

INTRODUCCIÓN A LOS SISTEMAS OPERATIVOS: TEMA 3 - MEMORIA

Thrashing (Hiperpaginación)

- Concepto: decimos que un sistema está en thrashing cuando pasa más tiempo paginando (atendiendo page faults) que ejecutando procesos.
- Como consecuencia, el sistema sufre una baja significativa de performance.

Ciclo de Thrashing

1. El kernel monitorea el uso de la CPU.
 2. Si hay baja utilización → aumenta el grado de multiprogramación (la misma debe tener un límite).
 3. Si el algoritmo de reemplazo es global, pueden sacarse frames de otros procesos.
 4. Un proceso necesita más frames. Comienzan los page-faults y robo de frames a otros procesos.
 5. Por swapping de páginas y encolamiento en dispositivos, baja el uso de la cpu.
 6. Vuelve a 1
- ➔ Una de las soluciones cuando entro en hiperpaginación (o la CPU se malgasta en administración de memoria) consiste en bajar el grado de multiprogramación.

Control de thrashing: soluciones propuestas

- Se puede limitar el thrashing usando algoritmos de reemplazo local. (no es solución super efectiva).
- Con este algoritmo, si un proceso entra en hiperpaginación no roba frames a otros procesos.
- Si bien perjudica la performance del sistema, es controlable.

Conclusión sobre la hiperpaginación

- Si un proceso cuenta con todos los frames que necesita, no habría thrashing (esto es, que el SO sepa siempre que frames necesita un proceso en el momento adecuado).
- Una manera de abordar esta problemática es utilizando la estrategia de Working Set, la cual se apoya en el modelo de localidad.
- Otra estrategia para afrontar el thrashing con el mismo espíritu es el algoritmo PFF (Frecuencia de Fallos de Pagina).

El modelo de localidad:

- Cercanía de referencias o principio de referencias.

INTRODUCCIÓN A LOS SISTEMAS OPERATIVOS: TEMA 3 - MEMORIA

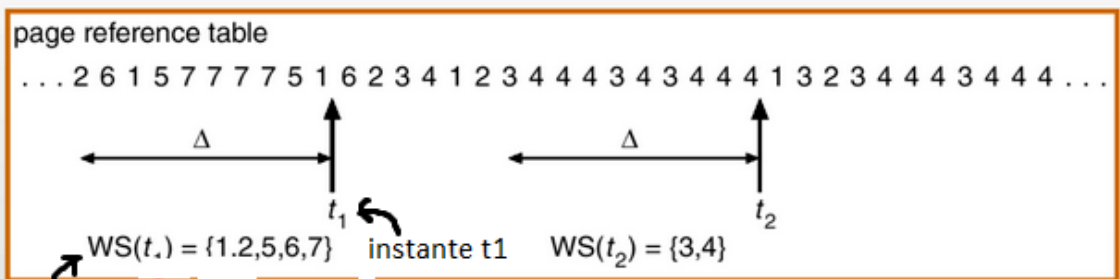
- Las referencias a datos y programas dentro de un proceso tienden a agruparse.
- La localidad de un proceso en un momento dado se da por el conjunto de páginas que tiene en memoria en ese momento.
- En cortos períodos de tiempo, el proceso necesitará pocas “piezas” del proceso (ej, una página de instrucciones y otra de datos).
- En la vida del proceso, el mismo va ejecutándose en diferentes localidades, cada localidad con su conjunto de instrucciones próximas entre sí. Cuando ocurre el cambio de localidad el proceso requiere más frames para seguir sus tareas.
- Continuando:
 - Un programa se compone de varias localidades.
 - Ej: cada rutina será una nueva localidad: se referencian sus direcciones (cercanas) cuando se está ejecutando.
 - Para prevenir la hiperactividad, un proceso debe tener en memoria (conjunto residente) sus páginas más activas (esto implica menos page faults).

El modelo de working set:

- Basado en el modelo de localidad definido anteriormente.
- Ventana del working set (Δ): las referencias de memoria más recientes.
- Working set: consiste en el conjunto de páginas que tienen las más recientes Δ que referencian a páginas.

✓ $\Delta = 10$

Si en T_1 debo elegir una página víctima, y tengo mi conj. residente y mi working set. Voy a elegir aquella página que no se encuentre en el working set.



- WorkingSet en el instante t_1
- Si el valor de Δ es MUY chico: lo que puede suceder es que el WS que arme no me cubra la localidad real del proceso. Chica no me sirve.
- Si el valor de Δ es MUY grande: puede que me quede en mi WS páginas que ya no tenga que usar.
- Surge el problema, que valor correcto darle a Δ (es solucionable).

INTRODUCCIÓN A LOS SISTEMAS OPERATIVOS: TEMA 3 - MEMORIA

- El problema mayor, ¿Cómo es que el sistema operativo tiene registro de las páginas que va usando?, se debería apoyar en el hardware
- El SO puede (como solución), ver la tabla de paginas de cada proceso y analizar cada bit de Referencia (si está en 1 implica que la página se estuvo usando). Entonces, los bits de R que están en 1 son puestos a 0. En la próxima vez que el SO aparezca a ver la tabla de páginas, va a ver cuales páginas tienen de nuevo el bit en 1 para empezar a analizar y comparar con la vez anterior que miró la tabla. Como para determinar cuál página está siendo usada frecuentemente, (resulta en una determinación que se aproxima). El problema es que ésta opción resulta costosa, por lo que el modelo de Working Set implica mucho CPU.
- En resumen, la técnica pinta buena, pero es costosa de implementar (consumo de CPU).
- Medida del working set:
 - Si la demanda total de frames es mayor a la cantidad de frames disponibles. Entonces hay THRASHING.

- ✓ m = cantidad frames disponibles
- ✓ WSS_i = medida del working set del proceso p_i .
- ✓ $\sum WSS_i = D$;
- ✓ D = demanda total de frames.
- ✓ Si $D > m$, habrá **thrashing**.

-
- Para achicar la demanda, debo bajar el grado de multiprogramación.

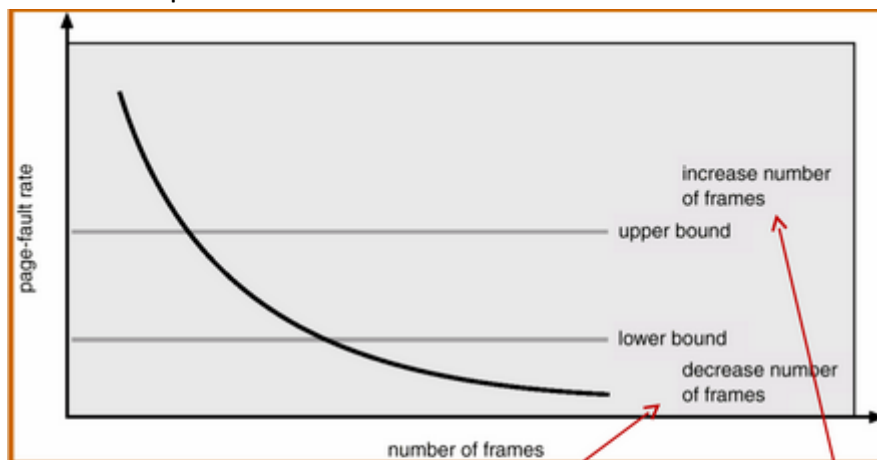
Prevención del thrashing:

- SO monitoreando cada proceso, dándole tantos frames hasta su WSS (medida del working set del proceso P_i).
 - Si quedan frames, puede iniciar otro proceso.
 - Si D crece, excediendo m , se elige un proceso para suspender, reasignándose sus frames.
 - Así, se mantiene alto el grado de multiprogramación optimizando el uso de la CPU.
-

INTRODUCCIÓN A LOS SISTEMAS OPERATIVOS: TEMA 3 - MEMORIA

Prevención del thrashing por PFF:

- La técnica PFF (Page Fault Frequency o Frecuencia de Fallo de Páginas), en lugar de calcular el Working Set de los procesos, utiliza la tasa de fallos de pagina para estimar si el proceso tiene un conjunto residente que representa adecuadamente al WS.
- Hace que el Page Fault quede en un valor “manejable”
- PFF: Frecuencia de page faults.
- PFF alta → se necesitan más frames para el proceso.
- PFF baja → los procesos tienen muchos frames asignados (el proceso tiene frames de sobra).
- Si o sí reemplazo local.



✓ Establecer tasa de PF aceptable

- ✓ Si la tasa actual es baja, el proceso pierde frames
- ✓ Si la tasa actual es alta, el proceso gana frames.

- Continuando:
 - PFF es la técnica usada por los sistemas operativos hoy en día.
 - Se deben establecer límites superior e inferior de las PFF's deseadas.
 - Si se excede PFF máx → se le asigna un frame más al proceso, ya que el mismo genera muchos page fault y probablemente lo esté necesitando.
 - Si la PFF está por debajo del mínimo → se le quita un frame al proceso, asumiendo que el mismo tiene frames de más.
 - Puede llegar a ocurrir que se suspenda un proceso si no hay más frames disponibles. Sus frames se reasignan a procesos de alta PFF.