Nombre	Legajo	Carrera

Cátedra de Informática III
Escuela de Ingeniería Electrónica
Departamento de Sistemas e Informática

Parcial I – 31 de mayo de 2011

1. A partir del concepto de "plataforma" desarrollado en clase, indique cuál de los siguientes elementos califica como tal, **justificando** esa calificación:
   a) Un PIC (Peripheral Interface Controller) Si puede programarse
   b) Un sistema basado en FPGA Sí puede programarse
   c) Un "pen drive" No puede programarse

2. ¿Que rol juega el directorio UDDI en la arquitectura de Web Services?

Permite localizar un servicio web determinado en Internet, obteniendo su URL y la descripción de su interface

3. ¿Cómo se identifica un socket en una red TCP/IP?

   Mediante dos parámetros: dirección de IP y número del puerto

4. Dados los siguientes atributos, indicar al lado de cada uno de ellos si son: (P) propios de los procesos o (T) si son propios de los threads

Atributos propios de los procesos	Atributos propios de los threads
Espacio de direcciones	Contador de programa
Variables globales	Registros
Ficheros abiertos	Pila
Procesos hijos	Estado
Alarmas pendientes	
Señales y controladores de señales	
Información de contabilidad	

5. Describa el problema de la Inversión de Prioridades

Puede darse al implementar la solución de Peterson. Consideremos un ordenador con dos procesos, H con alta prioridad y L con baja prioridad. Las reglas de planificación son tales que H pasa a ejecución inmediatamente siempre que se encuentre en estado listo. En un cierto momento, estando L en su región crítica, H pasa al estado listo (por ejemplo, debido a que se completa una operación de E/S que lo mantenía bloqueado). De inmediato H comienza la espera activa pero, ya que L nunca se planifica mientras H esté ejecutándose, L nunca tendrá la oportunidad de abandonar su región crítica, con lo cual H quedará para siempre dando vueltas al bucle de espera activa

6. Determinar si las siguientes sentencias son verdaderas (V) o falsas (F):
   a) Un thread puede estar en cualquiera de los estados de un proceso tradicional: en ejecución, bloqueado, listo o terminado.V

b) Un thread en ejecución tiene actualmente la CPU y está activo. V

c) Un thread bloqueado está esperando a que algún suceso lo desbloquee. V

d) Un thread puede bloquearse esperando a que tenga lugar algún suceso externo o a que algún otro thread lo desbloquee. V

e) Un thread listo está planificado para ejecutarse y lo hace tan pronto como le llega su turno.V

f) Las transiciones entre los estados de un thread son las mismas que las transiciones entre los estados de un proceso.V

g) Los threads no son más fáciles de crear y destruir que los procesos. (F)

h) Los threads proporcionan ganancia en el rendimiento cuando hay una necesidad substancial tanto de cálculo en la CPU como de E/S.V

7. ¿Cuál **es la característica fundamental** de un sistema de tiempo real que lo distingue de otros sistemas informáticos?:

   a) Eficiencia

   b) Previsibilidad (esta)

   c) Concurrencia

   d) Fiabilidad

8. i.-En el manejo de excepciones, un modelo donde el manejador pudiese decidir si reanudar la operación que provocó la excepción o terminarla, se denomina:

   a) Modelo de reanudación

   b) Modelo de notificación

   c) Modelo de terminación

   d) Modelo de escape

   e) Modelo híbrido (esta)

   Elija la opción correcta e indique cuál es el modelo que adopta Java al respecto(sólo el nombre): terminación o escape

   ii.-Con respecto al manejo de excepciones en Java, indique al lado de cada aseveración si son verdaderas (V) o (F):

   a) Java declara las excepciones como constantes (F)

   b) Java declara las excepciones como un objeto de tipo Throwable (V)

   c) En Java siempre hace falta declarar explícitamente las excepciones (F)

   d) En Java cuando se produce una excepción se crea una instancia de una clase de excepción adecuada, de acuerdo al error que se produjo y, se pasa como parámetro al manejador correspondiente (V)

   e) En Java algunas excepciones deben listarse en las signaturas de los métodos (forman parte de la interfaz del método) mediante una cláusula throws, mientras que en el caso de algunas otras no están sujetas a este requerimiento (V)

   f) En Java las excepciones verificadas o comprobadas, extienden de Exception (pero no de RunTimeException) y deben cumplir con el requerimiento catch or specify: es decir, manejarlas en algún bloque try del método o, especificarlas en la signatura del mismo mediante una claúsula throws (V)

   g) En Java la región o bloque protegido se indica mediante la palabra finally (F)

   h) En Java la región o bloque protegido se indica mediante la palabra try (V)

   i) En Java el dominio o ámbito de un manejador es un bloque (V)

j) En Java el dominio o ámbito de un manejador es una sentencia (F)

k) En Java cuando se produce una excepción en tiempo de ejecución en un bloque protegido, se busca si se encuentra el manejador apropiado (en los bloques catch) y sino se encuentra ninguno, se señala la excepción al invocante del método; si en el método invocante no se encuentra tampoco el manejador adecuado se sigue hacia arriba en la cadena de invocaciones. Si se llega al nivel superior en dicha cadena (main()) se aborta la ejecución. (V)

l) Java no permite la propagación de excepciones, es decir, buscar manejadores en la cadena de invocaciones (F)

m) En Java, en un bloque try/catch, colocar como último bloque catch uno que indique: catch(Exception e), se utiliza con el propósito (catch all) de atrapar cualquier excepción que se haya producido en el bloque try asociado pero que no haya sido atrapada por los bloques catch anteriores a éste (V)

n) En Java el bloque finally tiene garantizado su ejecución sin importar lo que ocurra en la ejecución del bloque try asociado (V)

o) En Java el bloque finally es obligatorio usarlo siempre que se use un bloque try (F)

p) En Java el bloque finally se utiliza para tareas de "limpieza" y es opcional al usar un bloque try. Es una herramienta clave para prevenir pérdidas de recursos (V)

q) En Java una RunTimeException indica condiciones excepcionales internas a la aplicación y que usualmente no pueden anticiparse o recuperarse de ellas (V)

r) Las excepciones de tipo RunTimeException deben listarse en la signatura de los métodos que las pueden generar, ya que son excepciones comprobadas o verificadas (F)

s) Un ejemplo de RunTimeException es NullPointerException cuando se pasa una referencia nula en lugar de un nombre de archivo pasado a FileReader, al intentar leer un archivo con FileReader. Otro ejemplo, es la excepción que se produce cuando se indica erróneamente el índice de un array (V)

t) Las excepciones de tipo RunTimeException usualmente indican errores de programación (bugs), con lo cual tiene más sentido corregirlos que atraparlos (V)

9. i.-En la técnica de tolerancia a fallos basada en redundancia dinámica de software, existen 4 fases constitutivas, ¿Cuál de las siguientes no es una de ellas?:

   a) Reparación de fallos y continuación del servicio

   b) Detección de errores

   c) Eliminación de fallos (esta)

   d) Valoración y confinamiento de los daños

   e) Recuperación de errores

   ii.-En relación a la recuperación de errores, indicar cuál de las siguientes afirmaciones es falsa:

   a) La recuperación de errores hacia delante intenta continuar desde el estado erróneo con correcciones selectivas en el estado del sistema

   b) En la corrección de errores hacia delante, se llama punto de recuperación (ckeckpoint) al punto en el que se restaura el proceso (F)

   c) La recuperación de errores hacia atrás se basa en retroceder el sistema a un estado anterior correcto que se guardó previamente y, ejecutar un segmento de programa alternativo (con otro algoritmo)

   d) La recuperación de errores hacia delante es específica de cada sistema e implica la correcta valoración de daños causados al entorno del sistema y la exacta determinación de la causa del fallo

   e) En la recuperación de errores hacia atrás no es necesario averiguar la causa ni la situación del fallo, por tanto sirve para errores de diseño

f) En la recuperación de errores hacia atrás no se puede deshacer los errores que aparecen en el sistema controlado

iii.-En la prevención de fallos existen dos fases, indique la opción correcta:

a) La tolerancia y degradación de fallos

b) La evitación y eliminación de fallos (esta)

c) La detección de errores y eliminación

d) La aceptación y corrección

10.-Indique los nombres de las partes del componente ideal tolerante a fallos, como así también el nombre de cada una de las señales marcadas en el siguiente dibujo:En las transparencias de la página están en castellano los nombres