III PAC

UNIVERSIDAD NACIONAL AUTÓNOMA DE HONDURAS

Facultad de Ingeniería
Departamento de Ingeniería en Sistemas

INVESTIGACIÓN
NO.5

Sistemas Operativos II - 0800
Ing. Norberto Mendoza

Ana Evelin Hernández Martínez
20171001620

Tegucigalpa M.D.C.
Ciudad Universitaria (UNAH)

7 de Octubre de 2020

Investigación No.5

Caracteristicas de la Paginación

Debido a que un proceso se ejecuta sólo en la memoria principal, esta memoria se denomina ***memoria real***. Pero el programador o el usuario persiben una memoria potencialmente mucho más grande la cual se encuentra localizada en el disco. Esta última se denomina ***memoria virtual***. La memoria virtual permite una multiprogramación muy efectiva que libera al usuario de las restricciones excesivamente fuertes de la memoria principal.

Caracteristicas de la Paginación con o sin el uso de la memoria virtual:*

- *Paginación Sencilla:*
    - Memoria principal particionada en fragmentos pequeños de un tamaño fijo llamados marcos.
    - Programa dividido en páginas por el compilador del sistema de gestion de memoria.
    - Fragmentación interna dentro de los marcos.
    - Sin fragmentación externa.
    - El sistema operativo debe mantener una tabla de paginación por cada proceso mostrado en el marco que se encuentra cada página ocupada.
    - El sistema operativo debe mantener una lista de marcos libres.
    - El procesador usa el número de páginas, desplazamineto para calcular direcciones absolutas.
    - Todas las páginas del proceso deben encontrarse en la memoria principal para que el proceso se pueda ejecutar, salvo que se utilicen solapamientos (*overlays*).
- *Paginación con Memoria Virtual:*
    - Memoria principal particionada en fragmentos pequeños de un tamaño fijo llamados marcos.
    - Programa dividido en páginas por el compilador o el sistema de gestión de memoria.
    - Fragmentación interna dentro de los marcos.
    - Sin fragmentación externa.
    - El sistema operativo debe mantener una tabla de páginas por cada proceso mostrado en el marco que se encuentra cada página ocupada.
    - El sistema operativo debe mantener una lista de marcos libres.
    - El procesador usa el número de páginas, desplazamineto para calcular direcciones absolutas.
    - No se necesita mantener todas las páginas de los procesos en los marcos de la memoria principal para que el proceso se ejecute. Las páginas se pueden leer bajo demanda.
    - La lectuara de una página a memoria puede requerir la escritura de una página a disco.

Entradas de las Tablas de Páginas

Mientras hablabamos sobre paginación sencilla y paginación con memoria virtual uno de los aspectos a destacar es que cada proceso dispone de su propia tabla de páginas y que todas las páginas se encuentran localizadas en la memoria principal. Cada entrada en la tabla de páginas corresponde a un número de marco de la correspondiente página en la memoria principal. Para la memoria virtual basada en el esquema de paginación tambien se necesita una tabla de páginas. Normalmente se asocia una única tabla de páginas a cada proceso. En este caso, sin embargo, las entradas de la tabla de páginas son más compleja. Devido a

que sólo algunas de las páginas de proceso se encuentra en la memoria principal, se necesita que cada entrada en la tabla indique *si la correspondiente página está presente (P) en memoria principal o no*. Si el bit indica que la página está en memoria, la entrada tambien debe indicar el número de marco de dicha página.

La entrada de la tabla de páginas incluye *el bit de modificado (M)*, que indica si los contenidos de la correspondiente página han sido alterados desde que la página se cargó por primera vez en la memoria principal. Si no había ningún cambio, no es necesario escribir la página cuando llegue el momento de reemplazarla por otra página en el marco de página que actualmente ocupa. Pueden existir también otros bits de control en estas entradas. Por ejemplo, si la protección y compartición se gestionan a nivel de página se necesitan también los bits para este propósito.

Estructura de la tabla de páginas. El mecanismo básico de lectura de una palabra de la memoriaimplica la traducción de la dirección virtual, o lógica, consistente en un número de página y un desplazamiento, a la dirección física, consistente en un número de marco y un desplazamiento, usando para ello la tabla de páginas. Debido a que la tabla de páginas es de longitud variable dependiendo del tamaño del proceso, no podemos suponer que se encuentra almacenada en los registros. En lugarde eso, debe encontrarse en la memoria principal para poder ser accedida.

La entrada en la tabla de páginas incluye la siguiente información:

- **Número de página.** Esta es la parte correspondiente al número de página de la direcciónvirtual.
- **Identificador del proceso.** El proceso que es propietario de está página. La combinación denúmero de página e identificador del proceso identifica a una página dentro del espacio de direcciones virtuales de un proceso en particular.
- **Bits de control.** Este campo incluye los flags, como por ejemplo, válido, referenciado, y modificado; e información de protección y cerrojos.
- **Puntero de la cadena.** Este campo es nulo (indicado posiblemente por un bit adicional) si no hay más entradas encadenadas en esta entrada. En otro caso, este campo contiene el valor del índice (número entre 0 y 2m-1) de la siguiente entrada de la cadena.

Buffer de Traducción Adelantada (TLB)

En principio, toda referencia a la memoria virtual puede causardos accesos a memoria física: uno para buscar la entrada la tabla de páginas apropiada y otro para buscar los datos solicitados. De esa forma, un esquema de memoria virtual básico causaría el efecto de duplicar el tiempo de acceso a la memoria. Para solventar este problema, la mayoría de esquemas de la memoria virtual utilizan una cache especial de alta velocidad para las entradas de la tabla de página, habitualmente denominada buffer de traducción anticipada (***translation lookaside buffer - TLB***).

Esta cache funciona de forma similar a una memoria cache general y contiene aquellas entradas de la tabla de páginas que han sido usadas de forma más reciente. Dada una dirección virtual, el procesador primero examina la TLB, si la entrada de la tabla de páginas solicitada está presente(acierto en TLB), entonces se recupera el número de marco y se construye la dirección real. Si la entrada de la tabla de páginas solicitada no se encuentra (fallo en la TLB), el procesador utiliza el número de página para indexar la tabla de páginas del proceso y examinar la correspondiente entrada de latabla de páginas. Si el bit de presente está puesto a 1, entonces la página se encuentra en memoria principal, y el procesador puede recuperar el número de marco desde la entrada de la tabla de páginas para construir la dirección real. El procesador también autorizará la TLB para incluir esta nueva entrada de tabla de páginas. Finalmente, si el bit presente no está puesto a 1, entonces la página solicitada no se encuentra en la memoria principal y se produce un fallo de

acceso memoria, llamado fallo de página. En este punto, abandonamos el dominio del hardware para invocar al sistema operativo, el cual cargará la página necesaria y actualizada de la tabla de páginas.

Hay numerosos detalles adicionales relativos a la organización real de la TLB. Debido a que la TLB sólo contiene algunas de las entradas de toda la tabla de páginas, no es posible indexar simplemente la TLB por medio de número página. En lugar de eso, cada entrada de la TLB debe incluir unnúmero de página así como la entrada de la tabla de páginas completa. El procesador proporciona un hardware que permite consultar simultáneamente varias entradas para determinar si hay una conciencia sobre un número de página. Esta técnica se denomina resolución asociativa (*asociative mapping*) que contrasta con la resolución directa, o indexación, utilizada para buscar en la tabla de páginas. El diseño de la TLB debe considerar también la forma mediante la cual las entradas se organizan en ella y qué entrada se debe reemplazar cuando se necesite traer una nueva entrada. Estos aspectos deben considerarse en el diseño de la cache hardware.

Para concluir, el mecanismo de memoria virtual debe interactuar con el sistema de cache (no la cachede TLB, sino la cache de la memoria principal). Una direcciónvirtual tendrá generalmente el formato número de página, desplazamiento. Primero, el sistema de me-moria consulta la TLB para ver si se encuentra presente una entrada de tabla de página que coincide.Si es así, la dirección real (física) se genera combinando el número de marco con el desplazamiento.Si no, la entrada se busca en la tabla de páginas. Una vez se ha generado la dirección real, que man-tiene el formato de etiqueta (tag) y resto (remainder), se consulta la cache para ver si el bloque quecontiene esa palabra se encuentra ahí. Si es así, se le devuelve a la CPU. Si no, la palabra se busca enla memoria principal.

Tamaño de Páginas

Una decisión de diseño hardware importante es el tamaño de página a usar. Hayvarios factores a considerar. Por un lado, está la fragmentación interna. Evidentemente, cuanto mayor esel tamaño de la página, menor cantidad de fragmentación interna. Para optimizar el uso de la memoriaprincipal, sería beneficioso reducir la fragmentación interna. Por otro lado, cuanto menor es la página,mayor número de páginas son necesarias para cada proceso. Un mayor número de páginas por proceso significa también mayores tablas de páginas. Para programas grandes en un entorno altamente multiprogramado, esto significa que determinadas partes de las tablas de página de los procesos activos debenencontrarse en la memoria virtual, no en la memoria principal. Por tanto, puede haber un fallo de páginadoble para una referencia sencilla a memoria: el primero para atraer la tabla de página de la parte solicitada y el segundo para atraer la página del propio proceso. Otro factor importante son las característicasfísicas de la mayoría de los dispositivos de la memoria secundaria, que son de tipo giratorio, favoreciendo tamaños de página grandes para mejorar la eficiencia de transferencia de bloques de datos.

Aumentando la complejidad de estos aspectos se encuentra el efecto que el tamaño de página tiene en relación a la posibilidad de que ocurra un fallo de página. Este comportamiento en términos generales, se basa en el principio de proximidad. Si el tamaño de página es muy pequeño, de forma habitual habrá un número relativamente alto de páginas disponibles en la memoria principal para cada proceso. Después de un tiempo, las páginas en memoria contendrán las partes de los procesos a las que se ha hecho referencia de forma reciente. De estaforma, la tasa de fallos de página debería ser baja. A medida que el tamaño de páginas se incrementa,la página en particular contendrá información más lejos de la última referencia realizada. Así pues, elefecto del principio de proximidad se debilita y la tasa de fallos de página comienza a crecer. En algún momento, sin embargo, la tasa de fallos de página comenzará a caer a medida que el

tamaño de la página se aproxima al tamaño del proceso completo (punto Pen el diagrama). Cuando una únicapágina contiene el proceso completo, no habrá fallos de página.

Para concluir, el aspecto de diseño del tamaño página se encuentra relacionado con el tamaño dela memoria física y el tamaño del programa. Al mismo tiempo que la memoria principal está siendocada vez más grande, el espacio de direcciones utilizado por las aplicaciones también crece. Esta tendencia resulta más evidente en ordenadores personales y estaciones de trabajo, donde las aplicacionestienen una complejidad creciente. Por contra, diversas técnicas de programación actuales usadas paraprogramas de gran tamaño tienden a reducir el efecto de la proximidad de referencias dentro un proceso. Por ejemplo,

- Las técnicas de programación orientada a objetos motivan el uso de muchos módulos de datos y programas de pequeño tamaño con referencias repartidas sobre un número relativamente alto de objetos en un periodo de tiempo bastante corto.
- Las aplicaciones multihilo (*multithreaded*) pueden presentar cambios abruptos en el flujo de instrucciones y referencias a la memoria fraccionadas.

Tabla con Ejemplos de tamaños de páginas:

Computer	**Tamaño de página**
Atlas	512 palabras de 48-bits
Honeywell-Multic	1024 palabras de 36-bits
IBM 370/XA y 370/ESA	4 Kbytes
Familia VAX	512 bytes
IBM AS/400	512 bytes
DEC Alpha	8 Kbytes
MIPS	4 Kbytes hasta 16 Mbytes
UltraSPARC	8 Kbytes hasta 4 Mbytes
Pentium	4 Kbytes o 4 Mbytes
PowerPc	4 Kbytes
Itanium	4 Kbytes hasta 256 Mbytes

Para un tamaño determinado de una TLB, a medida que el tamaño del proceso crece y la pro-ximidad de referencias decrece, el índice de aciertos en TLB se va reduciendo. Bajo estas circuns-tancias, la TLB se puede convertir en el cuello de botella del rendimiento.

Segmentación

La segmentación permite al programador ver la memoria como si se tratase de diferentes espacios de direcciones o segmentos. Los segmentos puedenser de tamaños diferentes, en realidad de tamaño dinámico. Una referencia a la memoria consiste enun formato de dirección del tipo (número de segmento, desplazamiento).

Esta organización tiene un gran número de ventajas para el programador sobre los espacios de direcciones no segmentados:

- Simplifica el tratamiento de estructuras de datos que pueden crecer.
- Permite programas que se modifican o recopilan de forma independiente, sin requerir que elconjunto completo de programas se re-enlacen y se vuelvan a cargar.
- Da soporte a la compartición entre procesos.
- Soporta los mecanismos de protección.

Tablas de Segmentos

Cuando todos estos segmentos se han cargado en la memoria principal, la tabla de segmentos del proceso se crea y se carga también en la memoria principal. Cada entrada de la tabla de segmentos contiene la dirección de comienzo del correspondiente segmento en la memoria principal, así como la longitud del mismo. El mismo mecanismo, una tabla segmentos, senecesita cuando se están tratando esquemas de memoria virtual basados en segmentación. De nuevo,lo habitual es que haya una única tabla de segmentos por cada uno de los procesos. En este caso sinembargo, las entradas en la tabla de segmentos son un poco más complejas. Debido aque sólo algunos de los segmentos del proceso pueden encontrarse en la memoria principal, se necesita un bit en cada entrada de la tabla de segmentos para indicar si el correspondiente segmento se encuentra presente en la memoria principal o no. Si indica que el segmento está en memoria, la entradatambién debe incluir la dirección de comienzo y la longitud del mismo.

Otro bit de control en la entrada de la tabla de segmentos es el bit de modificado, que indica si loscontenidos del segmento correspondiente se han modificado desde que se cargó por última vez en lamemoria principal. Si no hay ningún cambio, no es necesario escribir el segmento cuando se reemplace de la memoria principal. También pueden darse otros bits de control. Por ejemplo, si la gestión de protección y compartición se gestiona a nivel de segmento, se necesitarán los bits correspondientes a estos fines. El mecanismo básico para la lectura de una palabra de memoria implica la traducción de una dirección virtual, o lógica, consistente en un número de segmento y un desplazamiento, en una dirección física, usando la tabla de segmentos. Debido a que la tabla de segmentos es de tamaño variable,dependiendo del tamaño del proceso, no se puede suponer que se encuentra almacenada en un registro. En su lugar, debe encontrarse en la memoria principal para poder accederse.

Paginación y Segmentación Combinadas

Paginación y segmentación, cada una tiene sus propias ventajas. La paginación es transparente alprogramador y elimina la fragmentación externa, y por tanto proporciona un uso eficiente de la memoria principal. Adicionalmente, debido a que los fragmentos que se mueven entre la memoria y el disco son de un tamaño igual y prefijado, es posible desarrollar algoritmos de gestión de la memoria más sofisticados que exploten el comportamiento de los programas. La segmentación sí es visible al programador y tiene los beneficios que hemos visto anteriormente, incluyendo la posibilidad de manejar estructuras de datos que crecen, modularidad, y darsoporte a la compartición y a la protección. Para combinar las ventajas de ambos, algunos sistemaspor medio del hardware del procesador y del soporte del sistema operativo son capaces de proporcionar ambos.

En un sistema combinado de paginación/segmentación, el espacio de direcciones del usuario sedivide en un número de segmentos, a discreción del programador. Cada segmento es, por su parte, dividido en un número de páginas de tamaño fijo, que son del tamaño de los marcos de la memoria principal. Si un

segmento tiene longitud inferior a una página, el segmento ocupará únicamente una página. Desde el punto de vista del programador, una dirección lógica sigue conteniendo un número de segmento y un desplazamiento dentro de dicho segmento. Desde el punto de vista del sistema, el desplazamiento dentro del segmento es visto como un número de página y un desplazamiento dentro de la página incluida en el segmento.

Referencia Bibliográfica

Stalling, W. (2005). Sistemas Operativos (5th ed., pp. 342-357). PEARSON Educación.