



Universidad
de Huelva

DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA,
SISTEMAS INFORMÁTICOS Y AUTOMÁTICA

PROGRAMACIÓN

17 de septiembre de 2004

Alumno: _____

Puntuación: 2 ptos.

Correcta: +1 Errónea: -0,5

Mínimo: 14

Una técnica para solucionar el problema de la Exclusión Mutua tiene como objetivo conseguir que:

- a) los procesos se ejecuten concurrentemente para aumentar el uso de la CPU
- b) los procesos que deseen acceder a una Sección Crítica no lo hagan a la vez
- c) los procesos se comuniquen mediante memoria compartida o paso de mensajes
- d) ninguna de las anteriores

El uso de variables compartidas como mecanismo de solución de los problemas de la concurrencia:

- a) no puede utilizarse, ya que no es posible la Exclusión Mutua
- b) es la mejor solución para la sincronización y comunicación entre procesos
- c) su inconveniente es que los procesos consumen tiempo de CPU innecesario en esperas activas

Un programa concurrente se dice de Orden Parcial:

- a) porque el programador conoce de forma parcial los procesos a implementar
- b) porque el programa debe implementarse de forma parcial, utilizando procesos
- c) porque el programa establece como orden estricto de ejecución la secuencia de escritura
- d) porque no se puede establecer un orden de ejecución entre las sentencias de proceso diferentes

El término Exclusión Mutua en programación concurrente implica:

- a) El acceso restringido a un conjunto de instrucciones por un único proceso
- b) La capacidad de comunicación y sincronización entre procesos
- c) La posibilidad de sincronizar procesos evitando deadlock
- d) El uso de semáforos como primitiva para resolver problemas de concurrencia

La Sección Crítica dentro de un programa Concurrente es

- a) un conjunto de instrucciones que debe ejecutarse en Exclusión Mutua
- b) un conjunto de instrucciones para la negociación y liberación de la Exclusión Mutua
- c) un mecanismo para comunicación y sincronización de procesos

La sincronización bidireccional mediante semáforos se puede obtener con:

- a) P1 :: wait(s1); signal(s2); P2 :: wait(s2); signal(s1); initial(s1,1); initial(s2,0);
- b) P1 :: signal(s1); wait(s2); P2 :: wait(s1); signal(s2); initial(s1,0); initial(s2,0);
- c) P1 :: wait(s1); signal(s1); P2 :: wait(s2); signal(s2); initial(s1,0); initial(s2,0);
- d) P1 :: signal(s1); wait(s1); P2 :: signal(s2); wait(s2); initial(s1,0); initial(s2,0);

Que puede ocurrir en el siguiente pseudocódigo

PROCESS p1;	PROCESS p2;	BEGIN	
...	...	acceso := FALSE;	a) Alternancia
acceso := TRUE;	acceso := TRUE;	COBEGIN	b) Falta de Exclusión
WHILE acceso Do null;	WHILE acceso Do null;	P1;	<input checked="" type="radio"/> c) Interbloqueo
<Sección-Crítica>	<Sección-Crítica>	P2;	d) Se consigue Exclusión Mutua
acceso:=FALSE;	acceso:=FALSE;	COEND	
...	...	END.	
END;	END;		

Si 'S' es un semáforo inicializado a 1, la Exclusión Mutua a SENT podría conseguirse con:

- a) signal(S); SENT; wait(S);
- b) signal(S); wait(S); SENT;
- c) wait(S); SENT; signal(S);
- d) signal; SENT; wait;

Los semáforos S1 y S2 en el pseudocódigo siguiente llevan a cabo la misión de:

PROCESS P1;	PROCESS P2;	BEGIN	
...	...	initial(S1,0);	a) ejecución de Sent3 y Sent4 en Exclusión Mutua
Sent1;	wait(S2);	initial(S2,0);	b) ejecución de Sent2 y Sent3 en Exclusión Mutua
wait(S1);	Sent2;	COBEGIN	<input checked="" type="radio"/> c) interbloqueo
signal(S2);	signal(S1);	P1; P2;	d) establecer la secuencia sent1 → sent2 → sent3
Sent3;	Sent4;	COEND	
...	...	END.	
END;	END;		

El tipo CONDITION en un monitor

- a) es el mecanismo de sincronización para los monitores
- b) permite la Exclusión Mutua dentro de los procedimientos del monitor
- c) facilita la comunicación entre procesos
- d) utiliza las operaciones wait y signal para la sincronización de procesos



Universidad de Huelva

DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA, SISTEMAS INFORMÁTICOS Y AUTOMÁTICA

PROGRAMACIÓN

17 de septiembre de 2004

Alumno: _____

La operación resume(c1) en un monitor tiene como objetivo:

- a) detener al proceso la ejecuta
- b) si no existe un proceso bloqueado en c1, incrementar el valor de c1
- c) permitir la comunicación de procesos mediante la condición c1.
- d) ninguna de las anteriores

La facilidad REQUEUE en un recurso:

- a) detiene al proceso que la ejecuta en su cola asociada
- b) es la forma en que los procesos invocan a los procedimientos del recurso
- c) permite el paso de un procedimiento con guarda a otro dentro del recurso
- d) es la forma de comunicar los proceso dentro del recurso

Las RCC permiten la sincronización y la exclusión mutua entre procesos mediante la sintaxis:

- a) Cobegin SENT; Coend;
- b) Accept pg(I: Integer) WHEN True Do SENT;
- c) Guarded procedure pg(I: Integer) WHEN True; BEGIN SENT; END;
- d) Region pg When True DO SENT;

Los recursos

- a) A diferencia que los monitores permiten Exclusión Mutua en su código
- b) Mejoran el mecanismo de sincronización de los monitores bajo procedimientos con guardas
- c) Facilita la sincronización entre procesos con las operaciones delay y resume
- d) Son, como los monitores, una primitiva de paso de mensajes entre procesos

La programación concurrente en Java puede obtenerse

- a) Implementando la clase Thread
- b) Heredando de la interface Runnable o implementando la clase Thread
- c) Heredando de la clase Thread, y a la vez, implementando la clase Runnable
- d) Implementando la interfaz Runnable o heredando de la clase Thread

La exclusión mutua en Java puede obtenerse

- a) Con las operaciones delay y resume dentro de clases que heredan de la clase Thread
- b) Utilizando el modificador *synchronized*
- c) Con Recursos o Unidades Protegidas con métodos declarados con guardas
- d) Utilizando los métodos *wait()* y *notify()*

La sincronización en Java puede obtenerse

- a) Con las operaciones delay y resume dentro de clases que heredan de la clase Thread
- b) Utilizando el modificador *synchronized*
- c) Con Recursos o Unidades Protegidas con métodos declarados con guardas
- d) Utilizando los métodos *wait()* y *notify()*

El mecanismo de paso de mensajes estilo ADA

- a) Permite comunicación exclusivamente en el sentido emisor - receptor
- b) Es un mecanismo de paso de mensajes asíncrono
- c) Es un mecanismo de concurrencia que requiere una memoria compartida
- d) Ninguna de las anteriores

Los canales son

- a) Un mecanismo, de memoria compartida, para obtener exclusión mutua con las operaciones: ! y ?
- b) Un mecanismo de paso de mensajes asíncrono, flujo en una dirección y denominación indirecta
- c) Un mecanismo de paso de mensajes síncrono
- d) Un mecanismo para sincronizar procesos en memoria compartida

La Espera Selectiva puede estar afectada por guardas con la sintaxis

- | | | | |
|---|------------------------------|------------------------------------|--------------------|
| <input checked="" type="radio"/> a) SELECT
WHEN cond =>
Cita1 | b) SELECT
Cita1 WHEN cond | c) WHEN cond =>
SELECT
Cita1 | d) SELECT
Cita1 |
| OR
WHEN cond =>
Cita2 | OR
Cita2 WHEN cond | OR
Cita2 | OR
Cita2 |
| END; | END; | END; | WHEN cond
END; |

U nuevo delay
V[N] = dato;

→ 0) ✓