

# ISO P6

## ▼ 1

a) Los dispositivos orientados a bloques o dispositivos de bloque se caracterizan por ser de acceso aleatorio. La unidad mínima de transferencia es un bloque, el cual es escrito o leído independientemente de los demás.

Los dispositivos orientados a flujo o dispositivos de carácter transfieren flujos de caracteres. No utilizan direcciones ni tienen una operación de búsqueda,

b) Orientados a bloques: discos duros, disquetes, CDROMS, etc

Orientados a flujo: ratón, teclado, terminal de texto, cinta magnética, etc.

c)

## ▼ 2

La técnica de E/S programada consiste en el intercambio de bloques de datos entre la CPU y el módulo de E/S. La CPU tiene el control directo sobre la operación en el módulo. Si el periférico está preparado, realiza la transferencia de datos, y mientras lo hace, nunca interrumpe al CPU para informar el estado de la operación, por lo que la CPU debe estar constantemente comprobando si la operación terminó o no. Esto provoca ineficiencia en las operaciones

La técnica de E/S dirigida por interrupciones tiene un funcionamiento similar a la técnica anterior, sin embargo la principal diferencia es que la CPU no debe esperar a que termine la tarea de E/S, debido a que funciona a través de interrupciones. El sistema comprueba si se quiere realizar una interrupción al final de cada proceso y si la halla, la atiende. Para atenderla, debe guardar el contexto del programa actual y luego volver a tomarlo cuando la interrupción finalice. En esta técnica, el CPU puede realizar otras operaciones mientras se prepara la operación de E/S.

La técnica de DMA funciona a través de un módulo capaz de imitar funciones de la CPU, hasta de recibir el control del sistema. Cuando la CPU desea leer o escribir un bloque de datos, lo hace a través del módulo DMA. Esto le permite a la CPU continuar con otro trabajo, debido a que le cedió la operación de E/S al módulo, solamente actuando al principio y final de la operación. Como la operación nunca pasa por la CPU, no es necesario que esta actúe en ningún momento guardando el contexto o interviniendo.

## ▼ 3

La técnica de E/S mapeada en memoria usa el mismo bus de direcciones para memoria y dispositivos de E/S. Las instrucciones de la CPU son usadas para acceder a memoria y también para acceder a dispositivos.

La técnica de E/S aislada o independiente usa un tipo especial de instrucciones de la CPU para implementar E/S, como lo pueden ser instrucciones IN-OUT

## ▼ 4

El objetivo es conseguir que se administren los dispositivos de E/S con eficiencia, dado que son más lentos con respecto a la CPU y la memoria.

Para ello hará uso de la multiprogramación, planificación de requerimientos de los dispositivos, almacenamiento de los datos en memoria mientras son transferidos (o "Buffering"), mantenimiento en memoria de copias de los datos de reciente acceso (Caching), administración de la cola de requerimientos (Spooling), reserva de dispositivos y manejo de errores.

▼ 5

a) Los Drives o controladores contienen el código dependiente del dispositivo. Traducen los requerimientos abstractos en comandos para el dispositivo. Actúan como interfaz entre el SO y el HW. Se cargan en el espacio de memoria del Kernel como módulos.

b) Deben proveer funciones `open()`, `close()`, `read()`, `write()`, etc. Funciones especificadas por el SO.

▼ 7-8

Se lee un archivo en un disco, se determina cual es el dispositivo que almacena los datos. Luego, se traduce el requerimiento abstracto en bloques de disco. Se realiza una lectura física de los bloques en la memoria, marcando los datos como disponibles al proceso que realizó el requerimiento (se desbloquean los datos). Por último se retorna el control al proceso.



Tomado del pwp de E/S. Tremenda paja realizar un ejemplo

▼ 9

Planificación, buffering, caching, spooling? Me voy a enterar que son cuando vea los videos de teoria.

▼ 10

Un disco duro consta de platos, que son las unidades de discos,

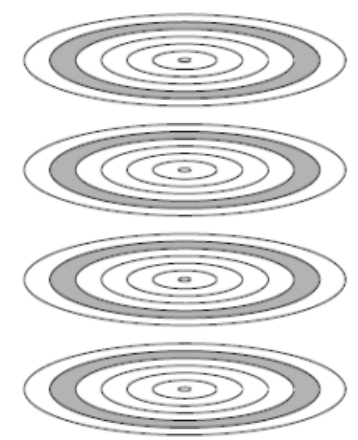
los discos tienen superficies o caras, que son los lados de un plato

Las superficies o caras tienen pistas

Las pistas se dividen en sectores. Las pistas están separadas por separadores de pistas

Los sectores tienen las direcciones de datos. Los sectores se dividen por separadores de sectores. Típicamente un sector ocupa 512 bytes.

Un cilindro N consiste en todas las pistas n de todas las caras. Alineación vertical. No se si la numeración es hacia adentro o hacia afuera



Existe un cabezal por cada cara

▼ 11

Seek time: es el tiempo que tarda el cabezal en posicionarse en la cabeza del cilindro

Latency time: es el tiempo que sucede desde que se posiciona la cabeza en el cilindro hasta pasa por él el sector que se busca

Transfer time: es el tiempo de transferencia del sector (bloque) del disco a la memoria

#### ▼ 14

SSTF, SCAN, LOOK, C-SCAN y C-LOOK puede producir inanición de requerimientos

#### ▼ 17

Un método de asignación de espacio en disco determina la manera en que el SO controla los lugares ocupados del disco ocupados por cada archivo de datos. Se controla la identificación del archivo y sector de inicio y final.

Asignación Contigua, Ligada e indexada son métodos (teóricos) de asignación de espacio en discos.

Asignación contigua consiste en asignar el espacio en disco de manera tal que las direcciones de todos los bloques que corresponden a un archivo se ubiquen de forma lineal.

Ventajas: la cabeza de lectura no debe moverse mucho

Desventajas: fragmentación externa

Asignación ligada consiste en que cada archivo es una lista ligada de bloques de disco. Existe un apuntador al bloque de inicio y uno al bloque de fin para cada archivo, y en cada bloque se halla un apuntador al siguiente bloque de la lista. (Básicamente un lista enlazada de bloques)

Ventajas: no produce fragmentación externa.

Desventajas: puede haber mucho movimiento del cabezal. Si se pierde la referencia al siguiente se perderá el archivo completo. Ineficiente para implementación de archivos directos. Campo extra para apuntador.

Asignación indexada funciona como la ligada pero en vez de tener un apuntador en cada bloque y que se apunten a modo de lista, los punteros se hallan en una sola localidad: un bloque índice.

De esta manera, cada archivo posee su bloque índice, el cual es un arreglo de direcciones de bloques a disco.

Ventajas: no produce fragmentación externa. Eficiente para implementación de archivos directos.

Desventajas: desperdicio cuando los archivos son chicos. Desbordamiento de bloque índice.

#### ▼ 18

Tabla de bits (o vector de bits) consiste en un arreglo de bits en el que cada uno representa un sector del disco. El bit 0 determina que no se halla ocupado y el bit 1 que el sector está ocupado.

Lista ligada consiste en una cabecera que posee la dirección del primer sector vacío, el cual tiene la dirección del siguiente, y así sucesivamente, creando una lista enlazada de sectores vacíos. La lista termina cuando un sector no apunta a nada.

Por agrupación es similar a lista ligada. La diferencia es que en vez de que un bloque apunte a un sólo sector, apunta a varios espacios vacíos.

Por contador consiste en que se apunta a conjuntos de bloques contiguos que estén vacíos, y cada apuntador tiene un número de inicio y tamaño del grupo de sectores vacíos.

Faltan ventajas y desventajas.

▼ 19

a) 32 direcciones x bloque

b)  $10 * 1 + 32 + 1024 + 32768 \text{ (Kib)} = 33834 \text{ Kib} = 4229.25 \text{ KiB} = 4,13 \text{ GiB}$

▼ 20

a)

$\text{nro bloque} = ((\text{nro de inodo} - 1) / \text{nro. de inodos por bloque}) + \text{bloque de comienzo de la lista de inodos.}$

Para bloques de 8 inodos:

Inodo 8  $\rightarrow [(8-1/8) + 2 = 2$

Inodo 9  $\rightarrow [(9-1/2) + 2 = 3$

Para bloques de 16 inodos

Inodo 8  $\rightarrow \text{bloque } 2$

Inodo 9  $\rightarrow \text{bloque } 9$

b)

$\text{Desplazamiento del inodo en el bloque} = ((\text{nro de inodo} - 1) \text{ MOD } (\text{número de inodos por bloque})) * \text{medida de inodo del disco.}$

▼ 21

hdparm se usa para manejar dispositivos de almacenamiento (discos) en Linux/GNU

▼ 22

16383 cilindros

16 cabezas

63 sectores

No sé que pasa.

▼ 24

Lo hice con sudo porque me tiraba error de permisos y con 20 en vez de 500 porque tiraba error

▼ 25

Parece ser que `—offset -t` inicia un contador

`—direct` asigna una velocidad específica de transferencia, pero sin pasar por la memoria caché del buffer del disco duro, por lo tanto, leyendo directamente del disco

La combinacion se utiliza para monitorear la velocidad del disco (`/dev/sda`)

▼ 26

```
sudo hdparm -t --direct --offset 20 /dev/sda → 1010, 1022, 1044, 1012, 1033
```

⇒ promedio = 1024

```
sudo hdparm -t /dev/sda → 1384, 1376, 1411, 1373, 1424
```

⇒ promedio = 1393

▼ 27

Debe tener algo que ver con los buffers. Probablemente sea para minimizar la frecuencia de acceso al disco.

▼ 28

```
lucas@LucasISO:~$ ls /sys/block/
loop0  loop10  loop12  loop14  loop3  loop5  loop7  loop9  sr0
loop1  loop11  loop13  loop2   loop4  loop6  loop8  sda
```

▼ 29

sys/block/ contiene un link simbólico para cada bloque de dispositivo del sistema. El link se corresponde a un directorio de /sys/devices

▼ 30

Imprimir "None"

▼ 31

"noop" es planificador (scheduler) FIFO.

"deadline" intenta garantizar un tiempo de inicio de servicio para una solicitud.

"CFQ" es el scheduler default. Coloca solicitudes sincronicas en varias colas de procesos y luego asigna intervalos de tiempo para que cada cola acceda al disco. Se asemeja al planificador Round Robin

Anticipatory IO ?

▼ 32

Supongo que asigna "noop" como scheduler del sda pero no tengo permiso para hacerlo.

▼ 33

Supongo que cambian los resultados (imagino que serán peores debido a que FIFO presenta malos tiempos)

▼ 34

No pude hacerlo

Ejercicios con soluciones dudosas: 4, 5, 7-8