

Si la reserva de marcos libre cae por debajo de un cierto límite se utiliza el algoritmo de reemplazamiento para seleccionar páginas reemplazables, cuyos marcos se agregan a la lista hasta alcanzar cierto límite máximo.

Cuando se necesita un marco libre se coge el situado en la cabecera de la lista de marcos libres. Por otro lado, si se selecciona alguna página *i* como reemplazable su marco *j* es colocado al final de la lista si la página no ha sido modificada. Además el campo válido de la entrada *i* de la tabla de páginas correspondiente se desactiva. Si la página ha sido modificada, entonces ingresa en una lista de páginas modificadas que serán escritas por lotes en la memoria secundaria. Una vez escritas pasan a ser colocadas al final de la lista de marcos libres.

Mientras la página *i* siga cargada en el marco *j* puede ser utilizada. Por este motivo cuando se produce un fallo de página se comprueba la lista de marcos libres por si alguno de ellos contuviera la página que ha producido el fallo, y de este modo ahorrarse la lectura de la página en memoria secundaria. Así, la lista de marcos libres se utiliza como una memoria caché software de páginas.

Preguntas de autoevaluación tema 8

8.1. Explicar qué es un dispositivo modo bloque y un dispositivo modo carácter. Señalar algunos ejemplos de cada tipo.

Modo bloque almacena información en bloques de tamaño fijo, normalmente 512 bytes o una potencia de dos múltiplos de la anterior, cada uno de los cuales tiene asignado un determinado número identificador o dirección. Ejemplos de dispositivos modo bloque son los discos y unidades de cinta.

Modo carácter envía o recibe información como una secuencia o flujo lineal de bytes, en ellos la información no se organiza con una estructura concreta por lo que no es direccionable, y en consecuencia no permite la realización de operaciones de búsqueda.

8.2. ¿Qué información debe pasar un proceso a la llamada al sistema que invoque para solicitar una petición de E/S?

La dirección lógica del dispositivo de E/S sobre la que desea operar (usualmente especificada mediante una ruta simbólica que puede ser traducida por el subsistema de E/S), la dirección lógica del espacio del proceso donde se almacenarán los datos que se lean en el dispositivo (o donde se encuentran los datos que se escribirán en el dispositivo) y el número de bytes que se van a transferir.

8.3. Señalar dos ejemplos donde el propio sistema operativo realiza operaciones de E/S durante la realización de alguna de sus tareas administrativas.

Dar formato a una cadena de caracteres para que sea mostrada en la pantalla de una determinada forma, o interpretar una cadena de caracteres introducida por teclado para asociarla a las variables del proceso adecuadas.

8.4. ¿Cuáles son las capas de software de E/S del núcleo de un sistema operativo?

Subsistema de E/S, drivers de los dispositivos de E/S y manejadores de las interrupciones.

8.5. ¿Qué función tiene el subsistema de E/S?

Es el componente del SO que se encarga de efectuar todas aquellas tareas necesarias para la realización de las operaciones de E/S que son comunes a todos los dispositivos independientes de los mismos. Es decir, gestiona la parte independiente del dispositivo de todas las operaciones de E/S.

8.6. Enumerar las tareas del subsistema de E/S.

❖ *Asignación y liberación de dispositivos dedicados.*

Algunos dispositivos E/S, como por ejemplo un disco, pueden ser utilizados simultáneamente por varios procesos. No existe ningún inconveniente en que varios procesos tengan abiertos diferentes archivos de un disco. Sin embargo, otros dispositivos, como una unidad de cinta magnética o una impresora, solo puede ser utilizados simultáneamente por un único proceso. A estos últimos dispositivos se les conoce como dispositivos dedicados. Es responsabilidad del subsistema de E/S determinar si una petición de E/S de un proceso A sobre un dispositivo dedicado puede ser aceptada o debe ser rechazada. Si la petición es aceptada, pero el dispositivo se encuentra ocupado atendiendo la petición de

otro proceso B, el subsistema de E/S puede bloquear al proceso A y colocarlo en una cola de procesos bloqueados en espera del uso del dispositivo. Cuando el dispositivo se encuentre disponible el primer proceso de la cola será desbloqueado y se le asignará el uso del dispositivo. De esta forma, el proceso A obtendrá cuando le llegue su turno el uso del dispositivo.

❖ ***Bloqueo de procesos que solicitan una operación de E/S.***

Cuando al subsistema de E/S le llega, a través de la interfaz de llamadas al sistema, una petición de un proceso para la realización de una operación de E/S, éste puede pasar o no al estado bloqueado al proceso que solicitó la operación de E/S. La decisión de bloquear un proceso dependerá de si el proceso invocó una llamada al sistema de tipo bloqueante o del tipo y estado del dispositivo involucrado en la operación de E/S solicitada.

❖ ***Planificación de la E/S.***

❖ ***Invocación del driver de dispositivo apropiado.***

❖ ***Almacenamiento temporal de datos de E/S o buffering.***

❖ ***Proporcionar un tamaño de bloque uniforme a los niveles superiores del software.***

❖ ***Gestión de los errores producidos en una operación de E/S.***

8.7. ¿Qué es un driver de dispositivo?

Contiene el código que permite a un SO controlar un determinado tipo de dispositivo de E/S. Un driver se diseña teniendo en cuenta las especificaciones de la interfaz de drivers del subsistema de E/S de cada SO y las características del dispositivo.

8.8. ¿Qué significa y qué ventajas presenta el hecho de que la interfaz para los drivers de los dispositivos de un subsistema de E/S sea uniforme?

Que las funciones que debe aportar el driver y las funciones del núcleo que puede invocar son las mismas para todos los drivers. Si la interfaz no fuera uniforme habría que modificar el código del SO para adaptar el subsistema de E/S a cada nuevo driver. Además el subsistema de E/S tendría la necesidad de conocer qué funciones puede ejecutar cada driver, ya que todos no pueden ejecutar las mismas.

8.9. ¿Con qué elementos interactúa el driver de un dispositivo?

Un driver de dispositivo interactúa con el subsistema de E/S y con el controlador de E/S que controla el dispositivo. Un driver suministra al subsistema de E/S el conjunto de funciones que se pueden realizar sobre el dispositivo, tales como lectura o escritura. Además un driver puede invocar a ciertas rutinas o procedimientos del núcleo. Los procedimientos a los que tiene acceso un driver quedan definidos por la interfaz de drivers del subsistema de E/S.

8.10. ¿Qué acciones realiza un driver?

- ❖ **Comprobar que los parámetros de la función invocada por el subsistema de E/S son correctos y que la operación de E/S solicitada se puede realizar.** En caso contrario devuelve un error.
- ❖ **Traducir los parámetros de dicha función en parámetros específicos del dispositivo.**
- ❖ **Comprobar si el dispositivo de E/S está ocupado atendiendo alguna petición anterior de E/S. Si el dispositivo está ocupado entonces coloca la petición en una cola.** Si el dispositivo no está atendiendo ninguna petición, comprueba si se encuentra preparado, ya que quizás el driver deba activar e inicializar el dispositivo.
- ❖ **Generar un conjunto de órdenes para el controlador del dispositivo dependiendo de la petición de E/S solicitada por el subsistema de E/S.** Dichas órdenes son cargadas en los registros del controlador. El driver puede comprobar que el controlador acepta una orden y que se encuentra listo para aceptar la siguiente.
- ❖ **Una vez transmitidas todas las órdenes al controlador, el driver debe esperar a que el controlador las ejecute.** Si el tiempo de espera estimado es importante, entonces el driver se bloquea usando algún mecanismo de sincronización hasta que el controlador finalice. Cuando la operación de E/S se complete el controlador activará una interrupción. El manejador o rutina de servicio de dicha interrupción despertará al driver, si éste se bloqueó.
- ❖ **Comprobar que no se han producido errores en la operación de E/S.** En dicho caso quizás transfiera al subsistema de E/S el resultado de la operación de E/S, por ejemplo, un bloque leído en el disco, o puede que simplemente le informe de que la operación se ha completado. Si se ha producido algún error y el driver sabe cómo resolverlo realiza la acción correspondiente. Si no sabe cómo resolverlo o la solución que plantea no surte efecto,

entonces informa del error al subsistema de E/S para que tome las medidas correspondientes oportunas.

- ❖ **Examinar la cola de peticiones de E/S pendientes, si existe alguna procede a atenderla.** Si la cola está vacía el driver se bloqueará en espera de la llegada de nuevas peticiones.

8.11. Explicar las características y el funcionamiento de los manejadores de interrupciones.

Cuando finaliza una operación de E/S en un dispositivo y éste se encuentra preparado para procesar otra, el controlador de E/S que lo supervisa genera una interrupción. En un computador que implemente un sistema de interrupciones vectorizadas cada interrupción suministra un número denominado número del vector de interrupción que se utiliza como índice en una tabla, denominada tabla de vectores de interrupción. Esta tabla usualmente se encuentra almacenada en las posiciones más bajas de memoria. Cada entrada de esta tabla se denomina vector de interrupción, que entre otras informaciones contiene la dirección de comienzo del manejador (handler) de la interrupción.

Los manejadores de interrupciones forman parte del núcleo del SO y son extremadamente dependientes del hardware. Por este motivo, a la hora de portar un SO a una nueva arquitectura, esta parte del núcleo siempre tiene que ser reescrita.

Con objeto de que el rendimiento del computador sea óptimo, los manipuladores de interrupciones tienen una alta prioridad de ejecución. Además su código suele ser pequeño y rápido de ejecutar. Las acciones específicas que realiza un manejador de interrupciones dependen de cada tipo de interrupción.

Si el driver del dispositivo se bloqueó en espera de que el controlador de E/S estuviera preparado para procesar otra petición de E/S, entonces una acción que debe realizar un manejador de interrupción es el desbloqueo del driver del dispositivo mediante el uso del mismo mecanismo de sincronización (semáforo, mensajes, ...) que utilizó el driver para bloquearse.

En el caso en que no se realice DMA un manejador de una interrupción también se puede encargar de transferir datos desde un registro del controlador del dispositivo a un buffer en el espacio del núcleo, o viceversa, en función de si se trata de una operación

de lectura o de escritura, respectivamente. Esta función, en ocasiones, la puede realizar el propio driver en lugar del manejador.

8.12. ¿Qué es el buffering? ¿Qué problemas resuelve?

En la transferencia directa una transferencia se almacenaría el bloque directamente en una zona contigua de la región de datos del espacio de direcciones virtuales del proceso. En un sistema con paginación dicha zona formará parte de una página *i* del proceso.

Esta forma de realizar la transferencia de datos obliga al SO a mantener cargada en un marco *j* de la memoria principal la página *i* mientras no se complete la operación de E/S, ya que en caso contrario se perderían los datos. Las operaciones E/S son lentas en comparación con los accesos a memoria principal, por lo que el marco quedará bloqueado un tiempo importante. Cuanto mayor sea el número de marcos bloqueados, menor será el número de marcos candidatos a ser reemplazados al atender un fallo de página lo que produce sobrepaginación.

Otro problema es la imposibilidad de poder intercambiar al proceso hasta que la operación E/S no se haya terminado.

Para evitar los problemas que plantea la transferencia directa de datos desde el dispositivo al espacio de usuario del proceso los SO implementan la técnica conocida como buffering, que consiste en almacenar temporalmente en buffers los datos que se transfieren entre un dispositivo y un proceso, o viceversa. Un buffer es un área de memoria principal a la que únicamente tiene acceso el SO, es decir, pertenece al espacio del núcleo.

El subsistema E/S es el encargado de asignar buffers para las operaciones de E/S. Por su parte, las rutinas de servicio de las interrupciones o los drivers, se encargan de transferir los datos entre los buffers y los dispositivos, o viceversa, dependiendo de si se trata de una operación de entrada o salida.

8.13. Enumerar y explicar brevemente las diferentes estrategias para la realización del buffering.

- ❖ **Buffering con buffer único.** Los datos son transferidos primero desde el dispositivo al buffer ubicado en el espacio del núcleo. Posteriormente los datos se transfieren desde el buffer al espacio del proceso. En una operación de escritura, los datos se transfieren primero desde el espacio del proceso al buffer en el espacio del núcleo.

- ❖ **Buffering con dos buffers o buffering doble.** En una operación de E/S se asignan dos buffers. Cuando el primer buffer se llena, se pasa a llenar el segundo mientras se vacía el primero. Si cuando el segundo está lleno se puede comenzar a llenar el primero de nuevo.
- ❖ **Buffering circular.** Consiste en utilizar más de dos buffers para una operación E/S. Primero se llena el primero, luego el segundo, el tercero, etc... Cuando se termina de llenar el primer buffer se pueden comenzar a vaciar. Si se termina de llenar el último buffer del conjunto y el primero ya ha sido vaciado se puede comenzar otro ciclo.

8.14. ¿En qué consiste una caché de buffers de bloques de disco?

Se trata de un área de memoria principal del espacio del núcleo reservada para almacenar buffers. Dichos buffers contienen los bloques de disco que han sido recientemente transferidos.

El funcionamiento es similar al de una caché hardware. Cuando un proceso solicita la realización de una operación de lectura en disco, el SO comprueba si los datos solicitados se encuentran almacenados en la caché de buffers. En caso afirmativo, sirve los datos al proceso. Si los datos solicitados no están en la caché de buffers, entonces es necesario realizar una operación E/S. Cuando el subsistema de E/S asigna buffers para una operación de E/S en disco debe comprobar si existen buffers libres en la caché de buffers. Si no es así, tendrá que utilizar algún algoritmo de reemplazamiento, como los comentados al estudiar paginación (FIFO, LRU, etc), para seleccionar los buffers cuyo contenido va a ser reemplazado.

8.15. ¿Qué es el spooling y cómo se implementa?

En sistemas multiprogramados el SO se debe encargar de asignar y controlar el uso de los dispositivos E/S. En el caso de los dedicados como la impresora se utiliza la técnica de control llamada spooling. Se implementa con un proceso demonio y con un directorio especial. El proceso spooling es el único autorizado para escribir en el dispositivo. Si un proceso de usuario quiere escribir en el dispositivo se deben enviar los archivos al directorio de spooling.

8.16. ¿En qué casos se usa la técnica de spooling?

En el control de la impresora o en la transferencia de paquetes de datos de una red.

8.17. Describe las partes que componen un reloj programable y su funcionamiento.

Entre otros elementos un contador, un registro y un cristal de cuarzo.

8.18. ¿Qué es un tic de reloj?

Es el tiempo entre dos señales periódicas.

8.19. Enumerar y explicar brevemente las tareas del sistema operativo en las que utiliza relojes programables.

- ❖ **Planificación de procesos.** Si el SO utiliza el algoritmo de turno rotatorio para la planificación de procesos debe vigilar que el proceso en ejecución no exceda el cuanto de uso de CPU asignado. Para ello carga en un reloj programable el valor del cuanto y en cada tic de reloj lo va decrementando. Cuando llega a 0, significa que se ha terminado el cuanto, entonces despierta al planificador para que planifique otro proceso.
- ❖ **Mantenimiento del tiempo real.** Para mantener el tiempo real es necesario fijar un punto de referencia temporal inicial a partir del cual se va llevando la cuenta.
- ❖ **Disparo de alarmas.** Los procesos pueden solicitar al SO mediante alguna llamada al sistema que éste avise cuando haya transcurrido un determinado intervalo de tiempo. Una forma de implementar una alarma sería usar un reloj programable que fuese decrementando. El SO suele gestionar las alarmas manteniendo una lista de alarmas ya que si el número de alarmas es superior al número de relojes programables disponibles sería un error.
- ❖ **Invocación de tareas periódicas del sistema.**

8.20. Explicar la interacción entre el sistema operativo y el reloj del computador.)

El SO utiliza el driver del reloj para interactuar con el reloj del computador. Cuando el reloj genera una interrupción, el manipulador de la interrupción desbloquea el driver y éste debe realizar diferentes tareas, como por ejemplo: decrementar el cuanto del proceso en ejecución, incrementar el tiempo de uso del procesador de dicho proceso, incrementar el reloj de tiempo real y decrementar el tiempo de disparo en la lista de alarmas.

8.21. ¿En qué consiste el formateo a bajo nivel formateo físico de un disco duro?

Consiste en dividir cada superficie de los platos en pistas y cada pista en sectores.

8.22. Describir los elementos en qué se descompone un sector de un disco duro.

- ❖ **Cabecera o preámbulo.** Contiene entre otras informaciones un cierto patrón de bits que permite reconocer al controlador del disco el comienzo del sector, el número de sector y el número de cilindro.
- ❖ **Área de datos.** Almacena la información accesible para el SO y usuarios, su tamaño típico suele ser 512 bytes.
- ❖ **Código de corrección de errores.** Suele ocupar 16 bytes, contiene información que permite al controlador del disco detectar y corregir posibles errores de lectura.

8.23. ¿En qué consiste el método de acceso LBA?

Direccionamiento de bloques lógicos (logic block addressing). Con esta técnica un disco duro se considera un array de bloques lógicos de datos. Cada bloque tiene asignado un número de bloque identificativo o dirección lógica. El 0 suele ser el de la primera pista del cilindro más externo.

El driver del disco solo debe especificar al controlador del disco el número de bloque lógico que hay que leer o escribir. El controlador del disco se encarga de traducir dicho número en una tripleta CHS. En los discos que no soportan LBA, el driver tiene que pasarle en cada operación de E/S una tripleta CHS al controlador del disco.

8.24. ¿En qué consiste el particionamiento de un disco duro?

Consiste en establecer por software una o más particiones en el disco. Una partición es un conjunto de cilindros contiguos. Desde el punto de vista lógico cada partición se considera como un disco distinto.

La información sobre el cilindro de comienzo, el tamaño de cada partición y la partición activa se mantiene en el propio disco en una estructura denominada tabla de particiones que se localiza al final del registro de arranque maestro (MBR) dentro del sector 0 del disco.

8.25. ¿En qué consiste el formateo a alto nivel lógico de un disco duro?

También llamado formateo lógico, consiste en establecer el bloque o sector de arranque y las estructuras de datos de un sistema de archivos (lista o mapas de espacio libre y asignado, y un directorio raíz o inicial vacío). También coloca en la tabla de particiones información sobre el tipo de sistema de archivos colocado en cada partición.

8.26. ¿Bajo qué condiciones el algoritmo de planificación de peticiones de E/S a disco se implementen en el driver del disco? (Respuesta en sección 8.6.2)

Por software en el driver del disco, ya que es la capa de software que conoce los detalles de la estructura interna del disco. Por otro lado, si el controlador del disco puede aceptar un lote de peticiones, el algoritmo de planificación se puede implementar dentro del propio controlador del disco.

En los HD que soportan LBA el algoritmo de planificación del disco se implementa forzosamente en el controlador del disco ya que es el único que conoce los detalles de organización física del disco. Recuérdese que en dicho caso el driver del disco trabaja exclusivamente con direcciones o número de bloques lógicos. Con dicha información no es posible implementar los algoritmos SSTF (shortest search time first) ni LOOK, los cuales necesitaban conocer el número de cilindro de cada petición de E/S al disco.

8.27. Señalar dos técnicas para realizar el reemplazamiento de un sector defectuoso en un disco duro ¿En qué caso el sistema operativo se debe encargar de realizar esta tarea?

- ❖ **Sustitución directa.** Supóngase que el sector j de una pista está dañado, entonces este método consiste en usar un sector de reserva como sector j . El principal atractivo de esta estrategia es que se realiza bastante rápido, simplemente hay que posicionar la cabeza de lectura-escritura sobre el sector de reserva y escribir la cabecera del sector defectuoso al que sustituye. Sin embargo, puede alterar la organización del entrelazado del disco, empeorando el rendimiento del disco.
- ❖ **Desplazamiento de sectores.** Consiste en desplazar una posición cada sector situado entre el sector de reserva y el sector defectuoso. El sector de reserva se utiliza para contener al primer sector que se desplaza. La posición ocupada por el último sector desplazado es utilizada para

albergar al número de sector que estaba defectuoso. Esta estrategia presenta como ventaja que mantiene la organización de una pista por lo que no repercute en el rendimiento del disco. Sin embargo, se trata de una operación lenta, ya que múltiples sectores de una pista tienen que ser reescritos de nuevo.

8.28. Explicar la gestión de la salida por pantalla.

La salida es una red de puntos denominados píxels. Cada píxel tiene asociado un número binario que indica el color con que se representa en pantalla. Por ejemplo, se pueden utilizar un número de 24 bits para representar cada pixel, dedicando 8 bits para cada intensidad de rojo, verde y azul. Para evitar el parpadeo se debe mostrar una imagen de 60 a 100 veces por segundo.

La pantalla se suele conectar a través de un adaptador o controlador gráfico que puede estar integrado o en una tarjeta sobre la placa. Tiene una memoria de video (VRAM) que es una memoria RAM en la que se almacena los pixeles de la imagen que se mostrará en la pantalla.

La imagen que se carga en la VRAM es generada por paquetes de rutinas de creación y manipulación de gráficos independientes del hardware, las cuales son invocadas por las aplicaciones o por el gestor de ventanas de un GUI. Estos paquetes pueden formar parte del núcleo del SO (Windows) o ser externos al mismo (UNIX y Linux). En el primer caso, dichas rutinas se ejecutarán en modo núcleo, en el segundo caso en modo usuario. Para interactuar con el adaptador gráfico las propias rutinas (Windows) o el subsistema de E/S (Unix y Linux) invocan al driver del adaptador, que es el que conoce los detalles del hardware.

8.29. Explicar la gestión de la entrada del teclado.

Es un dispositivo modo carácter de entrada. Cuando el usuario pulsa o suelta una tecla el teclado envía el número de tecla a un registro del controlador E/S que lo supervisa. Éste genera una interrupción que al ser atendida por el manejador correspondiente provoca que se despierte al driver del teclado.

El driver extrae del registro de E/S el número de tecla y lo traduce en un carácter usando unos mapas de teclas, también llamados páginas de códigos, que mantiene el SO en función del idioma que haya elegido el usuario al instalar el SO. Suele tener un

tamaño de 8 bits, el más significativo se utiliza para indicar si la tecla ha sido presionada o no.

El driver también tiene que llevar la cuenta de las teclas que han sido presionadas y siguen presionadas.

Una vez establecido el carácter o la combinación de teclas pulsadas el driver almacena dicha información en un buffer para que sea leída y procesada por el proceso adecuado.

8.30. Explicar la gestión de la entrada del ratón.

Cada vez que el usuario desplaza el ratón o pulsa un botón del mismo se transmite un mensaje al registro del controlador de E/S que lo supervisa. Suele ser de 3 bytes y contiene la siguiente información: eje “x”, eje “y” y estado (pulsado o sin pulsar) en los diferentes botones del ratón. Los desplazamientos son relativos, es decir, se miden en relación a la posición del ratón indicada en el último mensaje enviado.

Cada vez que recibe un mensaje, el controlador de E/S que supervisa al ratón genera una interrupción, el manejador de la interrupción despierta el driver del ratón que lee el mensaje y lo coloca en una cola de mensajes para que sea leído y procesado por el proceso adecuado.

Preguntas de autoevaluación tema 9

9.1. ¿Qué es el subsistema de gestión de archivos?

El SO es el responsable de la gestión de los archivos de diferentes sistemas de archivos existentes en la memoria secundaria. Además debe fijar los tipos, atributos, la estructura interna y los mecanismos de acceso que soporta. También debe definir la estructura de los directorios en la que los usuarios y aplicaciones organizan sus archivos, así como mecanismos de búsqueda de archivos y directorios. Por otra parte el SO se encarga de asignar espacio a los archivos y de administrar el espacio libre en la memoria secundaria.

Al componente del SO que realiza todas estas tareas se le denomina subsistema de gestión de archivos.

9.2. ¿Qué es un archivo?