

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 soporte a la memoria virtual, en particular TLB (Translation lookaside buffer) y TP (tabla de páginas), para entender el impacto de estos componentes sobre el desempeño de un proceso.

Problemática:

La memoria RAM 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 tiempo de ejecución de un proceso de acuerdo con las características de un sistema de memoria virtual, en particular, TLB y espacio asignado en RAM. Su tarea en este caso es escribir un programa en Java que calcule el tiempo total que invierte el sistema para resolver direcciones virtuales y para cargar las páginas necesarias en memoria RAM, incluyendo el tiempo la resolución de fallas de página para que un proceso se ejecute. Para la resolución de fallas de página usaremos el algoritmo de envejecimiento; Tanenbaum 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, edición 3, año 2009).

Tareas:

1. Escribir un programa que simule la administración de memoria y calcule el tiempo que se invierte en la traducción de direcciones virtuales y en cargar datos/páginas en la memoria RAM.
2. Analizar los resultados y escribir el informe correspondiente.

Contexto:

Para simplificar el problema de la administración de memoria (y la solución) solamente simularemos el manejo de memoria de un proceso. Además, no es necesario actualizar el contenido de los marcos de página y el swap, solo llevar registro del estado de la TLB y la tabla de páginas.

Su programa debe pedir los datos para la simulación, estos datos son:

- Número de entradas de la TLB
- Número de marcos de página en memoria RAM que el sistema le asigna al proceso
- Nombre del archivo con las referencias (este archivo tiene la lista de referencias de página que el proceso genera). El archivo debe tener 1024 referencias para un programa cuyo tamaño corresponde a 64 páginas. La distribución de las referencias puede corresponder a un proceso con localidad ALTA o BAJA.

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

Direcciones:

- Tiempo de respuesta para consultar una dirección que sí está en la TLB: 2 ns
- Tiempo de respuesta para consultar una dirección en la TP (no está en la TLB): 30 ns
- Tiempo si hay falla de página: 2 lecturas en la TP (una cuando falla y una cuando se resuelve)

Datos:

- Tiempo de respuesta para datos que están en la RAM: 30 ns
- Tiempo para resolución de una falla de página: 10 ms

Para cargar datos en la TLB use un algoritmo FIFO, teniendo en cuenta que si un dato ya está cargado no es necesario volver a cargarlo.

Su programa debe correr dos threads:

- Un thread se encargará de correr 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).
- El otro thread cargará las referencias del proceso e irá actualizando el estado de la TLB, la RAM y la tabla de páginas. Este thread debe correr cada dos milisegundos (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).

El programa debe simular el comportamiento del sistema de administración de memoria, incluyendo tomar decisiones de reemplazo de páginas con base en el algoritmo de envejecimiento. Además, el programa debe calcular los tiempos de interés (resolución de direcciones virtuales y cargar páginas en memoria).

El informe debe incluir:

- 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 (tiempos).
- Una serie de gráficas que ilustren el comportamiento del sistema. Para eso cree gráficas en las que muestre:
 - Número de marcos de página asignados vs. Tamaño de TLB vs. Tiempo para cargar datos
 - Número de marcos de página asignados vs. Tamaño de TLB vs. Tiempo para resolver direcciones
- Entre los casos considerados incluya: Número de marcos: 8, 16, 32; Tamaño de TLB: 8, 16, 32 y 64 entradas. Además, considere otras configuraciones que le permitan entender cómo cambia el tiempo de ejecución con base en la configuración de la memoria.
- 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 fuente. Además, recuerde crear un subdirectorio docs para almacenar el informe en formato Word o pdf. **Escriba nombres y carnés de los integrantes del grupo al comienzo del informe** (necesitamos esta información para poder asignar la calificación). 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 definidos en el curso para este caso. No debe haber consultas entre grupos.
- Habrá una coevaluación del trabajo en grupo. Cada miembro del grupo evaluará a sus dos compañeros y las notas recibidas por un estudiante ponderarán la nota final de caso para ese estudiante. Si un grupo se disuelve, la calificación del caso se recalculará así:
 - Si la disolución se reporta al profesor durante los últimos tres días hábiles antes de la fecha límite de entrega, entonces la calificación de la entrega se multiplicará por 0,5 (50%)
 - Si la disolución se reporta, se justifica de forma apropiada, y el profesor acepta la disolución con anterioridad a los últimos tres días hábiles antes de la fecha límite de entrega, entonces la calificación de la entrega se multiplicará por 0,75 (75%)
- En el parcial se incluirá una pregunta sobre el desarrollo de alguna de las funcionalidades del caso.
- 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. Tenga en cuenta que desarrollar sus habilidades para trabajo en grupo le puede ayudar en su futuro profesional para integrarse más fácilmente a un grupo de trabajo y trabajar de forma productiva.
- El proyecto debe ser entregado en bloque por uno solo de los integrantes del grupo.
- **La fecha límite de entrega es el 14 de octubre, 2022 a las 23:50 p.m.**

Información adicional:

- A continuación, se ilustra el tipo de gráficas esperadas en el informe.

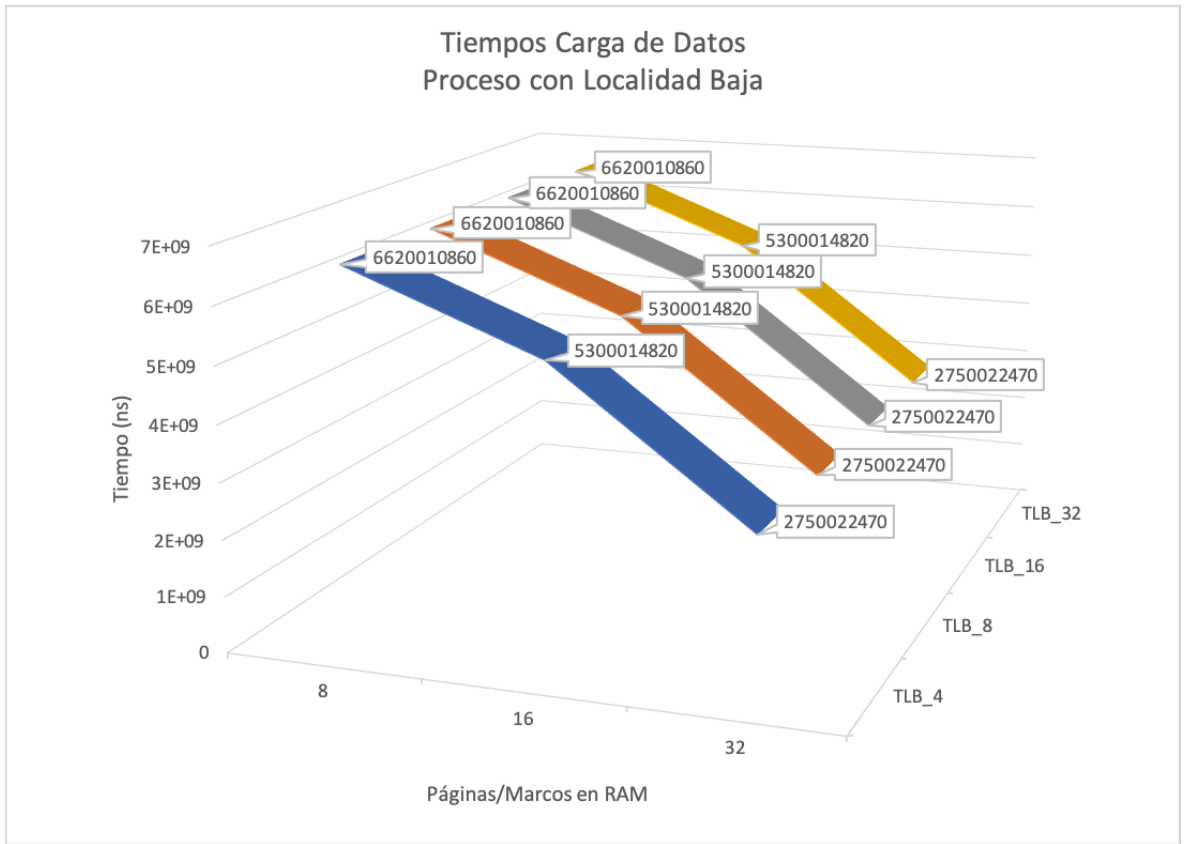
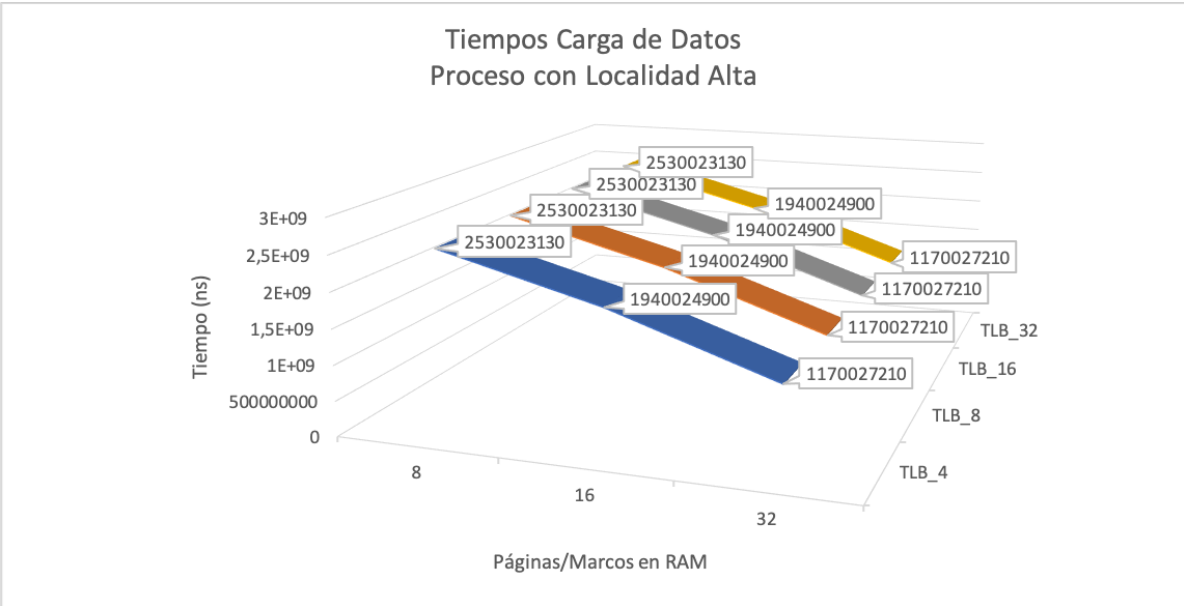


Figura 1. Variación en tiempo total para cargar los datos de un proceso.

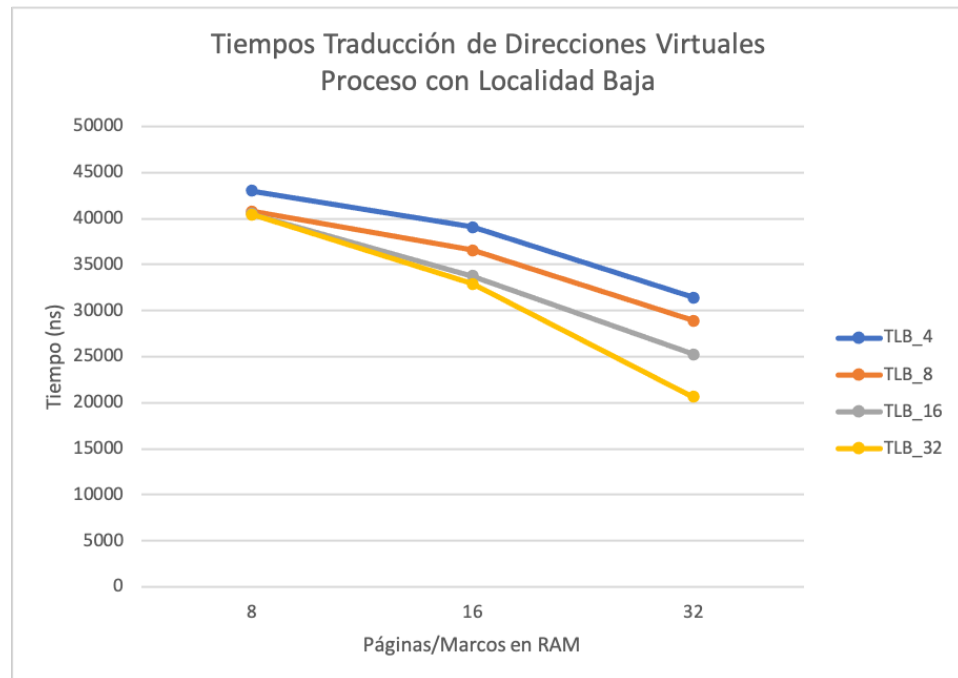
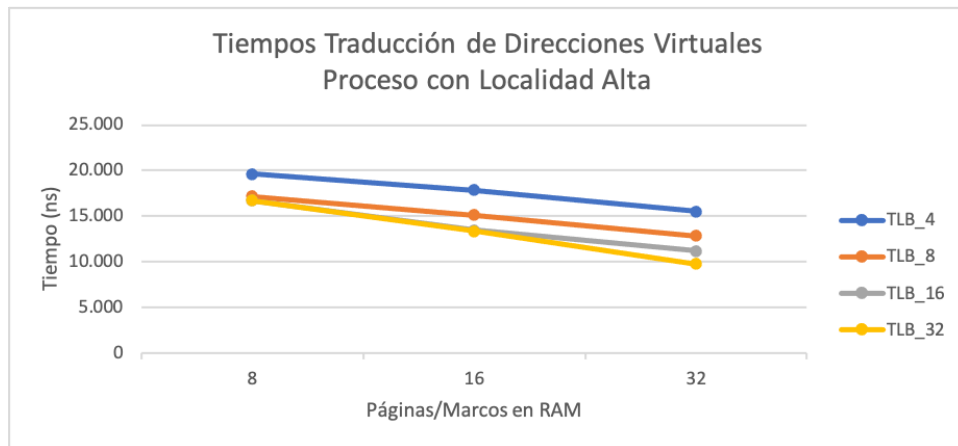


Figura 2. Variación en tiempo total para traducir direcciones virtuales de un proceso.

- Con este enunciado se entregarán dos archivos de referencias que corresponden a las gráficas. Uno tiene localidad ALTA y uno tiene localidad BAJA. Cada archivo tiene 1024 referencias. El objetivo es ilustrar el formato esperado y además puede revisar la ejecución de su programa.
- Por el efecto de la concurrencia, los números que su grupo obtenga pueden variar un poco (debería ser un máximo de +/- 5% de los valores mostrados).

Referencias:

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