dit
UPM

SOLUCIÓN AL EXAMEN DE
SISTEMAS OPERATIVOS (DISTRIBUIDOS)
6 de Septiembre de 2001
PRIMERA PARTE: sin libros ni apuntes, DURACIÓN: 70 minutos
Respete el espacio especificado para cada respuesta.

Pregunta 1 (*1 punto*)

Explicar de manera informal — si se prefiere, se puede dar seudocódigo — pero en todo caso lo más precisa posible, el funcionamiento del algoritmo de la *instantánea* de Chandy y Lamport.

El algoritmo comienza con un proceso determinado, el que quiere tomar la instantánea, enviándose a sí mismo un "mensaje de instantánea"(MDI). Y para todo proceso, la recepción de un primer MDI supone:

- *La interrupción de la computación de aplicación*
- *El almacenamiento del estado actual*
- *y la Iniciación a vacío del estado de los conjuntos que van a representar el estado de cada uno de los canales de comunicación de entrada.*

A continuación se reanuda la computación de aplicación y, para cada uno de los canales de entrada, y mientras no se reciba un MDI por él, se guarda copia de todo mensaje recibido en el conjunto que representa el estado de ese canal en la instantánea.

Por último, cuando se ha recibido un MDI por todos los canales de entrada, se envía el estado almacenado del proceso, junto con el de todos sus canales de entrada, al proceso que solicitó la instantánea.

Pregunta 2 (*1 punto*)

¿Cuáles son las razones por las que se utilizan hebras y no directamente procesos sin más?

La razón es básicamente histórica. El uso del sistema operativo UNIX y derivados para aplicaciones con alto nivel de concurrencia (y paralelismo en algunos casos) hizo aconsejable disponer de un mecanismo software similar al proceso, pero más eficiente que éste —en tiempos de multiplexación, creación y destrucción, y en uso de memoria para sus descriptores—. El resultado, llamado hebra, es básicamente una multiplexación de "segundo nivel"del proceso, donde todas las hebras de un mismo proceso comparten el mismo espacio de memoria central, y así se consigue acelerar mucho la multiplexación entre hebras de un mismo proceso.

Pregunta 3 (*1 punto*)

¿Se puede decir que AFS es un sistema de ficheros sin estado? Justifíquelo.

No lo es porque el servidor debe saber qué clientes tienen copias de los ficheros, para anularlas cuando se cierra un fichero modificado. Al arrancar debe recuperar esa información preguntándosela a los clientes.

Pregunta 4 (*1 punto*)

En el algoritmo de Cristian de ajuste de relojes la petición lleva la hora del demandante, que es incorporada en la respuesta. ¿Por qué se hace, si el que hizo la petición puede saber a qué hora la envió?

> *Porque las peticiones son datagramas que se pueden perder. Mandando la hora y repitiéndola el servidor, tenemos la seguridad de a qué petición corresponde la respuesta y podemos calcular fácilmente el tiempo de ida y vuelta.*

Pregunta 5 (*1 punto*)

Describa brevemente los métodos que conozca para ubicar el propietario de una página en un sistema de memoria global distribuida que sea escalable a gran número de máquinas (no es necesario que sea tolerante a fallos).

> *Es escalable si la búsqueda del propieterio no tiene un gestor central, ya que sería un cuello de botella, ni utiliza multienvío, ya que las preguntas molestan a todos los procesadores.*
>
> *Una posibilidad es que la gestión de las páginas se asigne, por medio de una función determinista del número de página, a un procesador (por ejemplo, tomando como procesador el resto de dividir el número de página por el número de procesadores).*
>
> *Otra posibilidad es que cada procesador mantenga para cada página un* propietario probable*, mantenido por medio de heurísticos. De este modo, si el propietario probable resulta ser el verdadero, ni siquiera hay que pasar por un gestor. Si resulta no serlo, es muy probable que con el intercambio de unos pocos mensajes se localice.*

Pregunta 6 (*1 punto*)

Diga las diferencias entre un microkernel y un kernel monolítico.

> *Un kernel monolítico integra la implementación de todos los servicios, usualmente de manera no estructurada, mientras que un microkernel implementa solo las básicas, es decir, normalmente solo gestión de concurrencia básica (hebras) y comunicación (también básica) entre flujos concurrentes. El resto de los servicios se implementan mediante servidores, bien locales o bien remotos.*
>
> *Por lo anterior, un microkernel suele ser mucho más pequeño, versátil y fácil de depurar y mantener que un kernel monolítico, si bien suele también ser menos eficiente (debido a las llamadas a los servidores, ya sean remotos o locales).*

SOLUCIÓN AL EXAMEN DE
SISTEMAS OPERATIVOS (DISTRIBUIDOS)
6 de Septiembre de 2001
SEGUNDA PARTE: pueden utilizarse libros y apuntes, DURACIÓN: 60 minutos
Respete el espacio especificado para cada respuesta.

Pregunta 7

El corte C_s asociado a la hora vectorial VC_s de un suceso s es el corte coherente mínimo que le contiene (a s). Se pide demostrar (semi)formalmente esta afirmación, es decir, demostrar que:

Subpregunta 7.1 (*1 punto*)

C_s es un corte coherente.

Lo demostraremos por contradicción.

Supongamos primero que no es coherente, es decir, supongamos que existe un suceso u en C_s que tiene un predecesor v, $v \to u$, tal que v no está en C_s.

Por definición de hora vectorial, sin embargo, para todo suceso t,

$$(t \to s) \implies (t \in C_s)$$

Y por la regla de transitividad de la relación de precedencia causal,

$$(v \to u) \wedge (u \to s) \implies (v \to s)$$

lo cual implica asimismo que $v \in C_s$ y que se ha alcanzado la contradicción buscada.

Subpregunta 7.2 (*1 punto*)

C_s es el mínimo corte coherente que contiene a s.

Trabajemos también por contradicción, es decir, supongamos que el corte C_s no es el corte coherente mínimo que incluye a s. Habrá entonces un corte C que contiene a s y que no contiene a otro suceso u, sí estando sin embargo u incluído en C_s.

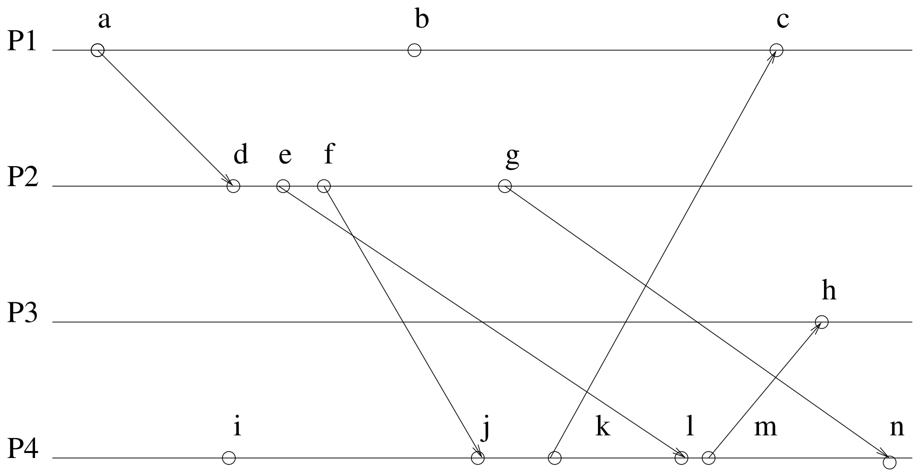
Ahora bien, por definición de hora vectorial,

$$u \in C_s \implies (u \to s)$$

y al tener $s \in C$ y cumplirse también que $u \to s$, por ser C coherente ha de incluir también a u, lo cual alcanza la contradicción buscada.

Pregunta 8

Dado el siguiente diagrama de sucesos y mensajes:



Subpregunta 8.1 (*1 punto*)

Asigne a todos los sucesos siguientes un orden total coherente con la causalidad y realizable por medio de algún algoritmo sencillo, expresando cuál es.

> *Asigno relojes de Lamport, multiplico por 10 y sumo el número de procesador. Luego ordeno por esos valores. Los sucesos, ordenados totalmente según este criterio y con sus relojes entre paréntesis, son:*
>
> $a(11), d(12), i(14), b(21), e(22), f(32), g(42), j(44), k(54), c(61), l(64), m(74), h(83), n(84)$

Subpregunta 8.2 (*0.5 puntos*)

Liste todos los sucesos concurrentes con j.

> *Son g y b.*

Subpregunta 8.3 (*0.5 puntos*)

Liste todos los sucesos que *sucedieron antes que* j.

> *Son a, d, e, f e i.*