

EXAMEN 2do PARCIAL – Bloque Unidades Didácticas 7 a 11 9 de Junio de 2014
Concurrencia y Sistemas Distribuidos Modelo A

APELLIDOS:				NOMBRE:	
DNI:		GRUPO:		FIRMA:	

Este bloque tiene una puntuación máxima de **10 puntos**, que equivalen a 2.5 puntos de la nota final de la asignatura. Indique, para cada una de las siguientes 50 afirmaciones, si éstas son verdaderas (**V**) o falsas (**F**). **Cada respuesta vale: correcta= 0.2, errónea= -0.2, vacía=0.**

Importante: Los primeros **3 errores** no penalizarán, de modo que tendrán una valoración equivalente a la de una respuesta vacía. A partir del 4º error (inclusive), sí se aplicará el decremento por respuesta errónea.

1. Sobre los sistemas distribuidos:

	A.- Todos los sistemas de tiempo real son ejemplos de sistemas distribuidos.
	B.- Algunos tipos de transparencia pueden comprometer la eficiencia del sistema, por ejemplo introduciendo retardos en las interacciones entre componentes de un sistema distribuido.
	C.- Un sistema distribuido ofrece la imagen de un sistema coherente y único.
	D.- Los sistemas distribuidos proporcionan diferentes tipos de transparencia. Entre ellos: transparencia en el rendimiento, en la escalabilidad, en su disponibilidad, en la seguridad...
	E.- Si se desea que un sistema distribuido ofrezca escalabilidad de tamaño, no se debe hacer uso de técnicas de replicación, ya que al actualizar el estado de un componente tendremos que propagar tal actualización a todas las réplicas.

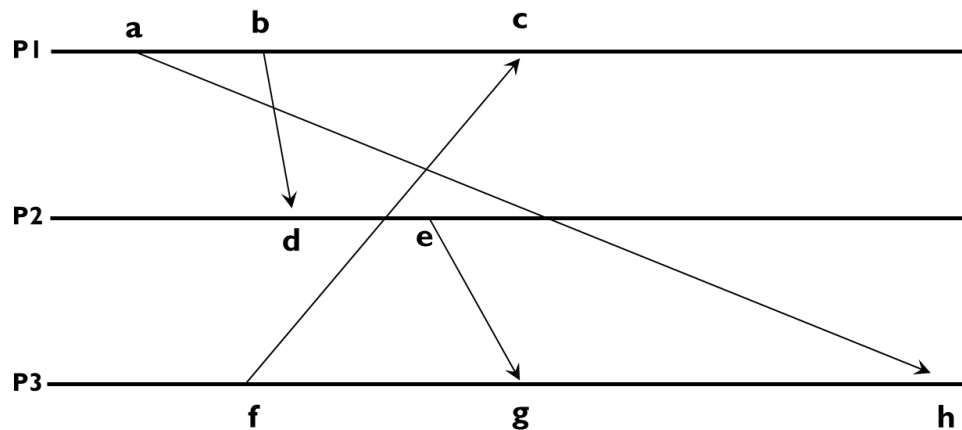
Sobre el mecanismo de comunicación RPC:

	A.- Al igual que ROI, siempre requiere que se utilice un servidor de nombres.
	B.- El stub cliente sirve para poder invocar procedimientos remotos como si fuesen locales
	C.- Sólo podemos definir como procedimientos remotos aquellos que no utilicen paso de parámetros por referencia.
	D.- Se basa en un modelo de invocación/respuesta con asincronía y persistencia
	E.- El programador debe escribir el código de los stubs cliente y servidor para cada procedimiento que pueda invocarse de forma remota

Sobre el mecanismo de comunicación ROI:

	A.- El cliente obtiene un proxy para invocar al objeto remoto.
	B.- El proxy incluye una referencia al objeto remoto
	C.- El mecanismo de ROI permite el paso de objetos como argumentos en las invocaciones, utilizando “paso por valor”. El paso de los objetos por referencia se simula mediante el paso por valor, de forma similar a como se emplea en el mecanismo RPC.
	D.- En Java RMI, se requiere emplear un lenguaje especial de definición de interfaces, denominado IDL (Interface Definition Language), para describir los interfaces de los objetos remotos
	E.- El programador debe escribir el código del proxy y esqueleto para cada objeto remoto

4. Dado el siguiente conjunto de eventos en un sistema distribuido, asumiendo que no hay otros eventos previos:



A.- Se cumple que $f \rightarrow h$
B.- Se cumple que $a \rightarrow g$
C.- Se cumple que $d \parallel c$
D.- Se cumple que $a \parallel e$
E.- Utilizando solamente el valor de los relojes de Lamport, no podemos determinar si los eventos "a" y "e" son concurrentes entre sí.

Respecto a la figura anterior:

A.- En "a" el reloj de Lamport $C(a) = 0$
B.- En "e" el reloj de Lamport $C(e) = 4$
C.- En "f" el reloj vectorial $V(f) = [1,1,1]$
D.- En "c" el reloj vectorial $V(c) = [3,0,1]$
E.- En "h" el reloj vectorial $V(h) = [2,2,3]$

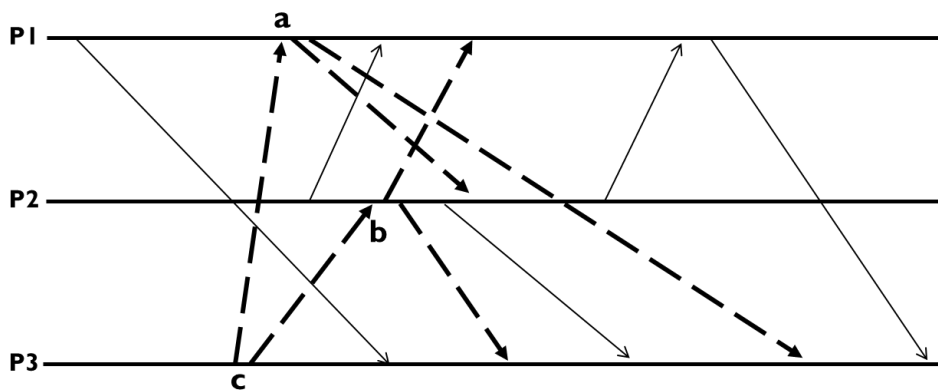
Respecto a los algoritmos de sincronización:

A.- El algoritmo de Chandy-Lamport requiere que los canales no pierdan ni desordenen mensajes
B.- El algoritmo de Bully falla si dos o más nodos inician el algoritmo de forma simultánea (es decir, si hay más de un iniciador)
C.- El algoritmo de elección de líder sobre anillos falla si dos o más nodos inician el algoritmo de forma simultánea (es decir, si hay más de un iniciador)
D.- El algoritmo centralizado de exclusión mutua visto en clase limita la escalabilidad y la tolerancia a fallos porque el coordinador supone un cuello de botella y un punto de fallo único
E.- El algoritmo distribuido de exclusión mutua visto en clase utiliza relojes lógicos para ordenar algunas solicitudes de entrada a la sección crítica.

7. Sobre el espacio de nombres y la resolución de nombres:

A.- Para el espacio de nombres suele utilizarse un esquema jerárquico (de tipo árbol).
B.- Algunos servicios de nombres permiten asociar atributos a las entidades y buscar por atributos.
C.- Por escalabilidad, suelen utilizarse varios servidores de nombres, que se ocupan de directorios distintos.
D.- La resolución de nombres iterativa supone que cada servidor debe resolver su parte y pasar el resto al siguiente servidor en la jerarquía.
E.- La resolución de nombres recursiva impone mayor responsabilidad al cliente, aligerando la carga sobre los servidores.

8. Respecto a la siguiente figura, suponga que P3 inicia el algoritmo de Chandy-Lamport en "c", siendo las líneas discontinuas los mensajes que se envían como consecuencia de la ejecución de este algoritmo y las líneas continuas los mensajes normales. Cuando finalice el algoritmo:



A.- Los estados locales de P1 en "a", P2 en "b" y P3 en "c" formarán parte del estado global
B.- Se habrá obtenido en este caso un corte inconsistente, pues alguno de los mensajes registrados como recibidos todavía no ha sido enviado.
C.- En el canal (P1,P3) se habrá registrado 1 mensaje
D.- En el canal (P2,P3) se habrá registrado 1 mensaje
E.- En el canal (P2,P1) se habrán registrado 2 mensajes

9. Sobre el servicio de localización:

	A.- Dado un identificador, retorna una dirección.
	B.- El modelo de difusión escala bien en distancia.
	C.- En el modelo de punteros adelante, el servicio de recolección de residuos ("garbage-collection") permite acortar la longitud de las cadenas de punteros, y por lo tanto agilizar accesos posteriores
	D.- En el modelo de difusión, asumiendo que el nodo A contiene la entidad a buscar, el fallo de un nodo distinto de A impide obtener la dirección de dicha entidad.
	E.- En el modelo de punteros adelante, asumiendo que el nodo A contiene la entidad a buscar, entonces el fallo de un nodo distinto de A que forma parte de la cadena de punteros impide obtener la dirección de dicha entidad.

Sobre las arquitecturas de un sistema distribuido:

	A.- Las arquitecturas de capas y las arquitecturas basadas en objetos son ejemplos de estilos arquitectónicos software.
	B.- La arquitectura de sistema define cuál es la organización lógica de los componentes, cómo están organizados y cómo deberían interactuar entre sí.
	C.- Las arquitecturas de sistema se dividen en arquitecturas centralizadas (que emplean distribución vertical), arquitecturas descentralizadas (que emplean distribución horizontal) y arquitecturas híbridas (que combinan distribución vertical y horizontal).
	D.- Los sistemas peer-to-peer son un ejemplo de arquitectura descentralizada con distribución horizontal.
	E.- La distribución horizontal se da cuando un cliente o servicio puede estar físicamente dividido en partes equivalentes a nivel lógico, pero cada parte funcionando en un nodo distinto.

EXAMEN 2do PARCIAL – Prácticas 3 y 4
Concurrencia y Sistemas Distribuidos

9 de Junio de 2014
Modelo A

Este bloque tiene una puntuación máxima de **10 puntos** (que aportará 1.5 puntos a la nota final de la asignatura). Indique, para cada una de las siguientes 20 afirmaciones, si éstas son verdaderas (**V**) o falsas (**F**). **Cada respuesta vale: correcta= 0.5, errónea= -0.5, vacía=0.**

Importante: Los primeros **2 errores** no penalizarán, de modo que tendrán una valoración equivalente a la de una respuesta vacía. A partir del 3º error (inclusive), sí se aplicará el decremento por respuesta errónea.

1. Sobre la práctica de Servicios de Dominio de Active Directory AD DS:

F	A.- El administrador de un dominio puede acceder a todas las máquinas de su dominio y subdominios, pero el administrador de un subdominio no puede acceder a ninguna máquina del dominio padre
F	B.- Los recursos compartidos pueden organizarse en unidades administrativas anidadas.
F	C.- Por defecto, cuando se crea una carpeta compartida, tanto el acceso local como el acceso remoto a la misma permiten las mismas operaciones.
F	D.- Sólo se pueden compartir carpetas dentro del mismo subdominio (ej. dentro de <i>amsterdam</i>)
F	E.- Los servicios AD DS, gestionados por el controlador de dominio, pueden ser instalados tanto en un sistema Windows Server, como en un sistema Windows 7, en Linux o en Mac OS.

2. Sobre la práctica de Servicios de Dominio de Active Directory AD DS:

V	A.- Se recomienda, pero no se requiere, que exista más de un controlador de dominio en cada dominio.
V	B.- La estructura de dominios de una organización puede estar compuesta por varios árboles.
F	C.- El usuario <i>eovic\Administrador</i> no puede eliminar un fichero creado por el usuario <i>amsterdam\Administrador</i> con los permisos asignados por defecto.
F	D.- Un usuario del subdominio <i>amsterdam</i> no puede acceder a un recurso compartido de una máquina que no pertenece a dicho subdominio.
V	E.- Un usuario sólo puede pertenecer a una unidad organizativa.

3. Sobre la práctica del Chat distribuido orientado a objetos basado en RMI:

	A.- El objeto canal mantiene una colección de usuarios y para difundir un mensaje invoca sobre ellos un método de forma remota.
	B.- Los canales distribuyen los mensajes de forma diferente a los usuarios conectados, dependiendo de si son usuarios de tipo ChatClient o de tipo ChatRobot.
	C.- Como se indica en el documento de la práctica, cuando un ChatRobot se conecta al servidor obtiene la lista de canales disponibles, a partir de la cual el usuario selecciona de forma interactiva el canal deseado.
	D.- Un ChatClient se debe lanzar tras el ChatServer. Sin embargo, como el ChatRobot no necesita interactuar con el usuario, el ChatRobot podría lanzarse antes del ChatServer.
	E.- Cuando se lanzan en distintas máquinas las aplicaciones ChatServer, ChatClient, o ChatRobot, deben tener asignados puertos (<i>ports</i>) diferentes.

4. Sobre la práctica del Chat distribuido orientado a objetos basado en RMI:

	A.- Si lanzamos más de una aplicación ChatClient o ChatRobot en la misma máquina, ambas aplicaciones obligatoriamente deben tener valores <i>myport</i> distintos.
	B.- Todo ChatClient debe recibir como parámetros la ubicación (host y puerto) del ChatServer.
	C.- ChatClient no abre ningún puerto, puesto que no es un servidor.
	D.- En la aplicación del Chat distribuido, ChatServer debe registrar primero su objeto principal "ChatServer" en el servidor de nombres, indicando también en dicho servidor el listado de canales que se ofrecen en el chat.
	E.- La aplicación "ChatServer", para poder registrar su objeto remoto, localiza primero el servidor de nombres (usando <code>LocateRegistry.getRegistry</code>) y luego registra el servicio (usando el método <i>bind</i> o el método <i>rebind</i> de la interfaz <code>Registry</code>).