

## 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

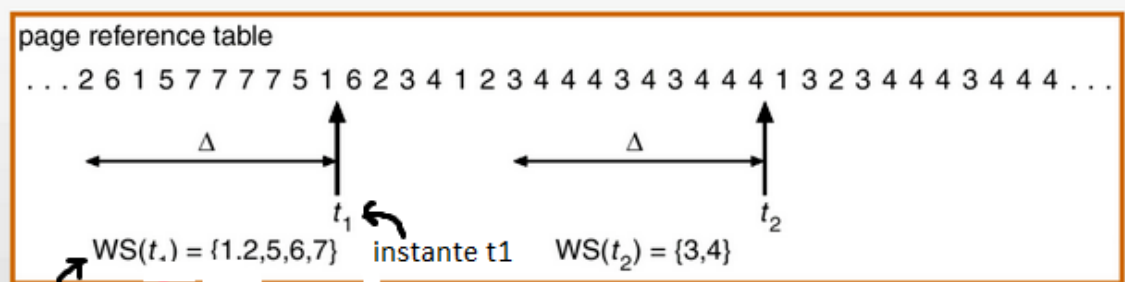
- Las referencias a datos y programas dentro de un proceso tienden a agruparse.
- La localidad de un proceso en un momento dado se da por el conjunto de páginas que tiene en memoria en ese momento.
- En cortos períodos de tiempo, el proceso necesitará pocas “piezas” del proceso (ej, una página de instrucciones 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 ( $\Delta$ ): las referencias de memoria más recientes.
- Working set: consiste en el conjunto de páginas que tienen las más recientes  $\Delta$  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  $\Delta$  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  $\Delta$  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  $\Delta$  (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.

---

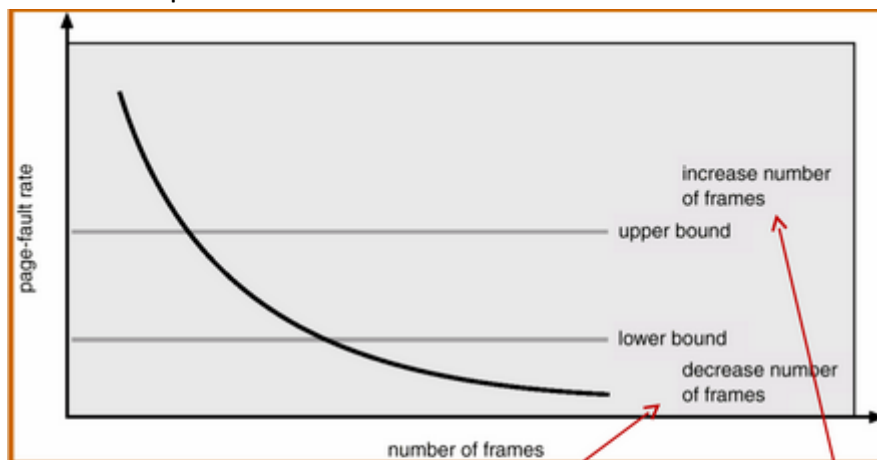
### Prevención del thrashing:

- 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.

## INTRODUCCIÓN A LOS SISTEMAS OPERATIVOS: TEMA 3 - MEMORIA

### Demonio de paginación

- Proceso creado por el SO durante el arranque que apoya a la administración de la memoria.
- Es ejecutado cuando el sistema tiene una baja utilización o algún parámetro de la memoria lo indica.
  - Poca memoria libre.
  - Mucha memoria modificada.
- Tareas del demonio
  - Limpiar paginas modificadas sincronizándolas con el swap.
    - Esto reduce el tiempo de swap posterior ya que las paginas están limpias.
    - Reduce también el tiempo de transferencia al sincronizar varias paginas contiguas.
  - Mantiene un cierto numero de frames libres en el sistema, viendo que procesos tienen un PFF alto o bajo.
  - Demora la liberación de una página hasta que la misma haga falta realmente.
- En Windows → proceso system.
- En Linux → kswapd

### Memoria compartida

- Gracias al uso de la tabla de páginas, varios procesos pueden compartir un marco de memoria; para ello ese frame debe estar asociado a una página en la tabla de páginas de cada proceso.
- El número de página asociado al marco puede ser diferente en cada proceso.
- **Codigo compartido**
  - Los procesos comparten una copia de código (read only), por ej: editores de texto, compiladores, etc.
  - Los datos son privados a cada proceso y se encuentran en paginas no compartidas.
- El frame de memoria que es compartido, físicamente está en un solo lugar.

### Mapeo de archivo en memoria

- ➔ Es una técnica que permite a un proceso asociar el CONTENIDO de un archivo a una región del propio espacio de direcciones virtuales.
- ➔ El contenido del archivo no es subido a RAM hasta que se generan Page Faults.

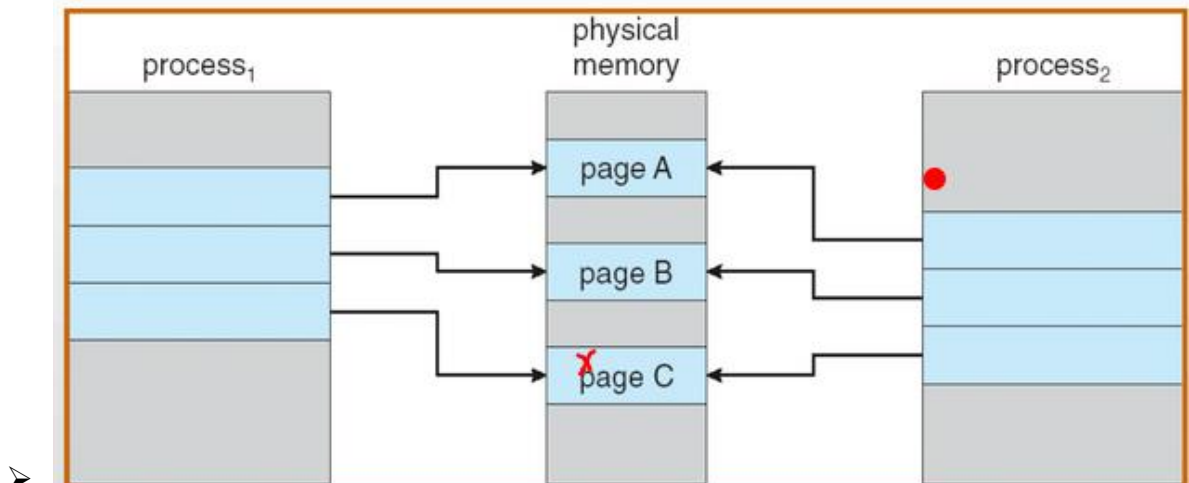
## INTRODUCCIÓN A LOS SISTEMAS OPERATIVOS: TEMA 3 - MEMORIA

- ➔ Básicamente es alinear el contenido de un archivo en un propio espacio de direcciones, por lo que si se accede a la porción que posee la alineación con dicho archivo, es como acceder al archivo.
- ➔ Continuando...
  - El contenido de la pagina que genera el PF es obtenido desde el archivo asociado (no del área de SWAP)
  - Cuando el proceso termina o el archivo es liberado, las paginas modificadas son escritas en el archivo correspondiente.
  - Permite la realización de E/S de una manera alternativa (sin usar funciones del SO como read o write) a usar operaciones directas sobre el sistema de archivos.
  - Es utilizado comúnmente para asociar librerías compartidas o DLLs.

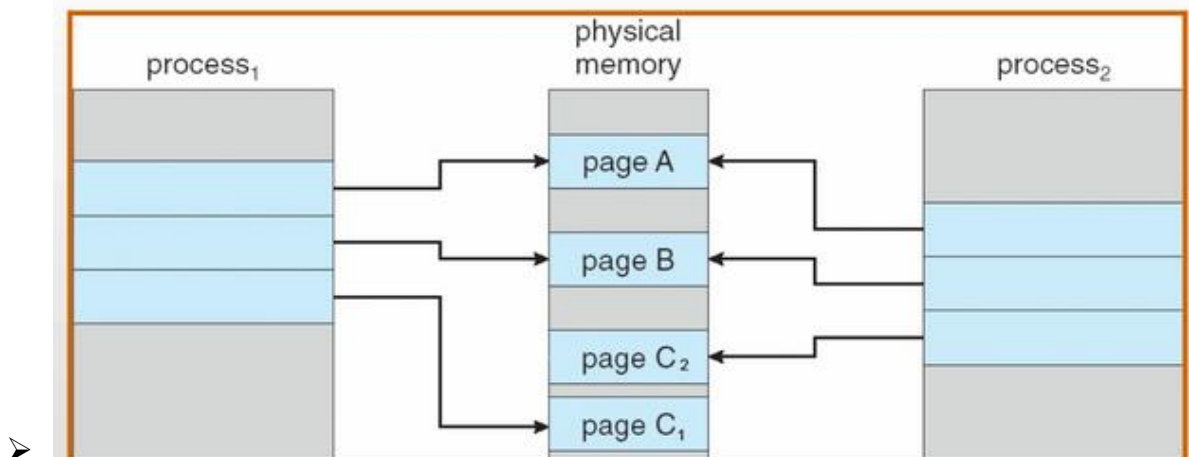
### Copia en Escritura

- La copia en escritura (Copy-on-Write, COW), permite a los procesos padre e hijo compartir inicialmente las mismas páginas de memoria (apuntan a los mismos frames).
  - Si uno de ellos modifica una pagina que comparten, entonces la página es copiada.
  - Usada mucho en Linux.
- COW permite la creación de procesos de forma más eficientemente debido a que sólo las páginas modificadas son duplicadas.
- Las páginas de datos son de SOLO lectura, (ya que tanto hijo como padre deberían tener una técnica de trabajo que les permita manejar datos distintos). Entonces al escribir una página de datos se aplica el COPY-on-WRITE.
- La pagina copiada va a partirse en dos, y tanto padre va a apuntar a copia 1 y el hijo a copia2.

### El Proceso 1 Modifica la Página C (Antes)



### El Proceso 1 Modifica la Página C (Después)



### Area de Intercambio (swap)

- En los diferentes SO
  - Linux: área dedicada, separada del sistema de archivos.
  - Windows: un archivo dentro del sistema de archivos.
- Técnicas para la administración de la SWAP:
  - Cada vez que se crea un proceso es reservada una zona del área de intercambio que tiene el mismo tamaño del proceso. A cada proceso se le es asignada la dirección en disco de su área de swap correspondiente. La lectura de dicha área se realiza sumando: número de página virtual + dirección de comienzo del área de swap asignada al proceso, el problema es que **desperdicio espacio**.
  - No se asigna nada inicialmente. A cada pagina se le asigna su espacio en disco cuando se va a intercambiar, y el espacio es liberado cuando la pagina vuelve a memoria. Problema: se

## INTRODUCCIÓN A LOS SISTEMAS OPERATIVOS: TEMA 3 - MEMORIA

debe llevar contabilidad en memoria (página a página) de la localización de las páginas en disco, es la **técnica más usada ya que optimiza el espacio usado en área de swap**.

- Continuando:
  - Cuando una página no está en memoria, sino en disco, como se puede saber en que parte del área de swap está?
    - Respuesta: el PTE de dicha pagina tiene el bit V=0 y todos los demás bits sin usar (toda la información guardada no tiene sentido).