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

EXAMEN FINAL EXTRAORDINARIO DE SISTEMAS OPERATIVOS DISTRIBUIDOS
Convocatoria de Septiembre. 7–9–99
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: 1 hora

Pregunta 1

Un sistema de distribuido basado en grupos abiertos de procesos necesita notificar a los miembros de los grupos la entrada o salida de miembros. ¿Qué tipo de multienvío se necesita para esta notificación? Considere los mensajes de notificación y los restantes multienvíos al grupo, ya sea procedentes del exterior como internos.

(*1/3 página; 1 punto*)

El tipo de sincronización es el denominado orden de sincronización *(*sync order*), que requere que en todas las réplicas todos los mensajes se entregen en todos los miembros por igual antes o después de los mensajes de sincronización (incluyendo los mensajes de sincronización). Los mensajes que no sean de sincronización pueden entregarse en cualquier orden (estipulado por la aplicación) dentro de un intervalo de sincronización.*

Pregunta 2

¿Cómo se evita la inversión de tiempo cuando haya que atrasar un reloj adelantado?

(*4 líneas; 1 punto*)

Haciendo correción gradual, bajando su frecuencia hasta que marque la hora verdadera (aproximadamente, depende de los algoritmos empleados).

Pregunta 3

Describir brevemente el principio básico de funcionamiento del algoritmo de la "instantánea" (*snapshot*) de Chandy y Lamport.

(*1/2 página; 1 punto*)

Cada proceso, al recibir el primer mensaje de "tomar instantánea" (TM) por uno de sus canales de entrada, detiene temporalmente su computación, anota el estado de sus variables, marca el estado del canal por donde ha recibido ese TM como vacío, y reenvía (hace echo de) el mensaje TM recibido, por todos sus canales de salida.

A continuación, el proceso anota todos los mensajes que reciba por cada canal de entrada distinto del anterior, y hasta que llegue por ese canal otro mensaje de TM, como parte del estado de ese canal de entrada. Y cuando haya recibido un mensaje de TM por todos sus canales de entrada, el proceso envía a quién haya solicitado la instántanea el estado que ha anotado para sus variable y para sus canales de entrada.

Todo el proceso se desencadena cuando el proceso que desea que se tome la instantánea, fuerza ésta a base de enviarse a sí mismo un mensaje inicial ficticio de TM.

Pregunta 4

Dar con cierto grado de detalle y alguna cifra, si bien aproximada, los factores que explican el coste adicional de realizar un servicio en un proceso servidor remoto, en vez de hacerlo como llamada a un kernel local.

(*1/2 página; 1 punto*)

Al distribuir los servicios del núcleo se pretende que, en la medida de lo posible, esta distribución sea transparente*, es decir, no se note. Desde el punto de vista de la eficiencia en la realización de los distintos servicios del núcleo, hay que conseguir que el tiempo que se invierte en la ejecución de un servicio distribuido sea del mismo orden que en un servicio centralizado. Aunque las redes locales tienen velocidades de transferencia muy altas, comparables en algunos casos a las velocidades de transferencia de la memoria central, sin embargo existen latencias hardware (acceso a la red) y software (preparación de mensajes a enviar y manipulación de mensajes recibidos) que son del orden de milisegundos. Por ello, los servicios que involucran acceso a disco (sistemas de ficheros, por ejemplo), donde ya el acceso al disco supone del orden de 10–20 ms., pueden distribuirse sin que haya degradación apreciable. Este no es actualmente el caso, sin embargo, para otros servicios como los de sincronización, que se sirven en núcleos centralizados en tiempos mucho más cortos (típicamente, en menos de un milisegundo).*

Pregunta 5

Mostrar mediante una tabla de dos columnas el esquema básico de la cache hardware basada en direcciones lógicas. ¿Cuál es la razón de que se use este tipo de cache y no otra basada en direcciones físicas?

(*1/2 página; 1 punto*)

...	...
direccion logica	datos
...	...

Figura 1: Cache hardware basada en direcciones lógicas

La razón de usar este tipo de cache es que la búsqueda en la misma puede hacerse en paralelo con la traducción de la dirección lógica a dirección física —abortando esta traducción, claro está, si la búsqueda en la cache tiene éxito—, y no después de realizar la traducción. De esta manera, se acelera considerablemente la traducción de direcciones por el hardware.

dit
UPM

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

EXAMEN FINAL EXTRAORDINARIO DE SISTEMAS OPERATIVOS DISTRIBUIDOS
Convocatoria de Septiembre. 7–9–99
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: 1 hora

Pregunta 1

Un computador tiene direcciones lógicas de 64 bits y gestión de memoria por paginación con páginas de 32 koctetos. El sistema operativo, que se reserva la mitad superior de las direcciones lógicas y los 4 primeros megaoctetos de la memoria física, simula segmentación (paginada) mediante la técnica de las **regiones**, con segmentos de tamaño máximo de 64 megaoctetos.

Se pide dar, preferentemente en base hexadecimal, una posible configuración de la tabla de páginas (dos columnas, número de página y número de marco) para un programa que está cargado en memoria principal y dispone de tres segmentos de tamaños respectivos 50, 70 y 44 koctetos. Se supone que la memoria física es de 64 megaoctetos.

(*1/2 página; 2 puntos*)

Las páginas (y marcos) tienen, en binario, 2^15 octetos, por lo que el tamaño máximo de segmento, que es, también en binario, 2^26 octetos, corresponde a 2^11 páginas, en hexadecimal 800_H.

El SO ocupa 4 mego actetos, es decir 2^22 ó lo que es lo mismo, $2^7 = 80_H$ marcos.

Una posible configuración de la tabla de páginas pedida es:

Número de página	*Número de marco*
0	*80*
1	*81*
800	*82*
801	*83*
802	*84*
803	*85*
1000	*86*
1001	*87*

Pregunta 2

Tres procesos sin memoria física compartida, comparten las variables a, b y c en un sistema de memoria distribuida compartida. Inicialmente las tres variables valen cero. Los tres procesos ejecutan concurrentemente este código:

P1	P2	P3
a = 1; imprime(b, c);	b = 1; imprime(a, c);	c = 1; imprime(a, b);

Diga cuáles de los siguientes resultados respetan la coherencia secuencial y cuáles la violan, explicándolo:

1. $P1$ imprime 00, $P2$ imprime 00 y $P3$ imprime 00
2. $P1$ imprime 11, $P2$ imprime 11 y $P3$ imprime 11
3. $P1$ imprime 00, $P2$ imprime 10 y $P3$ imprime 01

4. $P1$ imprime 10, $P2$ imprime 10 y $P3$ imprime 11

(1/2 página; 2 puntos)

1. *No la respeta, porque las impresiones se ejecutan antes de las asignaciones.*
2. *Sí la respeta, porque primero todos ven las asignaciones y luego las impresiones*
3. *La salida de $P1$ implica que termina antes de que empiecen $P2$ y $P3$. La salida de $P2$ implica que termina después de empezar $P1$ y antes de empezar $P3$. La salida de $P3$ que éste termine antes de comenzar $P1$, lo que es contradictorio.*
4. *Sí la respeta, ya que la siguiente ejecución secuencial produce ese resultado:*

P1	*P2*	*P3*
a = 1;		
	b = 1;	
	imprime(a, c);	
imprime(b, c);		
		c = 1;
		imprime(a, b);

Pregunta 3

Una máquina sincronizándose con el algoritmo de Cristian recibe un mensaje con dos marcas de tiempo: $12:15:16,892$ y $12:15:17,092$, siendo la segunda la hora del servidor. Si su propio reloj marca las $12:15:17,100$ ¿Cómo debería ajustarlo? ¿Cuál sería el error máximo si el retardo mínimo de un mensaje en la red es de 50 milisegundos y el servidor tarda en responder 8 milisegundos?

(1/4 página; 1 punto)

Debe ajustarlo a $12:15:17,092+(208-8)/2 = 12:15:17,192$ *y el error máximo será de* $(208-8)/2-50 = 50$ *milisegundos.*