

Proyecto de Sistemas Operativos

Objetivo del proyecto

Comprender las características de los distintos algoritmos de paginación y su impacto en el rendimiento de un sistema operativo doméstico.

Descripción del Proyecto

Implemente una simulación que contemple las siguientes características:

- Características de la computadora simulada
 - **Núcleos de procesamiento:** 1
 - **Instrucciones por segundo:** 1
 - **Tiempo de acceso a disco:** 5s
 - **Memoria RAM:** 400KB
 - **Disco Duro:** Indefinido (Asuma que nunca se acabará el espacio en disco duro)
 - **Tamaño de página:** 4KB
- Características de un proceso
 - El proceso será modelado en función de sus accesos a memoria
 - Cada proceso estará compuesto de dos estructuras:
 - Total de Memoria - Una lista de punteros que tenga la siguiente información, esta lista de punteros deberá estar dada por un archivo de texto separado por comas, el formato se explicará más adelante.
 - Identificador del puntero (Un número serial único para toda la simulación)
 - Tamaño (en B)
 - Memoria Asignada - Una lista de punteros que tenga la siguiente información
 - Identificador del puntero

- Dirección de memoria
 - Lista de Accesos - Una secuencia de accesos a memoria
 - Identificador del puntero
- Asuma que a cada puntero se le debe asignar al menos una página de memoria. Considere que el MMU debe llevar un registro de cada dirección de memoria lógica a cuál o cuáles páginas está asociada.
- Cuando un proceso completa su último acceso a memoria, deberá esperar 10s y luego liberar agradadamente toda su memoria asignada.
- Algoritmo
 - Su sistema deberá ser capaz de calcular el algoritmo óptimo de paginación (Página más tardía)
 - Adicionalmente deberá implementar los siguientes algoritmos de paginación:
 - LRU
 - Second Chance
 - Aging
 - Random
 - Al iniciar la simulación deberá mostrar una interfaz en la que el usuario introduzca los siguientes valores
 - Archivo con lista de accesos
 - Semilla para random
 - Algoritmo a simular
 - Una vez seleccionadas las opciones el usuario podrá iniciar la simulación en la cual su sistema deberá:
 - Barajar la lista de llamadas de memoria, de forma que el sistema intercale entre distintos procesos, pero que cada proceso mantenga su orden interno de ejecución. Use la semilla de random para esto, de forma que pueda replicarse el escenario.
 - Visualizar los siguientes elementos en simultáneo para el algoritmo óptimo y el seleccionado por el usuario:
 - Visualizar el estado de la RAM
 - Visualizar el estado de la MMU
 - Listar la cantidad de procesos corriendo
 - Mostrar el tiempo total de simulación hasta el momento
 - Mostrar la cantidad de RAM utilizada en KB y en %

- Mostrar la cantidad de V-RAM utilizada en KB y en %RAM (por ejemplo, dado que hay 400KB de RAM, si hay 500 KB en V-RAM deberá mostrar 125%)
- Mostrar la cantidad de tiempo que se ha gastado en fallos de paginación (Thrashing) y el porcentaje respecto al tiempo total de simulación. Si el Thrashing supera el 50% deberá marcarlo en rojo.
- Mostrar la cantidad de memoria que se ha desperdiciado por fragmentación interna.

Forma de Entrega

Podrá programar su proyecto en cualquier plataforma o lenguaje siempre y cuando corra en un entorno Linux. Considere que se revisará en un Fedora. Es indispensable que su proyecto contemple una visualización de la simulación, la cual se ilustra en el Anexo 2.

Deberá entregar el código fuente de su proyecto o un enlace con la simulación funcionando. Si su código fuente no compila o no corre, su asignación será anulada automáticamente. Deberá acompañar también su entrega con un informe que tenga las siguientes secciones:

- Portada
- Introducción (10%)
 - Sobre la gestión de memoria en los SO
 - Sobre qué es paginación y memoria virtual
- Desarrollo (60%, 15% cada uno)
 - Descripción del algoritmo óptimo y descripción de la implementación en el proyecto, ilustre con capturas de pantalla.
 - Descripción de los algoritmos FIFO y Second Chance, y descripción de la implementación en el proyecto, ilustre con capturas de pantalla.
 - Descripción de los algoritmos LRU y MRU, y descripción de la implementación en el proyecto, ilustre con capturas de pantalla.
 - Descripción de los algoritmos LFU y Aging, y descripción de la implementación en el proyecto, ilustre con capturas de pantalla.
- Enlace al código e instrucciones de compilación y ejecución
- Conclusiones (30%)
 - Al menos 3 conclusiones concretas
- Referencias bibliográficas en el formato de la plantilla provista

- Si omite alguna cita o referencia bibliográfica, o si comete errores en el formato de las referencias, su entrega será anulada automáticamente.

Respecto a su sistema, se evaluará de la siguiente manera:

- Implementación del Algoritmo Óptimo (10%)
 - Debe ser posible visualizar el algoritmo para evaluarlo
- Implementación de los Algoritmos (60%, 15% cada uno)
 - Debe ser posible visualizar los algoritmos para evaluarlos
- Diseño de la interfaz y visualización (20%)
 - Que la interfaz represente toda la información solicitada
 - Que la interfaz se pueda operar de forma intuitiva
- Repetibilidad del escenario (10%)
 - Que sea posible repetir las condiciones de una simulación utilizando la misma semilla

Se diseñará un archivo de pruebas con un listado de procesos que sigue el formato requerido en este documento. Si su simulación no corre con el archivo de pruebas no se evaluará. Note que el formato es de un archivo separado por comas, tal como se ilustra en el anexo 1.

Puede desarrollar este proyecto en grupo de máximo 4 personas.

Anexos

1. Formato del archivo de procesos:

1	PID, Ptr, Size
2	1, 001, 500
3	1, 002, 1024
4	1, 003, 512
5	2, 004, 256
6	2, 005, 512
7	3, 006, 128
8	3, 007, 1024
9	3, 008, 512
10	3, 009, 512
11	4, 010, 256

2. Ejemplo de la Visualización

