

RELACIÓN TEMA 3:

1. Cuestiones generales sobre memoria virtual:

a) ¿Cuáles son las diferencias entre utilizar un mecanismo de paginación sin memoria virtual y con memoria virtual?

En la paginación sin memoria virtual:

- Cada proceso dispone de su propia tabla de páginas y todas están en MP.
- Cada entrada a la tabla es: número de marco de la página correspondiente en memoria principal.

En la de memoria virtual:

- No todas las páginas están en MP.
- Las entradas a la TP son más complejas, tienen un bit de presencia en MP (si está, otro que indique el número de marco de la página). Además tiene un bit de modificado, que indica si los contenidos se han alterado desde que la página se cargó en memoria.

b) ¿Qué campos de información se incluyen necesariamente en cada entrada de la tabla de páginas de un sistema de memoria virtual con paginación (memoria virtual paginada)? Explique para que sirve cada uno de ellos.

- Número de página (en la dirección virtual).
- Dirección base del marco, dónde está almacenado el marco de página.
- Protección, el modo de acceso a la página.
- Bit de validez/presencia.
- Bit de modificación.

c) ¿Para qué sirve el TLB de la MMU?

Es un cache de alta velocidad para las entradas a la tabla de página, contiene aquellas entradas que han sido usadas más recientemente.

Dada una dirección virtual, el procesador primero examina la TLB, si TLB hit, recupera el número de marco, si TLB miss, indexa la tabla de páginas.

Agiliza el mecanismo de traducción, pues no es necesario realizar tantos accesos a memoria (normalmente hay dos: uno para buscar la entrada de páginas y otro para buscar los datos).

d) ¿Qué es el buffering de páginas y para qué sirve?

El buffering de páginas es una política de reemplazo, utiliza un algoritmo de reemplazo FIFO sencillo, aunque se añaden dos estructuras a las que se incorpora una página cuando es reemplazada:

- Lista de páginas libres

- Lista de páginas modificadas

La lista de páginas libres es una lista de marcos de páginas disponibles para lectura de nuevas páginas. Cuando una página se va a leer, se utiliza el marco de página en la cabeza de esa lista.

Lo más importante es que la página que se reemplaza se mantiene en la memoria, de esta forma, las referencias se devuelven con un bajo coste.

Además, las páginas modificadas se escriben en grupos, es decir que se reduce el número de operaciones de E/S, y por tanto de tiempo de acceso a disco.

e) Para la gestión de memoria en un sistema de memoria virtual con paginación, ¿qué estructura/s de datos necesitará mantener el Sistema Operativo para administrar el espacio asignado a los procesos y el espacio libre?

Mantiene:

- Tabla de Páginas, que tiene información necesaria para la traducción.
- Tabla de Ubicación en Disco, mantiene la ubicación de cada página en almacenamiento auxiliar.
- Tabla de Marcos de Página, que mantiene información relativa a cada marco de página en el que se divide la MP.

Ejercicio 2, pereza copiar enunciado

36 bits en la virtual

29 bits en la física

Ejercicio 3: considere un sistema en el que existe un HDD para el sistema de archivos y otro HDD para el espacio de intercambio de la memoria virtual. Analice y responda las preguntas para cada uno de los siguientes escenarios:

a) Escenario en el cual la tasa de utilización de la CPU es del 15% y el dispositivo de paginación tiene una tasa de utilización del 97%. ¿Qué indican estas medidas de prestaciones del sistema?

Este sistema tiene un problema de trashing, pasa la mayor parte del tiempo traduciendo direcciones porque las referencias a memoria fallan. Es decir tiene una tasa de fallos de página elevada.

b) Escenario en el cual la tasa de utilización de la CPU sigue siendo del 15% pero la tasa de utilización del dispositivo de paginación es del 20%. ¿Qué indican estas medidas de prestaciones del sistema?

El sistema tarda mucho tiempo en realizar la transferencia de página, es decir que no tarda tiempo ni traduciendo ni ejecutando, gasta tiempo al recuperar las páginas.

Ejercicio 6: conjunto residente y conjunto de trabajo de un proceso en un sistema de memoria virtual paginada:

a) ¿El tamaño del conjunto residente de un proceso depende directamente del tamaño del programa ejecutable asociado?

No depende del tamaño del ejecutable, se cargan en memoria las páginas según el algoritmo de selección, y van cambiando a medida que ejecuta.

b) ¿El tamaño del conjunto de trabajo de un proceso depende directamente del tamaño del programa ejecutable asociado?

No depende, pues el tamaño del conjunto de trabajo es el conjunto de páginas referenciadas en un intervalo de tiempo. No hace falta referenciar todas las páginas.

c) El tamaño del conjunto residente y el tamaño del conjunto de trabajo de un proceso...¿es siempre el mismo durante toda la “vida” del proceso? Caso de no ser así, explique que escenarios provocan que el tamaño de dichos conjuntos varíe uno con respecto al otro.

Ambos conjuntos van cambiando a medida que avanza la ejecución.

El conjunto residente es el conjunto de páginas que tiene en memoria un proceso en un instante determinado.

Se pueden cambiar con una sustitución o reemplazo de una página, es decir, cuando se trae a memoria principal una página nueva. Esta situación se da después de un fallo de página.

Si no se da un fallo de página, el programa mantiene sus conjuntos de trabajo y residente.

Ejercicio 7: ¿Por qué los registros del TLB de la MMU (la cual se direcciona mediante direcciones virtuales) pueden producir incoherencias y requieren que el sistema operativo los invalide(flush de TLBs) en cada cambio de contexto y, en cambio, una memoria direccionada mediante direcciones físicas no requiere tal “limpieza” del TLB

Las direcciones lógicas son relativas a cada proceso, por lo que no se corresponden con la misma dirección en otro proceso. Sin embargo, las direcciones físicas son inequívocas, se refieren a una posición de memoria.

Ejercicio 9: Analice qué puede ocurrir en un sistema que usa paginación por demanda si se recompila un programa mientras se está ejecutando. Proponga soluciones a los problemas que pueden surgir en dicha situación.

Si se recompila el programa, las páginas cargadas en memoria continuarán ejecutando, pero cuando se realice un reemplazo de las páginas se llevará una página nueva recompilada, que no corresponde con las direcciones lógicas anteriores. Fallará la ejecución.

Si las páginas no cargadas están en la zona de intercambio, no falla pues al recompilar no se actualizan.

Ejercicio 10: para cada uno de los siguientes campos de la tabla de páginas explique si es la MMU o el kernel del SO quién los lee y escribe (en caso de escritura indique si el bit se activa (1) o desactiva (0)), y en qué momentos:

- NÚMERO DE MARCO: Lo lee la MMU para hacer la traducción. Lo escribe el SO al asignar memoria física a una página.
- BIT DE PRESENCIA: El SO lo escribe cuando asigna la memoria, pone a 1. El MMU lo lee para comprobar el acceso.
- BIT DE PROTECCIÓN: El MMU lo consulta, para ver si se tiene permiso para operar con el marco de página. El SO lo escribe.
- BIT DE MODIFICACIÓN: El MMU lo activa cuando detecta un acceso, el SO lo lee para guardar la página, y lo desactiva cuando se acaba de escribir en memoria secundaria
- BIT DE REFERENCIA: Cuando se accede a la página, el MMU activa. El SO consulta y desactiva según el algoritmo.

Ejercicio 11: ¿Cuánto puede avanzar como máximo la “aguja” del algoritmo de reemplazo de páginas del reloj durante la selección de una página?

Avanza, si no se encuentra ninguna página con el bit de usado a 0, completa un ciclo entero del buffer de páginas (va poniendo a 0).

Ejercicio 12:

- 999)d : pagina 0 desplazamiento 999: marco de página 4
- 2121)d : pag 2, desplazamiento 73: marco 1
DF: $1 \cdot 1024 + 73$
- 5400)d: pag 5, despl 280: marco 0

Ejercicio 13:

- 3 marcos: 9 faltas
 - 4 marcos: 10 faltas
- Se dan las mismas faltas, pues se traen constantemente de vuelta páginas que se han retirado.

Ejercicio 15: Indique las ventajas y desventajas del algoritmo de frecuencia de faltas de página con respecto al algoritmo basado en el conjunto de trabajo con un tamaño de ventana τ y un parámetro τ .

- Si se producen muchas faltas de página, se realizan comprobaciones de más, lo que resulta en una sobrecarga hardware innecesaria, pues hay muchos fallos seguidos y no superarán el intervalo.
- Si hay pocos fallos de página, se pueden tener páginas que no se han referenciado desde hace mucho en memoria, y por tanto están ocupando memoria.

Ejercicio 19:

En el modelo del conjunto de trabajo, todas las páginas que no han sido referenciadas en una ventana de tiempo son eliminadas. Para añadir este mecanismo hardware, podemos proceder de la siguiente manera:

- Cuando se accede a una página el bit se activa. Tenemos que desactivar el bit de página cuando pasa la ventana de tiempo, y en cada referencia eliminar los conjuntos que estén en memoria con el bit activo.
- Es decir, cada x referencias la función `clean` borre las páginas cuyo bit U esté a 0.

Ejercicio 20: mecanismo lock/unlock que protege marcos de página

- a) Para realizar estos mecanismos necesitamos:
 - TABLA DE MARCOS DE PÁGINA, almacena información sobre la protección.
- b) Pueden ser útiles cuando varios usuarios están compartiendo una página, o se puede usar como un buffer de E/S que debe estar bloqueado durante la transferencia (no se referencia en mucho tiempo y se podría eliminar).
- c) Se puede dar la hiperpaginación, pues si se bloquean demasiadas páginas se estará constantemente realizando trabajo de swapping. Para solucionar este problema se puede poner un límite de páginas bloqueadas.

Ejercicio 24: Gestión de memoria en Linux

- a) Se copia to ESTUDIAR DIBUJO FINAL DE DIAPOS

FINNN