	16
Páginas de 16B, matriz 8x8			
Marcos Asignados	Total referencias	Hits	Fallas
4	712	437	275
8	712	531	181
12	712	566	146
16	712	584	128
Páginas de 32B, matriz 4x4			
Marcos Asignados	Total referencias	Hits	Fallas
4	88	80	8
8	88	82	6
12	88	82	6
16	88	82	6

Páginas de 32B, matriz 6x6			
Marcos Asignados	Total referencias	Hits	Fallas
4	324	271	53
8	324	305	19
12	324	313	11
16	324	313	11
Páginas de 32B, matriz 8x8			
Marcos Asignados	Total referencias	Hits	Fallas
4	712	553	159
8	712	664	48
12	712	664	48
16	712	664	48

Referencias:

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

Cronograma Propuesto (se propone completar las actividades a más tardar en las siguientes fechas) :

13 marzo: Lectura del enunciado y acuerdo del contrato
16 marzo: Arquitectura general de la solución (diagrama de clases)
2 abril: Estrategia de solución
5 abril: Implementación + pruebas
7 abril: Informe y entrega
10 abril: Realizar Coevaluación