

Examen de Programación Concurrente

junio 2005

Dpto. LSIS. Unidad de Programación.

Normas

En este examen se entregan tres partes diferenciadas: un planteamiento general, un cuestionario, y una petición de implementación. De ellas se entregan para su evaluación la segunda y la tercera. El cuestionario, que se entrega al principio del examen, debe rellenarse **en la misma hoja** en la que se entrega (consulta a un profesor si necesitas usar espacio adicional). Se recogerá **una hora y media** tras haberse entregado. Después se entregará el enunciado de la última parte.

La puntuación total del examen es de **10 puntos**.

Las soluciones se proporcionarán antes de la revisión y las calificaciones se darán a conocer el **20 de junio de 2005**. La revisión de este examen tendrá lugar el **21 de junio de 2005**.

CSI te necesita

Un departamento de investigación policial tiene problemas porque su sección de criminología obtiene resultados incorrectos al no limpiar los laboratorios entre pruebas. Se decide, por tanto, regular automáticamente las tareas de limpieza y uso de los laboratorios como sigue.

Existen K laboratorios, idénticos entre sí. Cada investigador (de los que hay un número indeterminado) dispone de un dispositivo en el que marca qué tipo de prueba quiere realizar y recibe un aviso cuando hay un laboratorio disponible para esa prueba, diciéndole cuál es éste. El final de la prueba se notifica a través del mismo dispositivo.

Hay un limpiador que se dedica únicamente a dormir y limpiar, y es avisado por otro dispositivo que le despierta cuando hay que limpiar un laboratorio, diciéndole cuál. El mismo dispositivo sirve para señalar el momento en que el laboratorio está limpio.

Hay tres tipos de pruebas, A , B y C . No es posible realizar más de una prueba en un laboratorio en cada momento, ni limpiarlo cuando se está realizando una prueba. Las reglas que rigen para utilizar **un mismo laboratorio** en dos pruebas sucesivas son las siguientes:

- Una prueba de tipo B o C puede realizarse inmediatamente después de una de tipo A .
- Una prueba de tipo C puede realizarse inmediatamente después de una de tipo B .
- Es necesario limpiar el laboratorio en cualquier otro caso.

Los dispositivos de los investigadores y el limpiador tienen el interfaz que se muestra a continuación, que reside en el *package* `Dispositivo`. Las variables de tipo `Tipo_Dispositivo` referencia las estructuras de datos, estado, etc. que son propios del dispositivo físico en que se ejecuta el código.

```
procedure Leer_Peticion(D: in out Tipo_Dispositivo;  
                       L: out Tipo_Prueba);
```

— Se queda bloqueado hasta que se pulsa en el dispositivo D
— el identificador correspondiente a una de las posibles pruebas.

```
procedure Avisar_Preparado(D: in out Tipo_Dispositivo;  
                           L: in Tipo_Laboratorio);
```

— Realiza en el dispositivo D un aviso mostrando el identificador del laboratorio L .

```
procedure Esperar_Fin(D: in out Tipo_Dispositivo);
```

— Se queda bloqueado hasta que se pulsa en el dispositivo D una tecla
— que señala el fin de uso o limpieza de un laboratorio.

El código del proceso que se ejecuta en el avisador del investigador es el siguiente:

```

D: Dispositivo.Tipo_Dispositivo = ... ;    — Estado de este dispositivo
loop — Avisador del investigador
    Dispositivo.Leer_Peticion(D, P);
    Controlador.Pedir_Laboratorio(P, L);
    Dispositivo.Avisar_Preparado(D, L);
    Dispositivo.Esperar_Fin(D);
    Controlador.Liberar_Laboratorio(P, L);
end loop;

```

Una especificación parcial del recurso se encuentra a continuación:

C-TADSOL Controlador

OPERACIONES

ACCIÓN PedirLaboratorio: $\text{Controlador}[es] \times \text{TipoPrueba}[e] \times \text{TipoLaboratorio}[s]$
ACCIÓN LiberarLaboratorio: $\text{Controlador}[es] \times \text{TipoPrueba}[e] \times \text{TipoLaboratorio}[e]$
ACCIÓN EsperarLimpieza: $\text{Controlador}[es] \times \text{TipoLaboratorio}[s]$
ACCIÓN LaboratorioLimpio: $\text{Controlador}[es] \times \text{TipoLaboratorio}[e]$

SEMÁNTICA

DOMINIO:

TIPO: $\text{Controlador} = (\text{limpios: TipoCnjLaboratorio} \times \text{usadoA: TipoCnjLaboratorio} \times \text{usadoB: TipoCnjLaboratorio} \times \text{usadoC: TipoCnjLaboratorio})$
 $\text{TipoCnjLaboratorio} = \text{Conjunto}(\text{TipoLaboratorio})$
 $\text{TipoLaboratorio} = 1..K$
 $\text{TipoPrueba} = \text{PrA} \mid \text{PrB} \mid \text{PrC}$

INICIAL(c): $c = (\{1..K\}, \emptyset, \emptyset, \emptyset)$

INVARIANTE: $\forall c \in \text{Controlador} \bullet c = (l, ua, ub, uc) \wedge l \cap ua = \emptyset \wedge l \cap ub = \emptyset \wedge l \cap uc = \emptyset \wedge ua \cap ub = \emptyset \wedge ua \cap uc = \emptyset \wedge ub \cap uc = \emptyset \wedge \text{Card}(l \cup ua \cup ub \cup uc) \leq K$

CPRE: Se puede usar algún laboratorio para realizar esta prueba

CPRE: $\text{cont.limpios} \neq \emptyset \vee (\text{prueba} = \text{PrB} \wedge \text{cont.usadaA} \neq \emptyset) \vee (\text{prueba} = \text{PrC} \wedge (\text{cont.usadaA} \neq \emptyset \vee \text{cont.usadaB} \neq \emptyset))$

PedirLaboratorio(cont, prueba, lab)

POST: $\text{cont}^{ent} = (l, ua, ub, uc) \wedge (\exists p \bullet \text{lab}^{sal} = p \wedge ((l \neq \emptyset \wedge p \in l \wedge \text{cont}^{sal} = (l - \{p\}, ua, ub, uc)) \vee (\text{prueba}^{ent} \neq \text{PrA} \wedge ua \neq \emptyset \wedge p \in ua \wedge \text{cont}^{sal} = (l, ua - \{p\}, ub, uc)) \vee (\text{prueba}^{ent} = \text{PrC} \wedge ub \neq \emptyset \wedge p \in ub \wedge \text{cont}^{sal} = (l, ua, ub - \{p\}, uc))))$

CPRE: Cierto

PRE: $\text{cont} = (l, ua, ub, uc) \wedge \text{lab} \notin l \cup ua \cup ub \cup uc$

LiberarLaboratorio(cont, prueba, lab)

POST: $\text{cont}^{ent} = (l, ua, ub, uc) \wedge ((\text{prueba}^{ent} = \text{PrA} \wedge \text{cont}^{sal} = (l, ua \cup \{\text{lab}\}, ub, uc) \vee (\text{prueba}^{ent} = \text{PrB} \wedge \text{cont}^{sal} = (l, ua, ub \cup \{\text{lab}\}, uc) \vee (\text{prueba}^{ent} = \text{PrC} \wedge \text{cont}^{sal} = (l, ua, ub, uc \cup \{\text{lab}\})))$

CPRE: Algún laboratorio ha sido usado

CPRE: $\text{cont.usadoA} \cup \text{cont.usadoB} \cup \text{cont.usadoC} \neq \emptyset$

EsperarLimpieza(cont, lab)

POST: ...

CPRE: Cierto

PRE: $\text{cont} = (l, ua, ub, uc) \wedge \text{lab} \notin l \cup ua \cup ub \cup uc$

LaboratorioLimpio(cont, lab)

POST: ...

Examen de Programación Concurrente

junio 2005

Dpto. LSIS. Unidad de Programación.

Normas

Disponéis de **una hora y media** para responder a los dos apartados de esta parte.
Cada uno de ellos se deberá entregar en un conjunto de hojas separadas.

1. Implementación [6 puntos]

Para cada uno de los apartados siguientes ha de entregarse lo que se pide en ellos. Recordad **entregar la respuesta a cada apartado en un juego de hojas separadas**.

1.1. Objetos protegidos [3 puntos]

Realizar una implementación en Ada 95 de las operaciones **PedirLaboratorio** y **LiberarLaboratorio** del **C-TADSOL** utilizando objetos protegidos y que incluya la definición de tipos y variables de estado necesarias.

1.2. *Rendez-Vous* / Paso de mensajes [3 puntos]

Realizar una implementación en Ada 95 de las operaciones **PedirLaboratorio** y **LiberarLaboratorio** del **C-TADSOL** utilizando *Rendez-Vous* (con paso de mensajes explícito si se considera necesario, modificando en este caso el código del cliente para hacer explícito el envío / recepción a través de los canales) y que incluya la definición de tipos y variables de estado necesarias y el código de todas las operaciones solicitadas.