1

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):

1. ¿En qué consiste la administración de memoria paginada bajo demanda?
2. ¿En qué influye que sea una memoria paginada bajo demanda en la solución de este ejercicio?
3. ¿Cuántos frames tiene la memoria?
4. ¿Qué es una palabra de memoria?
5. ¿Saber el tamaño de la palabra de memoria influye en la solución de este ejercicio?
6. ¿Cuántos frames como máximo puede direccionar la dirección de memoria dada?

a) La administración de memoria paginada bajo demanda es una técnica de gestión de memoria que se utiliza en sistemas operativos para asignar y liberar memoria de manera eficiente. En lugar de asignar toda la memoria necesaria para un proceso en el momento de su carga, la memoria se asigna en páginas o fragmentos más pequeños llamados marcos (frames). Estos marcos se cargan en la memoria principal solo cuando se necesitan. Cuando un proceso accede a una dirección de memoria que no está en la memoria principal, se produce un fallo de página y el sistema operativo carga la página correspondiente desde el almacenamiento secundario a la memoria principal.

b) El hecho de que sea una memoria paginada bajo demanda influye en la solución del ejercicio porque implica que no todos los frames están cargados en la memoria principal de manera permanente. Solo se cargarán en la memoria principal los frames necesarios para satisfacer las solicitudes de lectura o escritura de los procesos en ejecución. Esto implica que no todos los frames estarán disponibles en todo momento y puede requerir una gestión cuidadosa de la memoria para garantizar un acceso eficiente y minimizar los fallos de página.

c) La memoria se administra de forma paginada con frames de tamaño 1KByte. Dado que la memoria total es de 32 MBytes, y cada frame tiene un tamaño de 1KByte (1024 bytes), podemos calcular la cantidad de frames dividiendo el tamaño total de la memoria entre el tamaño de cada frame:

32 MBytes = 32 \* 1024 \* 1024 bytes

Tamaño del frame = 1KByte = 1024 bytes

Cantidad de frames = (32 \* 1024 \* 1024) / 1024 = 32 \* 1024 = 32,768 frames

Por lo tanto, la memoria tiene 32,768 frames.

d) Una palabra de memoria es la unidad de datos que se lee o escribe en la memoria. En este caso, se menciona que una palabra de memoria es de 1 Byte. Por lo tanto, cada dirección de memoria en este sistema se refiere a una ubicación que contiene 1 Byte de datos.

e) El tamaño de la palabra de memoria no influye directamente en la solución de este ejercicio, ya que el ejercicio se enfoca en la administración de memoria paginada y la cantidad de frames. El tamaño de la palabra de memoria es útil para comprender cómo se organizan los datos en la memoria, pero no afecta directamente la administración de memoria paginada bajo demanda.

f) La dirección de memoria en este sistema tiene 16 bits, lo que implica que puede direccionar un total de 2^16 = 65,536 ubicaciones de memoria diferentes. Como se mencionó anteriormente, la memoria se administra en frames de tamaño 1KByte, por lo que podemos calcular la cantidad máxima de frames que se pueden direccionar dividiendo el tamaño total de la memoria entre el tamaño de cada frame:

Tamaño total de la memoria = 32 MBytes = 32 \* 1024 \* 1024 bytes

Tamaño del frame = 1KByte = 1024 bytes

Cantidad máxima de frames direccionables = (32 \* 1024 \* 1024) / 1024 = 32 \* 1024 = 32,768 frames

Por lo tanto, la dirección de memoria dada puede direccionar como máximo 32,768 frames.

2

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:

a) ¿Qué es una administración de bloques de disco contigua?

b) ¿Qué son los bloques?

c) ¿Qué es el factor de bloqueo?

d) ¿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.

a) La administración de bloques de disco contigua es un método de organización de archivos en el disco donde los bloques de datos se almacenan de forma consecutiva en el disco. Esto significa que los bloques de un archivo se ubican uno después del otro en el disco sin espacios vacíos entre ellos. El directorio del archivo contiene información sobre la ubicación inicial del primer bloque y la cantidad de bloques utilizados por el archivo.

b) Los bloques son unidades de almacenamiento en el disco que contienen una cantidad fija de datos. En este caso, se menciona que el tamaño del bloque es de 512 Bytes, lo que significa que cada bloque puede almacenar 512 Bytes de información.

c) El factor de bloqueo es la relación entre los registros lógicos y los registros físicos en el disco. En este caso, se menciona que el factor de bloqueo es de 5 Registros Lógicos por Registro Físico. Esto significa que cada registro físico en el disco contiene 5 registros lógicos consecutivos.

d) La expresión algebraica que permite encontrar en forma directa el bloque del disco que contiene un registro lógico buscado por su número de registro se puede calcular utilizando la siguiente fórmula:

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

Por ejemplo, supongamos que queremos encontrar el bloque del disco que contiene el registro lógico número 10, y el bloque de comienzo es el bloque 3. Utilizando la fórmula:

Bloque del disco = (10 - 1) / 5 + 3

Bloque del disco = 9 / 5 + 3

Bloque del disco = 1 + 3

Bloque del disco = 4

Entonces, el bloque del disco que contiene el registro lógico número 10 es el bloque 4.

3

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:

a) ¿Qué es una ráfaga de CPU?

b) ¿Qué es lo apropiativo?

c) ¿Qué es un algoritmo de planificación de CPU apropiativo por cálculo pronóstico de duración de la próxima ráfaga de CPU?

d) ¿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.

a) Una ráfaga de CPU se refiere al período de tiempo durante el cual un proceso se ejecuta en la CPU sin interrupciones ni cambios de contexto. Es el tiempo que un proceso utiliza los recursos de la CPU para realizar sus tareas.

b) En el contexto de la planificación de la CPU, "apropiativo" significa que el planificador de la CPU tiene la capacidad de interrumpir un proceso en ejecución y asignar la CPU a otro proceso si es necesario. En otras palabras, el planificador puede tomar decisiones en tiempo real y cambiar la asignación de la CPU según las prioridades de los procesos y las políticas de planificación.

c) Un algoritmo de planificación de CPU apropiativo por cálculo pronóstico de duración de la próxima ráfaga de CPU es un algoritmo que utiliza información de pronóstico sobre la duración de la próxima ráfaga de CPU de los procesos para tomar decisiones de planificación. En este enfoque, se estima la duración de la próxima ráfaga de CPU de cada proceso en función de su historial de ejecución anterior y otros factores, y se utiliza esta información para priorizar la asignación de la CPU.

d) Para determinar qué proceso debe ejecutarse en la CPU, consideraremos el pronóstico y el estado actual de los procesos. Dado que el proceso A lleva 20ms en ejecución y el pronóstico de CPU para el proceso B es de 30ms, podemos comparar los tiempos restantes.

El tiempo restante para el proceso A sería de 45ms (pronóstico inicial) - 20ms (tiempo transcurrido) = 25ms.

El tiempo pronosticado para el proceso B es de 30ms.

En este caso, el proceso B tiene una duración pronosticada menor que el tiempo restante del proceso A. Por lo tanto, sería más adecuado ejecutar el proceso B en la CPU, ya que se espera que finalice antes y se puede utilizar de manera más eficiente el recurso de la CPU.