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

EXAMEN FINAL ORDINARIO DE SISTEMAS OPERATIVOS
Convocatoria de Junio. 19–6–98
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: 1 hora

Pregunta 1

Explicar con ayuda de un dibujo el funcionamiento de la técnica de memoria virtual basada en un paginador externo.

(1/2 página; 1 punto)

En el caso de la memoria virtual en un sistema operativo distribuido con paginador externo, todo funciona igual que en la memoria virtual tradicional, con la única excepción de que las peticiones al "disco de intercambio" para traer páginas que han producido un fallo, o escribir páginas "víctimas" sucias, las hace ahora el núcleo comunicándose mediante mensajes con el paginador externo (un proceso de nivel usuario y posiblemente remoto) y no a un disco local. Por lo demás, el núcleo sigue haciendo todo su trabajo igual que antes en la memoria virtual clásica. En particular, detectando fallos de página, realizando la asignación de marcos, eligiendo las víctimas, etc. El paginador externo se comporta, de esa manera, solamente como una especie de "disco remoto de páginas de memoria virtual".

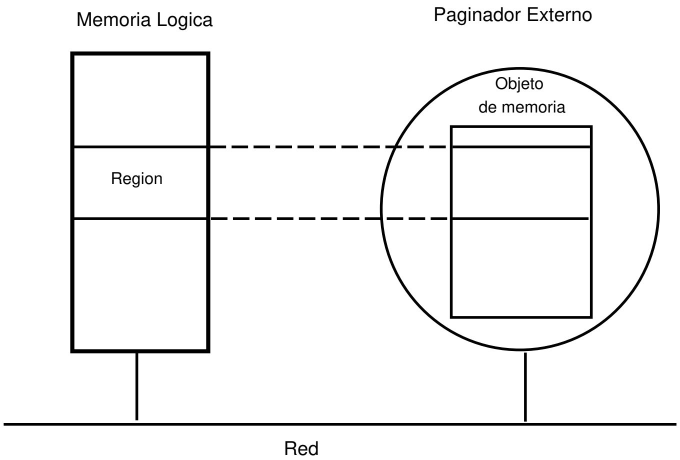

Figura 1: Memoria virtual mediante paginador externo

Pregunta 2

Dar la estructura típica de una credencial (*capability*). ¿Cuál es la razón de que suela incluir los permisos de acceso sin cifrar?

(1/2 página; 1 punto)

Una credencial suele estar formada por tres partes: identificación del recurso, un número aleatorio específico por recurso suficientemente grande —al menos 32 bits— para evitar su adivinación, y los permisos de *las operaciones concedidas sobre el recurso en sí (normalmente, un bit por operación). Para evitar la manipulación fraudulenta de la credencial, los dos últimos campos, número aleatorio y permisos, suelen ir cifrados de manera conjunta, bien mediante técnicas de cifrado simétrico o, algo frecuente, mediante una función de un solo sentido (sin inversa conocida). En este segundo caso de función de un solo sentido, se suelen incluir también en la credencial los bits de permisos sin cifrar, para facilitar, haciéndola muchísimo más eficiente, la comprobación de la validez de las credenciales —$2^N/2$ más eficiente, siendo N el número de operaciones para las que puede haber permisos en la credencial—. Así mismo, la inclusión de los permisos de acceso sin cifrar, permite que los procesos cliente puedan comprobar los permisos de que disponen de una manera directa.*

Pregunta 3

Explicar con ayuda de un dibujo cómo funciona la técnica de optimización de envío de mensajes dentro de una misma máquina, mediante regiones de memoria lógica y "copia en la escritura" (*copy on write*).

(*1/2 página; 1 punto*)

Para trasmisión de mensajes entre procesos con memorias lógicas diferentes, pero dentro del mismo computador, existe una técnica muy eficiente.

Al construir el mensaje a enviar, el proceso emisor lo sitúa a partir de un principio de marco de página. De esa manera, el mensaje puede tratarse como una región formada por un conjunto completo de páginas.

Y para enviar el mensaje al proceso receptor, el núcleo solo necesita crear una nueva región en el espacio de memoria lógica de ese proceso, asociada con los marcos de memoria física donde está situado el mismo, es decir, no se requiere ninguna copia física del mensaje en la memoria, sino solo ajuste de la tabla de paginación.

Por último, para poder compartir tanto el proceso receptor como el proceso emisor, los mismos marcos de memoria de manera segura, los marcos involucrados se colocan en las dos regiones con protección frente a escritura. Y cualquier intento de escibir por uno de los dos procesos en un marco cualquiera, hace que ese marco se duplique, una copia para cada proceso, teniendo ya entonces cada copia permiso de escritura en las tablas de páginas de los dos regiones.

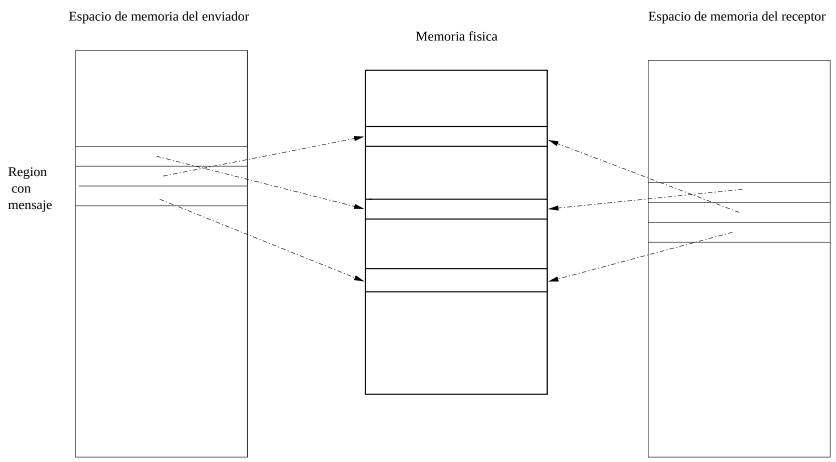

Figura 2: Transmisión de mensajes dentro de un mismo computador

Pregunta 4

Explique por qué, en general, un programa paralelo que funcione en un sistema de memoria compartida distribuida con coherencia de liberación (*release consistency*) o con coherencia de entrada (*entry*) o perezosa (*lazy*), es más eficiente que ejecutando con coherencia secuencial (*sequential consistency*). ¿Puede decir alguna ventaja de la coherencia secuencial?

(*1/2 página; 1 punto*)

Porque sólo es necesario propagar actualizaciones en las regiones críticas, con variables compartidas declaradas, ya sea al salir (release) o al entrar (lazy). De esas actualizaciones sólo es necesario transmitir lo realmente modificado, no páginas enteras. Es fácil evitar la compartición falsa, poniendo en páginas distintas regiones críticas distintas. Declarando modos de compartición se pueden hacer también optimizaciones, como en el caso de que cada proceso modifique elementos distintos de un array.

La coherencia secuencial permite ejecutar programas hechos para multiprocesador sin modificación alguna.

Pregunta 5

Esboce un algoritmo de exclusión mutua en anillo y diga las ventajas e inconvenientes que tiene.

(*1/2 página; 1 punto*)

Los que participan en el algoritmo se van pasando un testigo ordenadamente. Quien tenga el testigo, puede retenerlo mientras accede al recurso compartido. Tiene la ventaja de que es simple, y no sobrecarga a nadie demasiado. Como desventajas tenemos que hay mensajes circulando cuando no se necesita el recurso compartido, que cuantos más participen en el sistema, mayor será el retardo en obtener el recurso y que la cesión del recurso no es FIFO, sino que depende de la posición relativa en el anillo. Problemas comunes con otros algoritmos son la falta de robustez, en este caso por pérdida de testigo, que requiere una elección para regenerar uno sólo.

UPM

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

EXAMEN FINAL ORDINARIO DE SISTEMAS OPERATIVOS
Convocatoria de Junio. 19–6–98
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: 1 hora

Pregunta 1

¿Basta con que los sucesos de la frontera de un corte sean concurrentes dos a dos para que el corte sea coherente? Si la respuesta es negativa, exhibir un contraejemplo; si es afirmativa, razonarla (semi)formalmente en castellano (con algo de notación formal "aquí y allá").

(1/2 página; 2 puntos)

$$e_i \parallel e_j \equiv (VC(e_i)[i] > VC(e_j)[i]) \wedge (VC(e_j)[i] > VC(e_i)[j])$$
$$\Rightarrow$$
$$(VC(e_i)[i] \geq VC(e_j)[i]) \wedge (VC(e_j)[j] \geq VC(e_i)[j]) \equiv e_i, e_j \text{ \textit{coherentes}}$$

Q.E.D.

Pregunta 2

Cierto programa va a abrir un fichero en un sistema de ficheros tipo Coda. El fichero está en volumen replicado cuatro veces. Al ejecutar la operación de abrir, Venus se da cuenta de que no hay copia válida en la cache y pide los vectores de versiones. Éstos son (por orden de réplica):

```
(1156, 1156,  211, 1156)
(1156, 1157,  211, 1156)
( 211,  211,  211,  211)
(1156, 1156,  211, 1156)
```

- Si el servidor preferido al empezar es el cuarto, diga qué operaciones desencadenaría esta apertura y dónde.
- ¿Y si el tercer vector fuera `( 211,  211,  212,  211)`?

(1/2 página; 2 puntos)

- *Como el cuarto no conoce el último cambio, es invalidado como servidor preferido y borrada la cache de todo el volumen. Se elige como preferido el primero, se extrae el fichero de éste y se tratan de actualizar los que tienen copia obsoleta (sumisa), de modo que queden los vectores a (1157, 1157, 1157, 1157).*
- *Hay un conflicto, que ha de resolverse manualmente. El tercer servidor, cuando estaba aislado, ha hecho una modificación que es desconocida para los demás.*

Pregunta 3

Cierto sistema de tiempo real se utiliza para medir y registrar las frecuencias de tráfico en un sistema de autopistas (en vehículos por minuto). En particular, un dato importante a obtener es la frecuencia instantánea máxima. El reloj del sistema se sincroniza a mano, cuando el operador lo pone en hora. Para evitar errores, se utiliza una operación de ajuste gradual de reloj, que no permite variar la frecuencia más de un 1%.

- ¿Cuál será el error relativo máximo aproximado en la estimación de la frecuencia máxima de vehículos?
- ¿Y si no se hiciera ajuste gradual?
- ¿Cuánto tardaría en actualizarse aproximadamente la hora del sistema en un cambio de horario de verano a invierno (o viceversa) si el sistema no tuviera conocimiento de las normas de ahorro de energía?

(1/2 página; 1 punto)

En este ejercicio se ignora la deriva del reloj, que no se proporciona y que suele ser absolutamente despreciable frente al 1%. Suponemos además que medimos leyendo un contador (hardware o software) cada minuto, y que marcamos los minutos contando interrupciones de reloj o pulsos de un contador.

- *El error relativo en el recuento de coches es exactamente el mismo que el de la frecuencia: 1%, ya que el numero de coches por unidad de tiempo es igual al número de coches en un periodo multiplicado por la frecuencia de muestreo (proporcional a la del reloj).*
- *Al no modificar la frecuencia, no se produciría error alguno, ya que la cadencia de interrupciones o pulsos de reloj no varía.*
- *Cien horas.*