dit
UPM
Universidad Politécnica de Madrid
E.T.S.I. Telecomunicación
Departamento de Ingeniería de Sistemas Telemáticos

EXAMEN FINAL EXTRAORDINARIO DE SISTEMAS OPERATIVOS
Convocatoria de Septiembre. 7–9–98
PRIMERA PARTE (SIN LIBROS)

Puntuación y longitud *máxima* —puede de hecho ser más corta— de la respuesta, en cada pregunta. **Una página es una cara**.

DURACIÓN: 70 minutos

Pregunta 1

Dar dos ventajas y una desventaja que tiene un microkernel frente a un kernel clásico.

(*1/2 página; 1 punto*)

El microkernel, al ser más pequeño, consume menos recursos de memoria y, al mismo tiempo, resulta más sencillo de diseñar, comprender, construir y depurar y es, por ello, más fiable. Adicionalmente, al estar basado en gran medida en procesos servidores fuera del kernel, resulta más flexible, es decir, más fácil de reconfigurar.

Pero, sin embargo, el gran número de llamadas al núcleo, cambios de contexto y, muy en especial, llamadas remotas (o "locales") a procedimientos servidores, suelen hacer que los microkernels sean menos eficientes en tiempo de ejecución que sus correspondientes monolíticos.

Pregunta 2

Diga qué es un sistema distribuido asíncrono y explique algún algoritmo de sincronización de relojes que garantice que la diferencia entre dos relojes cualesquiera de un sistema asíncrono está acotada.

(*1/2 página; 1 punto*)

Un sistema asíncrono es aquél en el que no podemos acotar los tiempos de propagación de los mensajes. No existe ningún algoritmo que pueda acotar las diferencias entre dos relojes de un sistema asícrono, a menos que haya algún canal síncrono con una fuente de sincronización, debido a que todos los algoritmos de sistemas asíncronos necesitan intercambiar mensajes.

Muchos algoritmos, como el de Cristian, pueden acotar el error respecto a un reloj local, si descartan mensajes con retardos demasiado grandes. Sin embargo, no se puede garantizar teóricamente que tales mensajes lleguen antes que la deriva haga que el reloj local falsee las medidas. En la práctica, la probabilidad de que las diferencias entre los relojes superen ciertas cotas puede hacerse tan pequeña como se quiera.

Pregunta 3

Tenemos un sistema de memoria global compartida basado en paginación y que mantiene coherencia secuencial invalidando al escribir. Explique paso a paso lo que sucede cuando un proceso intenta escribir en sucesivas variables compartidas que están en la misma página, de la que existe copia local, porque ya se ha leído de ella.

(*1/2 página; 1 punto*)

1. *Inicialmente la página está residente, pero protegida contra escritura.*
2. *Al intentar escribir se produce una violación de permisos.*
3. *El núcleo toma el control e invalida todas demás las réplicas de la página.*
4. *Pone ahora permisos de lectura y escritura.*
5. *Las sucesivas escrituras se realizan normalmente.*

Pregunta 4

Mencione algún algoritmo de exclusión mutua distribuida que utilice relojes lógicos y diga de qué tipo son y para qué los usa.

(*1/2 página; 1 punto*)

El ejemplo explicado en el curso es el de Ricart y Agrawala, que utiliza los relojes lógicos para imponer un orden total acordado en las peticiones del recurso compartido. Usa pues relojes lógicos de Lamport completados con algún mecanismo para hacerlos únicos (poner el número de procesador). La concesión de recurso es distribuída y ha de acordarse en hacerla al primero *que lo pidió.*

Pregunta 5

Enumere y describa brevemente los tipos de multienvío (*multicast*) que conoce.

(*1 página; 2 puntos*)

En la nomenclatura de Colouris, podemos tener:

No fiable *Se transmite sin confirmación, ya sea con los mecanismos de multienvío de bajo nivel u mandando una copia a cada interlocutor.*

Fiable *Realmente* confirmado. *Si el que envía no se cae, todos los participantes vivos entregan el mensaje.*

Atómico *El mensaje lo entregan todos los vivos o ninguno.*

Totalmente ordenado *Los mensajes se entregan en el mismo orden en todas partes, independientemente del orden de envío y de quién envíe.*

FIFO *Los mensajes del mismo emisor se entregan en todos en el orden de envío.*

Causal *Si los envíos están relacionados causalmente según la relación de causalidad potencial, las entregas también lo están. Implica FIFO.*

Combinaciones de los anteriores *Las más importantes son las que combinan atomicidad con distintos tipos de ordenación (total, causal o ambos).*

De sincronización *En combinación con otros tipos de multienvío, imponen un orden global entre las entregas realizadas* antes *y* después *de la entrega del mensaje de sincronización.*

dit
UPM
Universidad Politécnica de Madrid
E.T.S.I. Telecomunicación
Departamento de Ingeniería de Sistemas Telemáticos

EXAMEN FINAL EXTRAORDINARIO DE SISTEMAS OPERATIVOS
Convocatoria de Septiembre. 7–9–98
SEGUNDA PARTE (CON LIBROS)

Puntuación y longitud *máxima* —puede de hecho ser más corta— de la respuesta, en cada pregunta. **Una página es una cara**.
DURACIÓN: 50 minutos

Pregunta 1

Explicar con ayuda de un dibujo cómo puede implementarse un sistema operativo determinado, UNIX por ejemplo, encima de un microkernel (técnica de "el trampolín", por ejemplo). ¿En qué se diferencia esta técnica de la clásica VM (*Virtual Machine*) de IBM?

(3/4 página; 2 puntos)

La implementación (o emulación) de UNIX mencionada permite, en especial, transportabilidad del código máquina ejecutable. Para ello, las llamadas de petición de servicio (instrucciones trap*) se interceptan por el microkernel que, sabiendo de qué se trata, inmediatamente las redirige mediante una "llamada ascendente" o* upcall *(de ahí el nombre de 'trampolín") hacia la biblioteca dinámica correspondiente que (re)implementa el sistema UNIX. El retorno de la llamada puede hacerse otra vez a través del núcleo o directamente desde la biblioteca que emula UNIX al programa de aplicación.*

A diferencia de la técnica anterior, que emula (solamente) las peticiones de servicio al SO, en la técnica VM no se emula ningún servicio de ningún SO, pero, sin embargo, se emulan todas las instrucciones máquina privilegiadas del hardware (que en principio intentan ejecutarse en modo no privilegiado, lo cuál produce un "desvío" al kernel de máquina virtual, que es quién las emula). Por ello, la técnica VM no requiere reimplementar el SO, pero, por contra, resulta más ineficiente que, por ejemplo, la del trampolín.

Pregunta 2

Dados dos sucesos e_i y e_j con horas vectoriales $VC(e_i)$ y $VC(e_j)$, respectivamente, ¿puede ocurrir que sea?
$(VC(e_i)[j] > VC(e_j)[j]) \wedge (VC(e_j)[i] > VC(e_i)[i])$
Si la respuesta es afirmativa, exhibir un ejemplo concreto; si es negativa, razonarla (semi)formalmente en castellano (con algo de notación formal "aquí y allá").

(1/2 página; 2 puntos)

La respuesta es negativa. Lo demostraremos por contradicción.

Supongamos que es cierto. Entonces, por la definición de hora vectorial, $(VC(e_i)[j] > VC(e_j)[j])$ implica que e_j está entre los sucesos que preceden a e_i según la relación de causalidad potencial y, análogamente, $(VC(e_j)[i] > VC(e_i)[i])$ implica también que e_i está entre los sucesos que preceden a e_j. Lo cuál, siendo la relación de casualidad potencial una relación de orden antisimétrica, es una contradicción.
Q.E.D.