dit
UPM

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

SOLUCIONES DEL EXAMEN ORDINARIO DE SISTEMAS OPERATIVOS
Convocatoria de Junio. 27–6–97
PRIMERA PARTE (SIN LIBROS)

DURACIÓN: 1 hora

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

Primera pregunta

Explicar con ayuda de un dibujo la forma de implementar hebras en el sistema Psyche.

(*1/2 página; 1 punto*)

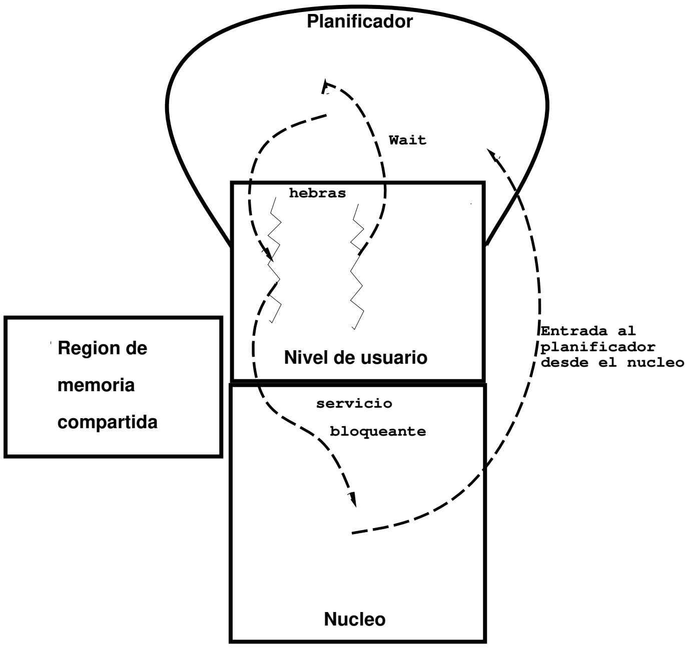

Figura 1: Planificación de hebras en el sistema Psyche

En Psyche la planificación de hebras se realiza en el nivel de usuario, pero con ayudas del núcleo para, por un lado, evitar que todo un proceso se bloquee cuando una de sus hebras realiza una llamada bloqueante al núcleo y, por otro, poder ejecutar hebras de un mismo proceso por más de un procesador simultáneamente (uso de máquinas multiprocesador).

Para lograrlo, el núcleo avisa al planificador de nivel usuario mediante llamadas dirigidas a una dirección bien definida de entrada al planificador en nivel de usuario, de cuándo una hebra inicia una operación de entrada/salida bloqueante y luego de cuándo se completa, así como de cuándo el proceso correspondiente dispone de un procesador virtual más, y de cuándo se le retira (los procesadores virtuales son solo el resultado de multiplexar los procesadores físicos).

Por último, para conseguir más eficiencia, en Psyche la comunicación de datos entre el planificador y el núcleo se realiza a través de una región de memoria compartida por ambos.

Segunda pregunta

Explique brevemente la funciones de los vectores de versión de un sistema de ficheros con replicación optimista.

(*1/2 página; 1 punto*)

A cada objeto replicado se le asocia un vector de versión cuyos componentes reflejan el número de modificaciones que su manejador conoce que se han realizado en cada una de las otras réplicas. Una réplica es dominante sobre otra, y la puede sustituir, si todos los componentes de su vector asociado son mayores o iguales que los correspondientes de la otra. En caso contrario, las réplicas son incoherentes (o inconsistentes), lo que generalmente hace necesaria una intervencin manual.

Tercera pregunta

Explique brevemente todas las propiedades de un multienvío atómico, ordenado y causal.

(*1/4 página; 0.5 puntos*)

Todos los mensajes se entregan a todos los interlocutores vivos o a ninguno, en el mismo orden y respetando la relación de causalidad potencial, es decir, que si un multienvío puede haber influido en otro, en ningún receptor vivo se entrega el primero antes que el segundo.

Cuarta pregunta

Explique por qué, en general, es necesario distinguir entre los sucesos de *recibir un mensaje*, la *llegada de un mensaje* y la *entrega de un mensaje.*

(*1/4 página; 0.5 puntos*)

La operación de recibir *es iniciada por el proceso receptor, independientemente de si el mensaje correspondiente* llega *antes o después. Una vez que ha* llegado *un mensaje, no se* entrega *hasta que se haya iniciado la operación de* recepción *y se cumplan las condiciones de entrega requeridas (orden total, causalidad, atomicidad, etc.).*

Quinta pregunta

Describa brevemente las técnicas usadas por el servicio de nombres de Internet (DNS) para que sea altamente disponible y escalable a millones de nombres.

(*1/2 página; 1 punto*)

La disponibilidad se logra por medio de la replicación de los servidores (especialmente los autorizados) y, en menor medida, de las caches.

La escalabilidad se consigue por medio de la jerquización del espacio de nombres combinada con la descentralización de los servidores autorizados, que funcionan en máquinas distintas. Además contribuyen a la escalabilidad el uso de caches, el uso de protocolos ligeros de consulta y, en menor medida, la replicación (si se eligen los servidores aleatoriamente o en función del tiempo de respuesta).

Sexta pregunta

Decir cuáles son las distintas formas cómo se puede heredar un espacio de direcciones basado en regiones, al crear un nuevo proceso.

(1/2 página; 1 punto)

Cuando en un espacio de direcciones basado en regiones existe herencia, ésta es por regiones, es decir, para cada región se puede decidir si se hereda o no. Si se hereda, existen tres formas de hacerlo:

- *Por* compartición*: La región original y la heredada apuntan exactamente a los mismos marcos de página física. Se necesita usar mecanismos de sincronización para evitar posibles problemas "de carrera".*
- *Por* copia*: La región heredada apunta a marcos de página en los que se ha hecho una copia física de las páginas contenidas en los marcos de la región original.*
- *Por* "copia al escribir"*: Se arranca igual que en la* compartición*, si bien ajustando los bits de protección de las dos tablas de páginas —proceso padre e hijo— para permitir solo lectura. Cuando haya un intento de escritura en una página por parte del proceso hijo o del proceso padre, esto se detecta por el hardware y se produce una desviación al núcleo, el cuál duplica entonces el marco de memoria correspondiente antes de reanudar la ejecución del proceso, que ahora sí puede escribir.*

dit
UPM

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

SOLUCIONES DEL EXAMEN ORDINARIO DE SISTEMAS OPERATIVOS
Convocatoria de Junio. 27–6–97
SEGUNDA PARTE (CON LIBROS)

DURACIÓN: 1 hora

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

Primera pregunta

Tenemos N máquinas idénticas, con una línea de serie de entrada y otra de salida. Se disponen en anillo, conectando la línea de salida de una a la de entrada de la siguiente. Nótese que los mensajes sólo circulan por el anillo en un sentido (incluso las confirmaciones). Se pide:

- Diseñe un algoritmo de sincronización de relojes adaptado a esta topología. Una de las máquinas debe sondear las demás, promediar los relojes obtenidos y ajustar todos a ese promedio.
- Determine el error máximo de las estimaciones que hace el coordinador sobre los relojes de las otras máquinas. Suponga despreciables los tiempos de procesador y que existe una cota mínima del tiempo de propagación de una máquina a otra (τ_{min}).
- Diga qué ocurre si se cae el coordinador y si podemos hacer algo para recuperarnos.

(*1 página; 2 puntos*)

Cuando sea el momento de resincronizar, el coordinador envía su reloj, T_0, junto con una petición de lectura, a su vecino. Cada esclavo, i, que recibe una petición de lectura, lee su reloj, T_i, lo concatena al mensaje recibido y envía el resultado por el otro extremo.

Cuando el coordinador recibe un mensaje de petición de lectura, ya tiene las lecturas de todos. Calcula el retardo total de la consulta, restando el valor del reloj local, T_N, y el primero del mensaje, enviado por él mismo. Considera que el retardo en un enlace es el resultado de dividir el retardo total por N, es decir,

$\tau_{estimado} = \frac{T_N - T_0}{N} = \frac{T}{N}$

Ahora corrige cada valor recibido con el retardo correspondiente a cada máquina, dependiendo de lo alejada que esté de la coordinadora. Puede tomarse como referencia la hora de salida o la hora de llegada del mensaje de consulta. Si se toma la de salida, tenemos

$T_i' = T_i - i \times \tau_{estimado}$

y si se toma la de llegada, tenemos

$T_i' = T_i + (N - i) \times \tau_{estimado}$

Obviamente, para cometer el menor error posible, hay que tomar el primer valor para las primeras máquinas (desde la 1 a la $N/2$) y el segundo valor para las restantes. Por supuesto que si se hace eso, hay que referir todas al mismo instante (el de salida o el de llegada).

Después promedia las medidas corregidas, incluyendo la de la coordinadora (T_N) y excluyendo las demasiado alejadas de las demás, calcula las diferencias entre sus valores y la media, y *se construye un mensaje de ajuste, con la cantidad que cada esclavo debe utilizar para corregir su valor:*

$$\theta_i = \frac{\sum_{j=1}^{j=N} T_j}{N} - T_i$$

Los esclavos reciben el mensaje con las correcciones, extraen y aplican la suya y lo reenvían al siguiente. Por supuesto, el coordinador también hace su propio ajuste.

El error en la determinación de una hora depende de la variabilidad de los retardos y de lo alejada que esté una máquina del coordinador. Como en el algoritmo de Cristian, el caso peor es el de máxima asimetría. Siendo T el tiempo que tarda el mensaje en dar la vuelta, la correción del reloj de la máquina alejada i saltos del coordinador es:

$$\frac{T}{N} \times i$$

Sin embargo, si sólo se hubieran producido retardos mínimos en ese trayecto, la correción real debía ser

$$\tau_{min} \times i$$

Y si en el trayecto se hubiera producido casi todo el retardo, entonces la corrección debería ser

$$T - \tau_{min} \times (N - i)$$

Luego el error es menor que el mayor de los errores posibles:

$$\epsilon \leq Max((\frac{T}{N} - \tau_{min}) \times i, T - \tau_{min} \times (N - i) - \frac{T}{N} \times i)$$

donde el máximo valor de i es $N - 1$ si referimos todas las medidas a un extremo, y la mitad si las referimos a los dos. Para ese valor tendrá lugar el error máximo.

Si se cae el coordinador, como si se cae cualquier máquina, desaparece toda posibilidad de comunicación. No existe recuperación automática posible, a menos que se puentee automáticamente la máquina caída. En este caso se puede hacer una elección en anillo. Para permitir esta posibilidad, cada medida de tiempo en el mensaje de sondeo debe identificar claramente a la máquina a la que pertenece, y prever así que puedan faltar medidas.

Segunda pregunta

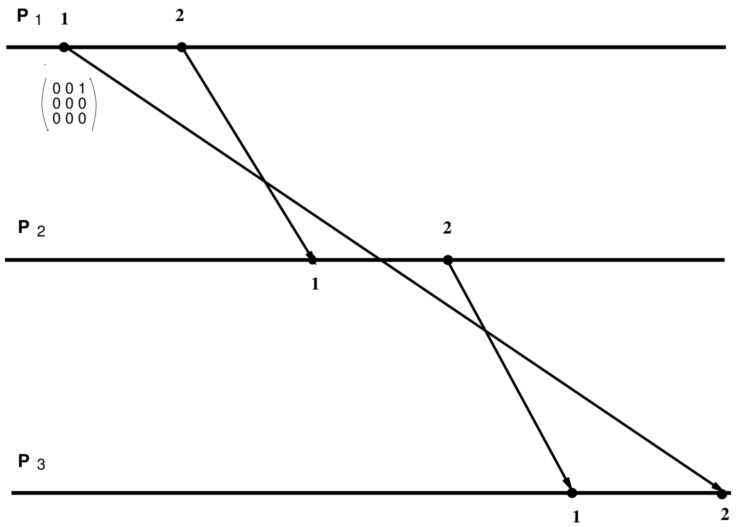

Figura 2: Relojes matriciales

Para poder realizar entrega (potencialmente) causal de mensajes uno a uno (*unicast*), se diseña un sistema de relojes matriciales de dos dimensiones en el que los únicos sucesos de interés son los de envío de mensajes, con cada elemento $[i, j]$ de la matriz significando el número de mensajes enviados desde el proceso i al proceso j. Se pide:

1. Para la computación distribuida que se muestra en la figura, ¿cuál es la hora matricial en todos los puntos marcados para cada proceso?

(*1/2 página; 0,5 puntos*)

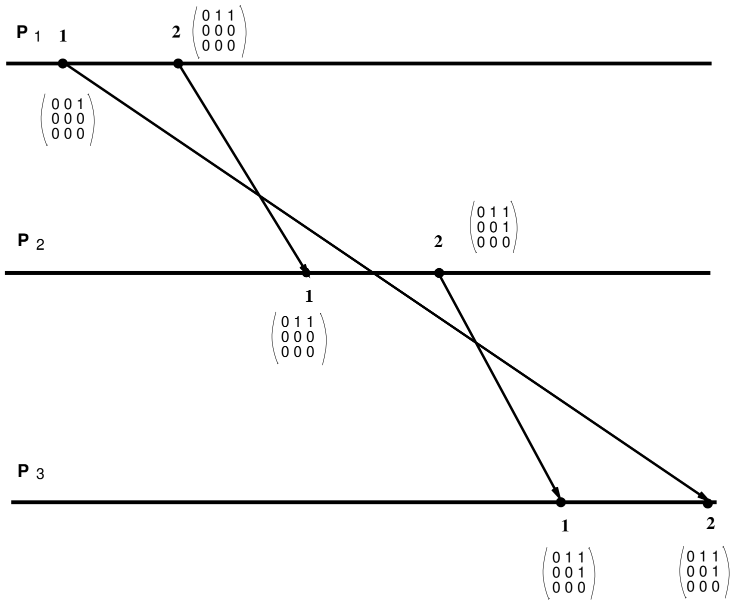

Figura 3: Horas matriciales en todos los sucesos de envío y recepción

Cuando un proceso P_i envia un mensaje a otro proceso P_j, se incrementa en una unidad la componente $MC_i[i,j]$, siendo MC_i el reloj matricial de P_i. El valor actualizado del reloj matricial de P_i se incluye con el mensaje en sí y, al recibirse éste en el proceso P_j, la hora matricial de P_j se actualiza según la regla:

$$(\forall m, n)\ MC_j[m,n] := max(MC_j[m,n], TS[m,n])$$

siendo TS la hora matricial que viene con el mensaje.

2. Definir una regla de entrega causal, que se apoye en los relojes matriciales anteriores.

(1/2 página; 2 puntos)

Con cada proceso P_i asociamos un vector (columna) D_i que cuenta los mensajes provenientes de otros procesos ya Delivered *(entregados) al proceso P_i. Cuando llega al proceso P_i un mensaje con hora matricial TS, se entrega éste sii la columna i-ésima de TS tiene un sólo elemento mayor que el correspondiente de D_i, y es mayor sólo en una unidad. Más concretamente:*

$$(\exists k \textit{ tal que } TS[k,i] = D_i[k] + 1) \wedge ((j \neq k)\ TS[j,i] \leq D_i[j])$$

Si la regla anterior no se cumple, se almacena el mensaje correspondiente hasta que, debido a la entrega de otros mensajes, la regla se cumpla para dicho mensaje, en cuyo momento se entrega éste.

3. Explicar cómo mediante la regla anterior se puede corregir la violación de causalidad que se muestra en la figura.

(1/4 página; 0,5 puntos)

Al llegar a P_3 el mensaje proveniente de P_2, sucede que tanto $TS[1,3]$ como $TS[2,3]$ son mayores en una unidad que, respectivamente, $D_3[1]$ y $D_3[2]$, por lo que ese mensaje se recibe en P_3 pero no se entrega, sino que se almacena. El mensaje de P_1 a P_3, sin embargo, sí cumple la regla de entrega. Al entregarlo, $D_3[1]$ pasa a valer 1, con lo que ya se puede entregar el mensaje de P_2 a P_3 que estaba almacenado, y $D_3[2]$ pasa entonces a valer también 1.