



IIC2333 — Sistemas Operativos y Redes — 1/2017
Soluciones Interrogación 2

Lunes 29-Mayo-2017

Duración: 2 horas

SIN CALCULADORA

1. [15p]

- 1.1) **[2p]** Considere un *driver* de disco con un *buffer* de solicitudes de lectura que contiene los cilindros 10, 22, 20, 2, 6, 38. En este periodo no llegan más solicitudes a la cola. El tiempo de búsqueda (*seek time*) es de T msec por cilindro. El brazo del disco se encuentra actualmente en la posición 20. ¿Cuánto demora en moverse el brazo para atender todas las solicitudes con los siguientes algoritmos?

- a) SSTF (*Shortest Seek Time First*)
- b) LOOK (el brazo se está moviendo a los cilindros con mayor número)

R.

- a) Orden de atención: 20, 22, 10, 6, 2, 38.
Desplazamientos: $|20 - 22| + |22 - 10| + |10 - 6| + |6 - 2| + |2 - 38| = 2 + 12 + 4 + 4 + 36 = 58$
Tiempo de desplazamiento: $58 \times T$ msec.
- b) Orden de atención: 20, 22, 38, 10, 6, 2.
Desplazamientos: $|20 - 22| + |22 - 38| + |38 - 10| + |10 - 6| + |6 - 2| = 2 + 16 + 28 + 4 + 4 = 54$
Tiempo de desplazamiento: $54 \times T$ msec.
-0.5p si parte en la dirección incorrecta
-0.5p si busca el extremo del disco antes de regresar (ése es SCAN)

- 1.2) **[3p]** Se tienen 8 discos de 1TB cada uno. Indique cómo ve el sistema operativo la cantidad de unidades de disco disponibles para: RAID 0, RAID 1+0, RAID 5, JBOD. ¿Qué ventaja presenta RAID 1+0 sobre RAID 0+1?

R.

0.5p. RAID 0: 1 disco de 8TB

0.5p. RAID 1+0: 1 disco de 4TB

0.5p. RAID 5: 1 disco de 7TB

0.5p. JBOD: 8 discos de 1TB cada uno.

1p. Ventaja de RAID 1+0 sobre RAID 0+1. En RAID 1+0 pueden fallar hasta $N/2$ discos y, con suerte, el sistema sigue funcionando. En cambio, en RAID 0+1, cuando falla el primer disco, falla un set de mirror, y si falla un segundo disco, entonces falla todo el sistema.

- 1.3) **[6p]** Considere el diseño de un sistema de archivos que contiene estructuras para almacenar directorios, registro de bloques libres, un conjunto de *metadata* para cada archivo, y un esquema de asignación de bloques libres. El sistema de archivos además provee una interfaz de llamadas: `create`, `mkdir`, `read`, `write`, `remove`, `rename`, `copy`

- a) **[1p]** Describa qué información incluiría cada entrada de su directorio.

R.

Al menos debería haber: nombre del archivo, y dirección del primer bloque. Opcionalmente puede haber tipo, tamaño, permisos, y otra *metadata*.

- b) **[2p]** ¿Dónde incluiría la información de *metadata* de cada archivo? Justifique su respuesta.

R.

Puede ser en el directorio, si hay una buena justificación (por ejemplo, porque podría ser más rápido de leer). La mejor respuesta es en el primer bloque del archivo (o bien en el *inodo* si es que suponen un esquema indexado). La justificación es que esta información es propia del archivo y no depende del directorio.

- c) [3p] ¿Qué elementos del diseño del sistema de archivo modificaría para poder soportar *hard links*?
R.

En la metadata del archivo (no en el directorio, y por eso responder “en el directorio” en la pregunta anterior no es muy buena idea), hay que incluir un contador de referencias. Esto es porque al borrar un archivo, hay que liberar sus bloques **solamente** cuando el contador de referencias se haga 0. La única diferencia al momento de crear un *hard link*, es que el número de su *inodo* será el mismo que el del archivo referencia.

1p. Mencionar que se necesita un contador de referencias.

1p. Mencionar que hay que poner atención al borrar un archivo.

1p. Mencionar que se debe modificar la metadata del archivo.

- 1.4) [4p] Para mantener el registro de bloques libres se suelen usar dos métodos: (1) una lista que almacena las direcciones de todos los bloques libres, ó (2) un *bitmap* con un bit por cada bloque, donde 1 representa un bloque ocupado, y un 0 representa un bloque libre. Suponga que se tiene un disco con B bloques, de los cuales L están libres, y cada dirección de un bloque utiliza D bits.

- a) [2p] ¿Bajo qué condiciones precisas el método (1) utiliza menos espacio que el método (2)?

R.

El método (1) usa $D \times L$ bits.

El método (2) **siempre** usa B bits.

La condición para que (1) use menos espacio que (2) es $DL < B$, o bien $L < \frac{B}{D}$

- b) [2p] Suponga que $D = 16$. ¿Qué porcentaje del disco debe estar libre?

R.

Para que se cumpla que (1) use menos espacio que (2), debe cumplirse que $L < \frac{B}{16}$. Esto es, que haya menos de un 16avo del disco libre.

2. [15p] Para esta pregunta, considere un sistema de archivos que utiliza bloques de disco de B KB, y punteros a bloques de disco de 32-bit. El sistema de archivo utiliza el modelo *multilevel indexed* en que un *inodo* contiene M byte ($M < B$) destinado a información de *metadata* del archivo (tamaño, nombre, permisos, etc...). El *inodo* también contiene 16 punteros directos a bloques de datos, 1 puntero de indirección simple, 1 puntero de indirección doble, y 1 puntero de indirección triple.

- 2.1) [2p] De acuerdo a la descripción, ¿cuánto es el espacio **no utilizado** de un *inodo*?

R.

El *inodo* ocupa un bloque (B KB). Dentro del *inodo* hay M de metadata, y $16 + 3 = 19$ punteros, cada uno de 4 byte. El espacio no ocupado del *inodo* es $1024B - M - 76$ byte.

0,5p por considerar metadata

1p por considerar punteros

0,5p por plantear la diferencia

Máximo 1,5p si hay errores de unidades

- 2.2) [4p] De acuerdo a la descripción, ¿cuál es el tamaño máximo, solamente considerando el contenido (datos), que puede tener un archivo en este sistema de archivos?

R.

Con los punteros directos, se puede alcanzar hasta $16B$ KB.

En un bloque de B KB, caben $1024B/4 = 256B$ punteros de 4 byte. Con el puntero indirecto simple, se puede alcanzar $256B$ bloques, cada uno de B KB. Se puede direccionar $256B^2$ KB.

Con el puntero indirecto doble, se puede alcanzar $256B \times 256B = 2^{16}B^2$ punteros. Se puede direccionar $2^{16}B^2 \times BKB = 64B^3MB$

Con el puntero indirecto triple, se puede alcanzar $256B \times 256B \times 256B = 2^{24}B^3$ punteros. Se puede direccionar $2^{24}B^3 \times BKB = 16B^4GB$.

El tamaño máximo de archivo es $(16 + 256B)BKB + 64B^3MB + 16B^4GB$

0,5p por cálculo con punteros directos

1p por cálculo con punteros indirecto simple

1p por cálculo con puntero indirecto doble

1p por cálculo con puntero indirecto triple

0,5p por plantear la suma total.

- 2.3) [3p] Suponga que $B = 2$. ¿Qué tamaño ocupa en disco un archivo de 20KB de datos?

R.

Con los 16 punteros directos se puede alcanzar $16 \times 2KB = 32KB$, por lo tanto basta usar los punteros directos.

El único bloque que no contiene datos es el *inodo* (2KB). Por lo tanto, el archivo utiliza 22KB en disco.

0p si responde 20KB

Máximo 2p si hay errores de cálculo, pero el razonamiento está correcto.

- 2.4) [3p] Suponga que $B = 2$. ¿Qué tamaño ocupa en disco un archivo de 20MB de datos?

R.

Con los 16 punteros directos, el puntero indirecto simple, y el puntero indirecto doble, se puede alcanzar $32KB + 1MB + 512MB$.

En el segundo nivel se necesitan $20MB - 1MB - 32KB = (19 \times 2^{10} - 2^5)KB$. Esto equivale a $(19 \times 2^{10} - 2^5)/2 = 19 \times 2^9 - 2^4$ bloques. Cada bloque almacena $512 = 2^9$ punteros. La cantidad de bloques de punteros que se necesita en este nivel es $(19 \times 2^9 - 2^4)/2^9 = 19 - 2^{-5}$. Se necesitan 19 bloques en el segundo nivel.

Por lo tanto, la cantidad de bloques adicionales es: 1, para el *inodo*, 1 para el puntero de indirección simple, y $1 + 19 = 20$ para el puntero de indirección doble. En total se necesitan 22 bloques más, de 2KB cada uno, por lo tanto son 44KB adicionales.

El archivo ocupa, en disco $20MB + 44KB$.

0p si responde 1MB

Máximo 2p si hay errores de cálculo, pero el razonamiento está correcto.

- 2.5) [3p] Suponga que $B = 1$. Si es la primera apertura de un archivo de 1GB (esto es, no hay ninguna información sobre este archivo ni en memoria principal, ni en caché), ¿cuántos bloques deben ser leídos desde el disco para acceder al último byte de este archivo?

R.

Con $B = 1$. Con punteros directos se puede almacenar 16KB. Con un nivel de indirección se puede almacenar 256KB. Con dos niveles de indirección se puede almacenar 64MB. Con tres niveles de indirección se puede almacenar 16GB.

Se necesita, entonces, acceder a un bloque en el tercer nivel. Para leer ese bloque de datos se debe leer: 1 bloque con el *inodo*, 1 bloque de nivel 1, 1 bloque de nivel 2, 1 un bloque de nivel 3, y el bloque de datos. No es necesario leer otros bloques, ya que la asignación indexada permite acceder a una parte aleatoria del archivo.

Se deben leer 5 bloques. Uno para *inodo*, y 3 bloques de punteros, y el bloque de datos.

1p por determinar que se debe llegar al nivel 3.

Máximo 2p Si hay error de un bloque.

Máximo 1p si llega a un resultado leyendo todos los bloques anteriores.

3. [18p] Redes

3.1) [3p] Tanto la conexión *dial-up* como la conexión *DSL* utilizan la infraestructura telefónica para transmitir datos.

a) [1.5p] ¿Por qué la conexión *DSL* alcanza velocidades mayores?

R.

La conexión *DSL* (Digital Subscriber Line) usa frecuencias mayores a las que usa la comunicación telefónica (voz), para transmitir datos en bandas de frecuencia distintas.

1p por mencionar el uso de frecuencias superiores

b) [1.5p] ¿Por qué la conexión *DSL* suele utilizar un modo asimétrico de transmisión (ADSL)?

R.

Se espera que el comportamiento normal del usuario domiciliario sea bajar más datos de los que sube. Por ello tiene sentido asignar mayor ancho de banda para transmitir datos hacia el usuario (*downstream*), que hacia el proveedor (*upstream*).

1p por mencionar que el ancho de banda se asigna más amplio en bajada que en subida

0,5p por mencionar que el comportamiento del usuario es mayor en bajada que en subida

3.2) [3p] Describa 3 diferencias importantes (y diferentes) entre un *switch* y un *router*.

R.

Algunas alternativas. 1p por cada una.

- *Switch* es de capa 2, *router* es de capa 3.
- *Switch* no es programable, *router* sí
- *Switch* usa tabla de MAC + puerta, *router* usa tabla de *forward*
- *Switch* usa direcciones MAC, *router* usa direcciones IP
- *Switch* se configura automáticamente, *router* no.

Decir que *switch* o *router* usan *broadcast* es incorrecto. Ése es el *hub*.

3.3) [4p] Explique cómo se comporta una red de *packet switching* versus una red de *circuit switching* cuando se intenta efectuar una alta cantidad de conexiones simultáneas.

R.

2p. Red de *packet switching* encola paquetes en cada nodo intermedio (*router*). Ante congestión puede enrutar paquetes por salidas alternativas si un enlace está congestionado. Maneja el tráfico de manera flexible. Si recibe más paquetes que los que puede almacenar en sus colas, los paquetes se descartan.

2p. Red de *circuit switching* establece camino entre origen y destino, y reserva ancho de banda. Ante congestión rechaza iniciar conexiones nuevas. Cantidad de conexiones simultáneas es limitada.

3.4) [8p] Para los siguientes problemas de la comunicación en red, indique (no justificar) a qué capa del modelo internet (TCP/IP) corresponde. Puede utilizar la numeración de capas OSI (1,2,3,4,7), o bien su nombre.

R. 1p cada uno. Basta indicar el nombre o el número correcto. —0,5p si se refieren a aplicación como capa 5, o si usan otra numeración.

a) Solicitar fragmentos perdidos de un mensajes transmitido desde el *host A* hacia el *host B*

R. Capa de transporte (4). Ejemplo de TCP.

b) Utilizar un formato específico de mensaje para enviar una solicitud desde *host A* y *host B*

R. Capa de aplicación (7). No es transporte porque habla de **mensajes** y TCP/UDP transmiten **segmentos**. No es una capa inferior (1,2,3) porque se refiere a solicitud entre *hosts* de origen y destino, y no a nodos intermedios.

c) Enviar un grupo de bits desde *host A* a un *switch* de la LAN a la que pertenece.

R. Capa de enlace (*data-link*, 2).

d) Intercambiar tablas de *routing* de manera de calcular las rutas más cortas entre *host A* y *host B*

R. Capa de red (*network*, 3)

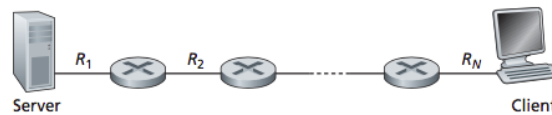
e) Determinar si el nodo de destino está disponible en la red.

R. Capa de red (*network*, 3)

- f) Enviar un archivo de video a través de la red desde *host A* hacia *host B*
R. Capa de aplicación (7). Se trata de envío de archivos completos, como lo hacen HTTP, FTP, torrents, etc. Puede interpretarse también como transporte (4) si se considera que se debe tomar el archivo, dividirlo, transmitirlo y reconstruirlo.
- g) Enviar una señal via par trenzado (UTP) a través de una interfaz de red (NIC)
R. Capa física (1). No es capa de enlace porque no hay semántica en la señal. Capa de enlace otorga un significado a una secuencia de señales.
- h) Serializar un objeto en el lenguaje de programación para transmitir por la red y decodificarlo apropiadamente en el receptor.
R. Capa de aplicación (7). En el modelo OSI se podría haber interpretado en capa 5 o 6, pero en el modelo TCP/IP esto es responsabilidad de la aplicación.

4. [12p] Redes II

- 4.1) Considere la transmisión de un mensaje desde el nodo *servidor* al nodo *cliente* a través de N enlaces con tasas de transmisión R_1, R_2, \dots, R_N medidas en Mbps, como en la figura:



- a) [4p] ¿Cuál es la tasa de transmisión máxima que se puede obtener con una red de *circuit switching*? Los R_i no son necesariamente iguales.
R. La tasa de transmisión está limitada por el enlace más lento: $R_m = \min R_1, R_2, \dots, R_N$. En la red de *circuit switching* todos los enlaces que son parte del circuito transmiten a la misma tasa: R_m .
- b) [4p] Suponga que se usa una red de *circuit switching*, y que $R_1 = R_2 = \dots R_N = R$. ¿Cuántas conexiones se pueden establecer simultáneamente si cada una requiere una calidad de servicio mínima de C Mbps? ($C < R$)
R. Si todos los enlaces son iguales, y se requiere C Mbps por cada conexión, pueden caber hasta $\lfloor R/C \rfloor$. Máximo 3p si no considera el redondeo hacia abajo.
- c) [4p] Suponga que se usa una red de *packet switching*, y que los R_i no son necesariamente iguales. ¿Cuál es la ~~la tasa máxima~~ el tiempo de transferencia para transferir F bits? No considere las demoras de almacenamiento y reenvío.
R. En cada enlace el tiempo de transferencia es F/R_i . En un red de *store-and-forward*, cada paquete debe ser recibido completamente antes de enviarlo por el siguiente enlace. El tiempo total de transmisión es

$$\sum_i^N \frac{F}{R_i}$$