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[[1]](#footnote-1) es de 1 Byte): ¿Cuántos frames como máximo puede direccionar la dirección de memoria dada?

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):

¿Cuál es el tamaño de una dirección de memoria?

El tamaño de una dirección de memoria es de 16 bits.

¿Cuál es el tamaño de un frame?

El tamaño de un frame es de 1KBytes = 1024 bytes.

¿Cuántos frames como máximo puede direccionar la dirección de memoria dada?

El número máximo de frames que se pueden direccionar con una dirección de memoria de 16 bits es 65536[[2]](#footnote-2)/1024 = 64.

¿Cuál es el tamaño máximo de una palabra de memoria?

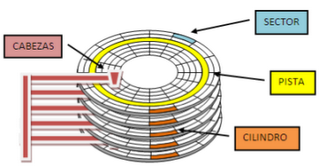
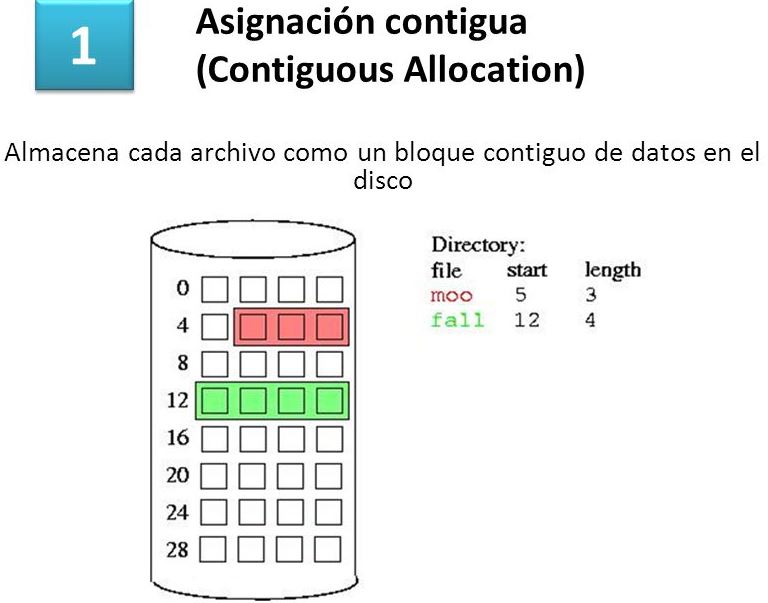
El tamaño máximo de una palabra de memoria es de 1 byte.

Explicación:

El tamaño de una dirección de memoria es de 16 bits, lo que significa que puede direccionar hasta 2^16 = 65536 bytes. Un frame tiene un tamaño de 1KBytes = 1024 bytes, por lo que el número máximo de frames que se pueden direccionar con una dirección de memoria de 16 bits es 65536/1024 = 64 (debido al desplazamiento). El tamaño máximo de una palabra de memoria es de 1 byte, independientemente del tamaño de la dirección de memoria o el tamaño del frame.

En un sistema con paginación bajo demanda, es posible que no todos los frames se encuentren en memoria física en todo momento. Cuando se necesita un frame que no está en memoria física, se debe traer a memoria física desde el almacenamiento secundario.

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[[3]](#footnote-3) 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.



La expresión algebraica que permite encontrar en forma directa el bloque del disco que contiene un registro lógico buscado por su número de registro es:

Bloque del disco = (Número de registro / Factor de bloqueo) + Dirección del bloque de comienzo

Por ejemplo, si el número de registro es 10, el factor de bloqueo es 5 y la dirección del bloque de comienzo es 100, entonces el bloque del disco que contiene el registro lógico 10 es:

Bloque del disco = (10 / 5) + 100 = 2 + 100 = 102

Por lo tanto, el bloque del disco que contiene el registro lógico 10 es el bloque 102.

Esta expresión algebraica se puede utilizar para encontrar el bloque del disco que contiene cualquier registro lógico de cualquier archivo.

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.

En un sistema operativo con un algoritmo de planificación de CPU apropiativo por cálculo pronóstico de duración de la próxima ráfaga de CPU, el proceso que debe ejecutar en la CPU se decide de la siguiente manera:

1. Se calcula el tiempo restante de ejecución del proceso actual.
2. Se calcula el pronóstico de CPU del siguiente proceso en la cola.
3. Si el tiempo restante de ejecución del proceso actual es menor que el pronóstico de CPU del siguiente proceso, entonces el proceso actual debe seguir ejecutándose en la CPU.
4. Si el tiempo restante de ejecución del proceso actual es mayor que el pronóstico de CPU del siguiente proceso, entonces el siguiente proceso debe ejecutarse en la CPU.

En el caso dado, el proceso A tiene un tiempo restante de ejecución de 25ms, mientras que el proceso B tiene un pronóstico de CPU de 30ms. Por lo tanto, el proceso A debe seguir ejecutándose en la CPU.

La justificación de esta decisión es que el proceso A es más probable que termine antes que el proceso B. Esto se debe a que el proceso A tiene un tiempo restante de ejecución más corto. Si el proceso A se interrumpiera, el proceso B tendría que esperar más tiempo para ejecutarse, ya que el planificador de CPU tendría que volver a calcular el pronóstico de CPU para el proceso B. Este cálculo puede llevar algo de tiempo, lo que podría afectar negativamente al rendimiento del sistema.

Por lo tanto, es mejor que el proceso A siga ejecutándose en la CPU, ya que es más probable que termine antes y no afectará negativamente al rendimiento del sistema.

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

1. En este contexto, una palabra de memoria es el tamaño más pequeño de datos que se puede direccionar en la memoria. [↑](#footnote-ref-1)
2. 16 bits pueden direccionar 65.536 direcciones. [↑](#footnote-ref-2)
3. Muy bien, vamos a empezar con el factor de bloqueo. El factor de bloqueo es el número de registros lógicos que se pueden almacenar en un registro físico. Es decir, cada vez que se quiere almacenar un registro lógico en el disco duro, el sistema operativo almacena un factor de bloqueo de registros lógicos en el mismo registro físico. [↑](#footnote-ref-3)