1. **la utilización del buffer cache en el sistema V UNIX, contempla el esquema “escritura\_diferida”. Indique cual es el objetivo de ese esquema y cuales son sus ventajas y desventajas.** El Kernell posterga la escritura física tanto como sea posible, cuando inicia la escritura señala al Buffer con una marca y escribe el bloque a disco a forma asíncrona, cuando finaliza la escritura se libera el  Buffer y es colocado a la cabeza de la lista libre

**Ventajas:**

* con los algoritmos de buffer se garantiza la integridad del sistema por que en memoria cache se obtiene una sola copia de los bloques de disco
* se establece un acceso uniforme a disco.
* El sistema no tiene restricciones de orden para poner los datos en la E/S. del proceso.
* Reduce la cantidad de trafico de disco incrementando el rendimiento del sistema
* Los algoritmos de buffer serializan los accesos evitando la corrupción de los datos

**Desventaja:**

* En escritura demorada el Kernell no escribe inmediatamente los datos a disco si el sistema es vulnerable a las caídas deja al disco en forma incorrecta.
* El buffers cache requiere una copia original de los datos cuando se hace la lectura y escritura para un proceso.

1. **Suponiendo que el Kernel marco un bloque de la memoria cache para “escritura\_diferida”. ¿Qué sucede cuando otro proceso se asigna a ese bloque?**
2. **De su cola hash** El kernel en representación de un proceso no encuentra el bloque de disco en su cola HASH de modo que intenta asignar un nuevo buffer de la cola de la lista libre, similar que el ambiente 2, sin embargo la lista libre no tiene buffer disponible por lo que el proceso es puesto a dormir hasta que algún otro proceso libere el buffer.
3. **De la lista libre** Cuando el kernel intenta asignar el buffer de la lista libre tiene la marca de escritura diferida de modo que debe escribir el contenido del buffer en el disco antes de poder utilizar ese buffer. El kernel inicia una escritura asíncrona a disco y trata de asignar otro buffer de la lista libre, cuando se completa la escritura asíncrona el kernel libera ese buffer y lo coloca a la cabeza de la lista libre, el buffer había empezado al final de la lista libre y realizó el recorrido hasta la cabeza de la lista libre.
4. **Los algoritmos para la asignación y liberación de inodos y bloques de disco son similares por el hecho que el kernel coloca en el superbloque índices de recursos libres, números de bloques y números de inodos. Para ello adopta la metodología de memoria cache. Para los bloques libres usa una lista encadenada de números de bloques libres, pero para los inodos libres no sigue el mismo método. Explique las tres razones para no adoptar la misma forma en ambos.**

los algoritmos para asignación y liberación de iNodos y de bloque de disco son similares en el sentido que el kernel usa el super-bloque con una memoria caché conteniendo índice de recursos libres, numero de bloques y números de iNodos, mantiene una lista encadenada de números de bloques de manera que todo número de bloque libre se encuentra en esa lista encadenada, pero tal cosa nos sucede en la lista de iNodos libres debido a tres razones:

1.- el kernel puede determinar si un nodo está libre mediante infección, si el campo tipo de archivo está libre, sin embargo no se puede seguir el mismo método con un bloque

2.- los bloques de un disco son apropiados para usar listas encadenada de bloque de disco puede contener sin dificultad una lista grande de numero de bloques que están libres pero los iNodos no tienen el espacio suficiente para contener listas grandes de números de iNodos libres.

3.- los usuarios consumen en mayor proporción bloques de discos que iNodos de manera que la aparente disminución de desempeño cuando se examina el disco buscando iNodos libres no es tan crítico como una búsqueda de bloques libres.

1. **En el sistema V UNIX el inodo tiene una tabla de 13 entradas. Las 10 primeras marcadas “directo” se refieren cada una a un bloque de disco conteniendo datos reales. La entrada “indirecto sencillo” se refiere a un bloque conteniendo una lista de números de bloques directos. La entrada “indirecto doble” contiene la lista de números de bloques indirecto sencillo. La entrada “indirecto triple” contiene una lista de números de bloques indirecto doble. Cada bloque consta de 512 Bytes (0 a 511), y un numero de bloque es direccionable por 4 Bytes.**

**Halle la dirección en el archivo de datos de los siguientes bytes:**

1. **159000 II) 8040 III) 4100**

**Respondiendo en cada caso:**

1. **El tipo de entrada que se aplica (directo, indirecto sencillo, …)**
2. **La posición secuencial del bloque de daros que contiene el byte (0, 1, 2, …)**
3. **El offset (desplazamiento) de byte (0, 1, 2, …)**
4. **159000**

159000 / 512 =**310.55**

* 159000 512

Directa: 10 \* 512 = 5120

Indirecta: 128 \* 512 = 65536

Indirecta doble: 1282 \*512 = 8388608

Indirecta triple: 1283 \*512 = 1073741824

-158720 310

280

a) Indirecta doble

b) bloque 310

c) 280

**II) 8040**

8040 / 512 =**15.70**

* 8040 512

-7680 15

360

a) Indirecto

b) bloque 15

c) 360

**III) 4100**

4100 / 512 =**8**

* 4100 512

-4096 8

4

a) Indirecto

b) bloque 15

c) 4