Universidad de los Andes
Ingeniería de Sistemas y Computación
ISIS 2203 Infraestructura Computacional

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.
- Comprender cómo varía el comportamiento de un proceso (con base en número de fallas de página generadas), de acuerdo con la memoria que el sistema le asigna.

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 asignada.

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 utilizando el algoritmo de envejecimiento. Tanebaum explica este algoritmo en su libro Sistemas Operativos Modernos, capítulo 3 – sección 3.4.7 "Simulación de LRU en software" (disponible en versión electrónica en la biblioteca).

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. Además, no es necesario encargarse de actualizar el contenido de los marcos de página y el swap, solo llevar registro del cambio de estado de la tabla de páginas y la memoria RAM.

Su programa debe leer los datos de configuración de un archivo de entrada. Cada archivo de entrada debe tener los siguientes datos (en el orden indicado):

- Número de marcos de página en memoria RAM que el sistema le asigna al proceso
- Número de páginas del proceso
- Nivel de localidad. Manejaremos un porcentaje (un número entre 0 y 100) que indicará el porcentaje de referencias que se concentrarán en el 25% de las páginas del programa. Observe que introducimos este parámetro para poder simular el comportamiento de programas con mayor o menor nivel de localidad.
- Secuencia de referencias a páginas del proceso. Tendremos una referencia por línea y usaremos un total de 1000 referencias en cada caso.

El programa debe simular el comportamiento del sistema de paginación para responder a cada una de las referencias generadas por el proceso. Este comportamiento incluye tomar decisiones de reemplazo con base en el algoritmo de envejecimiento. Al terminar la ejecución el programa debe presentar en la consola el número de fallas de página generadas.

Tenga en cuenta que usted debe correr su programa para varios archivos de configuración. Con este enunciado se entregarán cuatro archivos de configuración con las respectivas respuestas para que pueda evaluar su programa. Sin embargo, usted debería generar más archivos de configuración para poder construir las gráficas que permitan ilustrar la variación en número de fallas de acuerdo con las características de los procesos y la memoria que el sistema asigna.

Su programa debe correr dos threads:

- El primer thread se encargará de ir actualizando el estado de la tabla de páginas y los marcos de página en memoria real de acuerdo con las referencias generadas por el proceso. Este thread debe cargar una nueva

referencia cada cinco milisegundos. En un sistema real, esta operación ocurriría de forma aleatoria, cada vez que un proceso genera una nueva referencia.

- El segundo thread se encargará de ejecutar el algoritmo de envejecimiento (con base en el esquema presentado por Tanenbaum). Este thread debe correr cada milisegundo. En el sistema descrito por Tanenbaum este procedimiento corre cada pulso de reloj.
- 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 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 una gráfica por cada tamaño de programa estudiado y en cada una ilustre: número de fallas de página vs. número de marcos asignados vs. nivel de localidad.
- 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 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 por Sicua+ por uno solo de los integrantes del grupo.
- **La fecha límite de entrega es el 12 de abril, 2021 a las 23:55 p.m.**

Referencias:

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

Archivos de configuración anexos:

- referencias1.txt - 778 fallas de página
- referencias2.txt - 595 fallas de página
- referencias3.txt - 752 fallas de página
- referencias4.txt - 581 fallas de página