

SISTEMAS INFORMÁTICOS II

Asignatura: Grupo:
 Apellidos: Nombre:
 Ejercicio del día: 13 de junio de 2005. Examen final.



3.1	3.2	3.3	3.4	3.5	4.1	4.2	4.3
3							4

Contestar las preguntas de teoría en hojas aparte. Contestar a los problemas en esta hoja de enunciados.

Tiempo para la realización del examen: 3 horas.

1.- TEORÍA (10 puntos). Network Operating System, NOS. Funciones y modo de operación. Transparencia

2.- TEORÍA (10 puntos). Common Object Request Broker Architecture (CORBA). Estructura y descripción básica de los elementos que la componen.

3.- PROBLEMA (10 puntos). Una programa transaccional de reserva de localidades de cine tiene la siguiente estructura básica:

```

Begin Tx
found = false;
i=0;
while not found
    i = i + 1;
    Read Butaca(i) // Lectura de una serie de objetos
    If (Butaca(i).libre = true)
        Butaca(i) = false
        Write Butaca(i)
        found = true
    Imprimir_entrada(Butaca(i))
Commit
    
```

3.1.- (1 punto) Supuesto que Butaca(1).libre = false y Butaca(2).libre = true, escribir las acciones básicas a las que daría lugar este programa transaccional si se ejecutara de manera aislada (nota: contar cada elemento del array Butaca como un objeto, y no reflejar acciones realizadas sobre variables locales del programa).

```

BEGIN
READ Butaca(1)
READ Butaca(2)
WRITE Butaca(2)
COMMIT
    
```

SISTEMAS INFORMÁTICOS II

Asignatura: Grupo:
 Apellidos: Nombre:
 Ejercicio del día: 13 de junio de 2005. Examen final.



3.2.- (2 puntos) En las mismas condiciones del apartado anterior, escribir la historia resultante de la ejecución concurrente de dos transacciones en las que se asignara la misma butaca a ambas. ¿Qué tipo de violación de aislamiento se produce?

Nota: Numerar en secuencia los pasos de cada historia realizada. Comenzar todas las versiones de los objetos en el número 0.

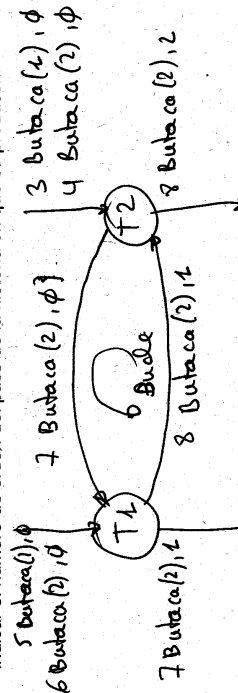
```

1 T1 BEGIN
2 T2 BEGIN
3 T2 READ <BUTACA(1), 0>
4 T2 READ <BUTACA(2), 0>
5 T1 READ <BUTACA(1), 0>
6 T1 READ <BUTACA(2), 0>
7 T1 WRITE <BUTACA(2), 1>
8 T2 WRITE <BUTACA(2), 2>
9 T2 COMMIT
10 T2 COMMIT
    
```

Actualización perdida.

3.3.- (2 puntos). Dibujar el diagrama de dependencias de la historia anterior, y justificar a partir de él la violación de aislamiento que se ha producido.

Nota: Al realizar el diagrama de dependencias, además de identificar los arcos que indican las dependencias con el nombre del objeto y la versión del mismo que produce la dependencia, indicar el número de orden del paso de la historia en que se producen.



La existencia de un bucle en el diagrama de dependencias indica la violación del aislamiento producida en los pasos 7 y 8 de la historia.

3.4.- (1 punto) Convertir la transacción realizada en el apartado 3.1 en bien formada y en dos fases, y hallar su modelo equivalente simple.

Tx BF y 2Fases:

```

BEGIN
SLOCK Butaca(1)
READ Butaca(1)
SLOCK Butaca(2)
READ Butaca(2)
XLOCK Butaca(2)
WRITE Butaca(2)
COMMIT
    
```

Modelo Equivalente Simple:

```

SLOCK Butaca(1)
READ Butaca(1)
SLOCK Butaca(2)
READ Butaca(2)
XLOCK Butaca(2)
WRITE Butaca(2)
UNLOCK Butaca(2)
UNLOCK Butaca(1)
    
```

SISTEMAS INFORMÁTICOS II

Asignatura Grupo
Apellidos Nombre
Ejercicio del día 13 de junio de 2005. Examen final.



3.5.- (4 puntos) Detallar los TxRPCs que se enviarán entre los distintos módulos del sistema de proceso de transacciones (transaction manager, log manager, lock manager, resource manager, contenedor de los registros de butaca y programa transaccional) con motivo de la ejecución de la transacción del apartado 3.4. Suponer que el sistema trabaja siempre con Two Phase Commit, y que el Commit se ejecuta satisfactoriamente.

Presentar el resultado en la siguiente tabla

Acción	TxRPC producido	Módulo origen	Módulo destino	Datos	Resultado
BEGIN	Begin Tx	Pgm Tx	Tx Mgr		TxId
SLOCK Butaca(1)	SLOCK	Pgm Tx	RML	TxId, Butaca(1) (clave)	TxId
	Join Tx	RML	Tx Mgr		
	SLOCK	RML	Lock Mgr	TxId, Butaca(1) (clave)	
READ Butaca(1)	Read	Pgm Tx	RML	TxId + Butaca(1) (clave)	Butaca(1) (clave)
SLOCK Butaca(2)	SLOCK	Pgm Tx	RML	TxId, Butaca(2) (clave)	
	SLOCK	RML	Lock Mgr	TxId, Butaca(2) (clave)	
Read Butaca(1)	Read	Pgm Tx	RML	TxId, clave Butaca(1) (clave)	Butaca(1) (clave)
XLOCK Butaca(2)	XLOCK	Pgm Tx	RML	TxId, Butaca(2) (clave)	
	XLOCK	RML	Lock Mgr	TxId, Butaca(2) (clave)	
WRITE Butaca(1)	Write	Pgm Tx	RML	TxId, Butaca(1) (clave)	
	Write-log	RML	Log Mgr	TxId, Butaca(1) (clave)	
COMMIT	Commit	Pgm Tx	Log Mgr	TxId, Butaca(1) (clave)	
	Prepare to Commit	Pgm Tx	Log Mgr	TxId	
	Ready to Commit	RML	Log Mgr	TxId	
	Write-log	Log Mgr	Log Mgr	TxId, Commit Record	
	Commit	Log Mgr	RML	TxId	
	Unblock	RML	Lock Mgr	TxId, Butaca(2) (clave)	
	Unblock	RML	Lock Mgr	TxId, Butaca(1) (clave)	
	Complete	RML	Tx Mgr	TxId	
	Write-log	Tx Mgr	Log Mgr	TxId, Complete Record	

SISTEMAS INFORMÁTICOS II

Asignatura Grupo
Apellidos Nombre
Ejercicio del día 13 de junio de 2005. Examen final.



4.- PROBLEMA (10 puntos). Se desea diseñar un Directory System Agent X 500, que denominaremos DSA1, en el que se espera recibir, a través del protocolo DAP, un tráfico que se puede suponer de Poisson con un valor medio de 50 peticiones / s.

4.1.- (2 puntos). Suponiendo que el tiempo de servicio del servidor de nombres se encuentra distribuido exponencialmente, calcular la capacidad media que debe tener el servidor, en peticiones por segundo, para que el 95 % de las peticiones que reciba tengan un tiempo total de estancia en el sistema de menos de 10 ms.

El modelo es M/M/1. Se pide un tiempo de estancia dado para el 95% de las consultas, por lo que hay que acudir a la función de distribución acumulada:

$$w(t) = 1 - e^{-(\mu - \lambda)t} \Rightarrow \mu = \lambda - \frac{w(1 - w(t))}{t}$$

En este caso:

$$\lambda = 50 s^{-1} \quad \mu = 50 - \frac{w(1 - 0.95)}{0.01} = 349.57 s^{-1}$$

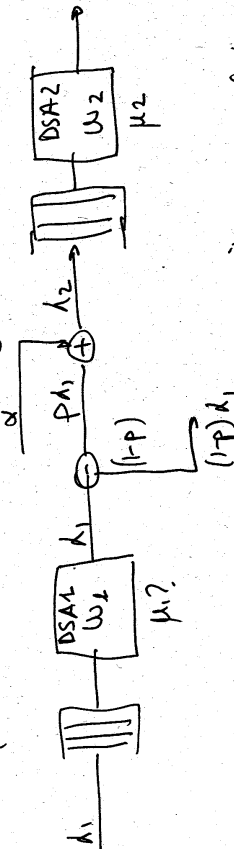
$$t = 10 ms \quad w(t) = 0.95$$

4.2.- (5 puntos). Cuando al procesar una petición recibida DSA1 no tiene los datos necesarios para resolverla, redirige la petición a otro Directory System Agent de nivel superior, que denominaremos DSA2, mediante protocolo DSP. Esto ocurre, en promedio, en un 25% de las peticiones que procesa. Este nuevo servidor contesta directamente al cliente que realizaba la petición.

Se sabe que DSA2 recibe consultas de muchos más servidores, componiendo el total de ellas un tráfico de Poisson con una media de 200 peticiones / s. Su tiempo de servicio, que también se puede considerar distribuido exponencialmente, hace que sea capaz de procesar un promedio de 300 peticiones / s.

Si se desea que el tiempo medio de respuesta de una petición realizada a DSA1, independientemente de quién la procese, sea igual o inferior a 6 ms., determinar cuál sería la capacidad media que debería tener DSA1 para poder cumplir este requerimiento. ¿Qué condición es más restrictiva para el sistema, esta o la establecida por el primer apartado, de modo que la elijamos en nuestro diseño, si DSA1 debe cumplir ambas?

El esquema del sistema es el siguiente:



Se supone en DSA1 condiciones que permitan aplicar el teorema de Burke: $\lambda_1 < \mu_1$

El tiempo medio de estancia en el sistema se calcula considerando que hay dos casos posibles.

- Peticiones atendidas solo por DSA1: $W = W_1$
- Peticiones atendidas por ambos servidores: $W = W_1 + W_2$

Como son conocidas las probabilidades de ambos casos:

$$\bar{W}_T = (1-p)W_1 + p(W_1 + W_2) = \underline{\underline{W_1 + pW_2}}$$

$$W_1 = \bar{W}_T - pW_2 \Rightarrow \frac{1}{\mu_1 - \lambda_1} = \bar{W}_T - p \cdot \frac{1}{\mu_2 - \lambda_2}$$

$$\Rightarrow \underline{\underline{\mu_1 = \lambda_1 + \frac{\lambda_1}{\bar{W}_T - p \cdot \frac{1}{\mu_2 - \lambda_2}}}}}$$

Los datos del problema son:

$$\left\{ \begin{array}{l} \lambda_1 = 50 \\ \lambda_2 = 200 \\ \mu_2 = 300 \\ \bar{W}_T = 6 \mu s \\ p = 0.25 \end{array} \right\} \mu_1 = 50 + \frac{\lambda_1}{0.006 - 0.25 \cdot \frac{1}{300 - 200}} = \underline{\underline{335.715^{-1}}}$$

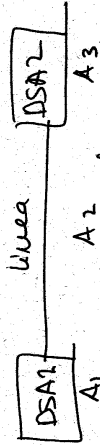
La condición calculada cumple el teorema de Burke.

La condición más restrictiva es aquella que exige un tendimiento mayor del servidor, que es la primera:

$$\mu_1 = 349.575^{-1}$$

Con ella, $W_1 = 3,338 \text{ ms}$, $\underline{\underline{W_T = 5.84 \text{ ms} < 6 \mu s}}$

4.3.- (3 puntos) Ambos servidores se encuentran conectados por una línea de comunicaciones que tiene una disponibilidad de 0.99. La disponibilidad de DSA2 es de 0.98. Si el MTTF del servidor de nombres bajo diseño es de 2000 horas, ¿cuál será el MTTR que deberemos garantizar para él, para conseguir de este modo que la disponibilidad del sistema total sea igual a 0.957? ¿Cuál sería, en este caso, la disponibilidad de DSA1 bajo diseño considerado aislado del resto del sistema?



Para que el sistema funcione correctamente de acuerdo a sus especificaciones deben encontrarse disponibles los tres elementos. Por tanto la disponibilidad del conjunto es el producto de las disponibilidades de sus componentes:

$$A_T = A_1 \cdot A_2 \cdot A_3; \quad A_1 = \frac{A_T}{A_2 \cdot A_3} = \frac{MTTF}{MTTF + MTTR}$$

$$\Rightarrow \underline{\underline{MTTR = \left(\frac{A_2 \cdot A_3}{A_T} - 1 \right) \cdot MTTF}}$$

Con los datos del enunciado:

$$\left\{ \begin{array}{l} A_2 = 0.99 \\ A_3 = 0.98 \\ MTTF = 2000 \end{array} \right\} MTTR = \left(\frac{0.99 \cdot 0.98}{0.957} - 1 \right) 2000 = \underline{\underline{42.53 \text{ h.}}}$$

y la disponibilidad de A1 aislado es, en este caso:

$$A_1 = \frac{MTTF}{MTTF + MTTR} = \frac{2000}{2042.53} = \underline{\underline{0.9792}}$$