PRE PARCIAL

EN RP

En un sistema con direcciones de memoria de 16 bits que administra una memoria de 32 MBytes de forma paginada bajo demanda con frames que tienen un tamaño de 1KBytes (donde la palabra de memoria es de 1 Byte): ¿Cuántos frames como máximo puede direccionar la dirección de memoria dada?

Conceptos teóricos:

1. **Administración de memoria paginada bajo demanda:** Es un esquema de gestión de memoria en el que la memoria física se divide en bloques de tamaño fijo llamados "marcos" o "frames", y la memoria lógica se divide en bloques de tamaño igual llamados "páginas". Cada página de memoria lógica se mapea a un marco de memoria física en una tabla de páginas. La memoria se asigna bajo demanda, lo que significa que las páginas se cargan en memoria física solo cuando se acceden a ellas.

2. **Direcciones de memoria:** En este sistema, las direcciones de memoria son de 16 bits, lo que significa que cada dirección puede representar hasta 216 (65,536) ubicaciones de memoria diferentes.

3. **Tamaño de la memoria física:** La memoria física tiene un tamaño de 32 MBytes, que es igual a 32 \* 220 bytes (32,768,000 bytes).

4. **Tamaño del frame:** Cada frame de memoria física tiene un tamaño de 1 KByte, que es igual a 1 \* 210 bytes (1,024 bytes).

Ahora, calcularemos nuevamente cuántos frames como máximo puede direccionar la dirección de memoria dada, considerando el desplazamiento:

El desplazamiento es la cantidad de bits necesarios para representar el desplazamiento dentro de un frame. Dado que cada frame tiene un tamaño de 1KByte (1,024 bytes), el desplazamiento se puede calcular utilizando log2(1,024) = 10 bits, ya que 210 = 1,024.

El número de bits restantes (después de considerar el desplazamiento) se utilizará para representar el número máximo de frames que se pueden direccionar.

Número máximo de bits para representar frames = Número de bits totales para direcciones - Número de bits para desplazamiento

Número máximo de frames = 16 bits - 10 bits

Número máximo de frames = 6 bits

El número máximo de frames que se pueden direccionar es 26 = 64 frames.

Por lo tanto, la dirección de memoria dada puede direccionar como máximo 64 frames.

SOLO COMO NOTA DE COLOR:

Memoria física total: 32 MBytes = 32,768 KBytes

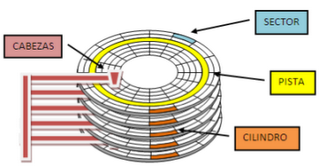
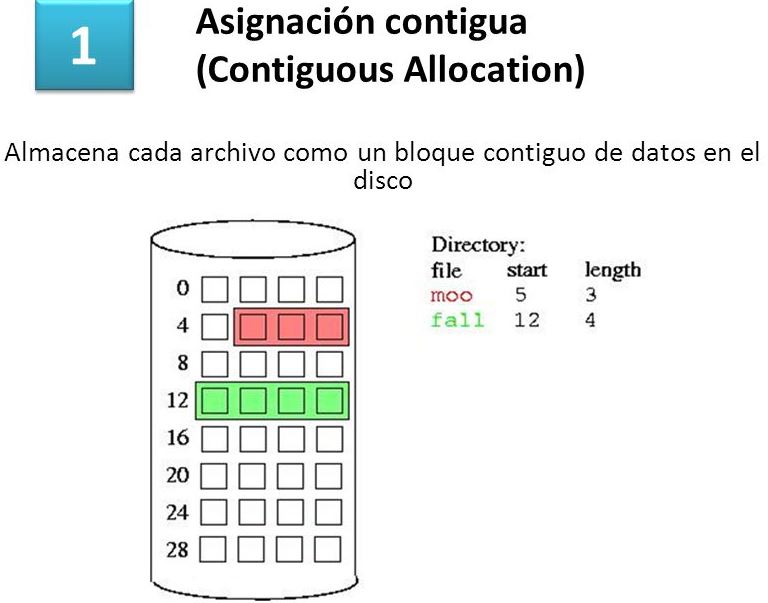
Memoria utilizada para el direccionamiento: 64 KBytes

Memoria sin utilizar = Memoria física total - Memoria utilizada para el direccionamiento

Memoria sin utilizar = 32,768 KBytes - 64 KBytes = 32,704 Kbytes

Es decir, un 99.81% de memoria ha quedado sin usar (al no ser direccionada).

Dada una administración de bloques de disco contigua en el que el directorio está compuesto por el nombre del archivo, la dirección del bloque de comienzo y la cantidad de bloques del archivo, el factor de bloqueo es de 5 Registros Lógicos por Registro Físico, el tamaño del bloque es de 512 Bytes y los registros lógicos se numeran en forma consecutiva comenzando con el número 1: ¿Cuál es la expresión algebraica que permite encontrar en forma directa el bloque del disco que contiene un registro lógico buscado por su número de registro? Dé un ejemplo.



Conceptos teóricos:

1. Administración de bloques de disco contigua: Es un método de organización del almacenamiento en disco donde los bloques de un archivo se ubican de manera contigua en el disco. Esto significa que los bloques consecutivos del archivo se almacenan uno tras otro en sectores físicos adyacentes del disco. La información del archivo se mantiene en una estructura de directorio que incluye el nombre del archivo, la dirección del bloque de comienzo y la cantidad de bloques que ocupa.

2. Factor de bloqueo: Es la relación entre el número de registros lógicos y el número de registros físicos que se almacenan en un bloque de disco. En este caso, se menciona que el factor de bloqueo es de 5 registros lógicos por registro físico, lo que significa que en un bloque de disco se almacenan 5 registros lógicos consecutivos.

3. Tamaño del bloque: Se refiere al tamaño en bytes de un bloque de disco. En este ejercicio, el tamaño del bloque es de 512 bytes.

Expresión algebraica:

Para encontrar el bloque del disco que contiene un registro lógico buscado por su número de registro, utilizamos la siguiente expresión algebraica:

Bloque de disco = (Número de registro lógico - 1) / Factor de bloqueo + Bloque de comienzo

Donde:

- "Número de registro lógico" es el número del registro lógico que estamos buscando.

- "Factor de bloqueo" es el número de registros lógicos que caben en un registro físico.

- "Bloque de comienzo" es el número del bloque de disco en el que comienza el archivo.

Ejemplo:

Supongamos que tenemos un archivo llamado "datos.txt" que ocupa 15 registros lógicos y comienza en el bloque de disco número 100. Además, el factor de bloqueo es de 5 registros lógicos por registro físico.

Queremos encontrar el bloque del disco que contiene el registro lógico número 8 del archivo "datos.txt".

Usamos la expresión:

Bloque de disco = (Número de registro lógico - 1) / Factor de bloqueo + Bloque de comienzo

Para el registro lógico número 8:

Bloque de disco = (8 - 1) / 5 + 100

Bloque de disco = 7 / 5 + 100

Bloque de disco = 1 + 100

Bloque de disco = 101

El bloque del disco que contiene el registro lógico número 8 del archivo "datos.txt" es el bloque número 101.

En un algoritmo de planificación de CPU apropiativo por cálculo pronóstico de duración de la próxima ráfaga de CPU, y dado un proceso A en CPU y un proceso B en estado de listo, en donde el pronóstico de CPU para el proceso B es de 30ms: ¿Qué proceso debe ejecutar en CPU, sabiendo que el pronóstico que llevó a la CPU al proceso A fue de 45ms, y que al momento de llegar a estado listo el proceso B, el proceso A lleva en ejecución 20ms.? Justifique su respuesta.

Para resolver este ejercicio, primero debemos entender los conceptos clave relacionados con el algoritmo de planificación de CPU apropiativo por cálculo pronóstico de duración de la próxima ráfaga de CPU. Los temas teóricos relevantes son:

1. Algoritmos de planificación de CPU: Los sistemas operativos utilizan algoritmos de planificación para decidir qué proceso debe ejecutarse en la CPU en un momento dado. Los algoritmos apropiativos permiten que un proceso en ejecución sea interrumpido para dar paso a otro proceso.

2. Cálculo pronóstico de duración de ráfaga de CPU: Es un método utilizado para estimar cuánto tiempo necesitará un proceso para completar su siguiente ráfaga de CPU. El pronóstico se basa en mediciones anteriores y puede ayudar al planificador a tomar decisiones más informadas.

3. Ráfaga de CPU: Es un período durante el cual un proceso se ejecuta en la CPU sin ser interrumpido.

4. Estado de listo: Los procesos que están listos para ejecutarse pero esperan su turno para acceder a la CPU se encuentran en el estado de listo.

Ahora, veamos cómo podemos resolver el ejercicio:

Dado que tenemos un proceso A en ejecución con un pronóstico de CPU de 45ms y un proceso B en estado de listo con un pronóstico de CPU de 30ms, debemos decidir qué proceso debe ejecutarse en la CPU.

Para ello, calcularemos el tiempo restante de ejecución para ambos procesos y compararemos sus pronósticos de CPU.

1. Proceso A: Lleva 20ms en ejecución, y su pronóstico de CPU es de 45ms. Por lo tanto, le quedan 45ms - 20ms = 25ms de tiempo de ejecución restante.

2. Proceso B: Tiene un pronóstico de CPU de 30ms, ya que aún no ha sido ejecutado.

Ahora, debemos comparar el tiempo restante de ejecución del proceso A (25ms) con el pronóstico de CPU del proceso B (30ms). Si el tiempo restante de ejecución del proceso A es menor o igual al pronóstico de CPU del proceso B, entonces el proceso A debe continuar ejecutándose en la CPU. De lo contrario, el proceso B debería tomar la CPU.

En este caso, el tiempo restante de ejecución del proceso A (25ms) es menor que el pronóstico de CPU del proceso B (30ms). Por lo tanto, el proceso A debe seguir ejecutándose en la CPU.

Justificación: El proceso A tiene un tiempo restante de ejecución menor que el pronóstico de CPU del proceso B, lo que significa que es probable que termine su ejecución antes de que el proceso B complete su próxima ráfaga de CPU. Por lo tanto, tiene más sentido permitir que el proceso A continúe ejecutándose para minimizar el tiempo de espera y maximizar la eficiencia del sistema.

\*\*\*

Es comprensible que existan algunas confusiones en torno a los conceptos de planificación de CPU y las interrupciones. Permíteme aclarar algunos puntos importantes:

1. Algoritmos apropiativos y ráfagas de CPU: En un algoritmo de planificación de CPU apropiativo, efectivamente los procesos pueden ser interrumpidos durante su ejecución, a diferencia de los algoritmos no apropiativos, donde un proceso puede completar su ejecución sin ser interrumpido. Los algoritmos apropiativos se utilizan para mejorar la capacidad de respuesta y la eficiencia en el uso del procesador.

2. Interrupciones en un sistema operativo: Las interrupciones son mecanismos fundamentales en los sistemas operativos. Permiten que el sistema responda a eventos externos o internos, como la llegada de una señal de hardware, el final de una operación de E/S o la expiración de un temporizador. Cuando ocurre una interrupción, el sistema operativo puede suspender temporalmente la ejecución del proceso actual para atender la interrupción y luego continuar la ejecución del proceso interrumpido o elegir otro proceso para ejecutar.

Ahora bien, si el proceso B hubiera tenido un pronóstico menor, digamos de 15ms, la situación sería diferente. Veamos cómo se resolvería en ese caso:

1. Proceso A: Lleva 20ms en ejecución, y su pronóstico de CPU es de 45ms. Por lo tanto, le quedan 45ms - 20ms = 25ms de tiempo de ejecución restante.

2. Proceso B: Tiene un pronóstico de CPU de 15ms, ya que aún no ha sido ejecutado.

En esta situación, el tiempo restante de ejecución del proceso A (25ms) sería mayor que el pronóstico de CPU del proceso B (15ms). Dado que el tiempo restante de ejecución del proceso A es mayor que el pronóstico de CPU del proceso B, el sistema operativo podría decidir interrumpir el proceso A y dar paso al proceso B, siempre que el proceso B esté en estado de listo y esperando su turno para ejecutarse.

Justificación: Si el tiempo restante de ejecución del proceso A es mayor que el pronóstico de CPU del proceso B, el sistema operativo podría optar por interrumpir el proceso A para permitir que el proceso B utilice la CPU. Esta decisión se tomaría para mejorar la eficiencia en el uso del procesador, ya que el proceso B tiene una ráfaga más corta, lo que podría agilizar el cambio de contexto y reducir el tiempo de espera de los procesos.

En resumen, los procesos en un algoritmo de planificación de CPU apropiativo pueden ser interrumpidos durante su ejecución. La decisión de si interrumpir o no un proceso depende de varios factores, como el tiempo restante de ejecución del proceso actual y el pronóstico de CPU del próximo proceso en estado de listo. El objetivo es maximizar la eficiencia del sistema y mejorar los tiempos de respuesta para los procesos.

EN VC

Considere un espacio de direcciones lógicas de 8 páginas de 1024 bytes cada una, y un espacio de direcciones físicas de 32 frames: ¿Cuantos bits son necesarios para especificar una dirección lógica? ¿Cuantos bits son necesarios para especificar una dirección física?

Dirección lógica = # para 8 páginas + # para 1024 bytes desplazamiento

= 3 bits + 10 bits = 13 bits

Dirección física = # para 32 frames + # para desplazamiento (igual que en página)

= 5 + 10 = 15 bits

Considere un sistema con un espacio lógico en memoria de 128 K páginas con 8 KB cada una, una memoria física de 64 MB y direccionamiento al nivel de byte: ¿Cuantos bits hay en la dirección lógica? ¿Cuantos bits hay en la dirección física?

Memoria lógica = 128.000 páginas \* 8 KB = 1.024.000 KB

Memoria física = 64 MB = 65.536 KB

Dirección lógica = # para 128.000 páginas + # para 8.192 bytes (8 KB) desplazamiento

= 17 bits + 13 bits = 30 bits

Dirección física = # para 65.536 KB => 8.192 frames <= capacidad frame = capacidad página

= 13 bits frames + 13 bits desplazamiento (igual que página) = 26 bits

Dado un sistema de paginación simple o pura donde cada frame direcciona 1024 palabras de 64 bits, en el que deben residir simultáneamente en memoria principal los procesos A, B, C y D con los requerimientos de memoria que se indican en el cuadro siguiente. Se requiere calcular cuantos frames demanda cada proceso y el mínimo de frames que debe poseer la memoria para satisfacer las necesidades de estos procesos.

|  |  |  |
| --- | --- | --- |
| Proceso | Memoria requerida en Kbytes | Memoria requerida en cantidad de Frames |
| A | 1248 KB | 1248 / 8 = 156 |
| B | 6240 KB | 6240 / 8 = 780 |
| C | 2600 KB | 2600 / 8 = 325 |
| D | 2080 KB | 2080 / 8 = 260 |

64 bits de cada palabra = 8 bytes

1024 palabras = 8192 bytes = 8 kilobytes

156 + 780 + 325 + 260 = 1521

Sea un sistema de memoria virtual paginada con direcciones lógicas de 32 bits que proporcionan un espacio lógico virtual de 220 páginas y con una memoria física de 32 MB. ¿Cuánto ocupará la tabla de marcos o frames si cada entrada de la misma ocupa 32 bits?

Cada dirección consta de: 8 bits de página y 24 bits de desplazamiento

26 bits referencian 16,777,216 Bytes = 16 MegaBytes

Si la memoria física es de 32 MB, solo entran 2 frames de 16 MB cada uno

La tabla de marcos ocupa entonces 1 bit

Considere un sistema que trabaja con un esquema de paginación a demanda. El tamaño de la página es de 4K, y las direcciones lógicas son de 32 bits. Se pide determine cuántas páginas se pueden direccionar y a qué página hace referencia la dirección 12345 (base 10).

4 KiloBytes = 4.096 Bytes

Necesito 12 bits para desplazamiento

Tengo 20 bits para páginas

Con 20 bits direcciono 1.048.576 páginas

La página 12345 es 11 (20 bits de página) 000000111001 (12 bits de desplazamiento)

11 es la página 3