

CC4302 Sistemas Operativos
Control 1 – Semestre Otoño 2013
 Prof.: Luis Mateu

Pregunta 1

Un thread ejecuta la función T1 y otro thread ejecuta paralelamente T2.

<pre>int T1() { int v1= prologo1(); int v2= reunion1(v1) epilogo1(v1, v2); }</pre>	<pre>int T2() { int v2= prologo2(); int v1= reunion2(v2) epilogo2(v1, v2); }</pre>
--	--

En un punto de la ejecución (y una sola vez) deben intercambiar los valores que calcularon en sus prólogos. Este intercambio se realiza invocando las funciones reunion1 y reunion2. Se tiene la siguiente implementación de estas 2 funciones:

<pre>volatile int flag1= FALSE, flag2= FALSE; int val1, val2;</pre>	
<pre>int reunion1(int v) { flag1= TRUE; val1= v; while (!flag2) ; return val2; }</pre>	<pre>int reunion2(int v) { flag2= TRUE; val2= v; while (!flag1) ; return val1; }</pre>

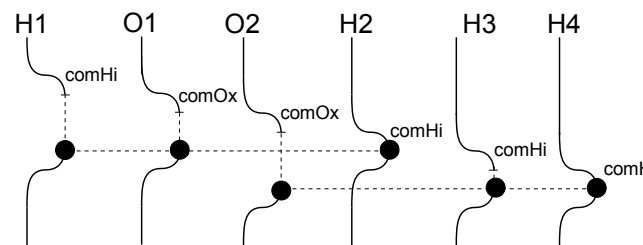
- (a) Discuta si esta solución es correcta (sin importar la eficiencia). Si es incorrecta haga un diagrama de threads que muestre una situación en donde el resultado es incorrecto. Además discuta si es posible hacer un pequeño cambio para que la solución sí sea correcta.
- (b) Escriba una solución correcta y *eficiente* (es decir sin *busy-waiting*) de reunion1 y reunion2 usando 2 semáforos de nSystem.

Pregunta 2

En el programa de arriba a la derecha, varios threads productores de oxígeno ejecutan el procedimiento oxigen y varios threads productores de hidrógeno ejecutan el procedimiento hidrogen. Un productor de oxígeno aporta su átomo invocando combineOxi y un productor de hidrógeno aporta su átomo invocando combineHidro. Estos procedimientos deben esperar hasta formar una molécula de agua (H₂O).

<pre>void oxigen() { for(;;) { Oxigen *o= produceOxi(); H2O *h2o= combineOxi(o); use1(h2o); } }</pre>	<pre>void hidrogen() { for(;;) { Hidrogen *h= produceHidro(); H2O *h2o= combineHidro(h); use2(h2o); } }</pre>
---	---

El siguiente diagrama de threads muestra cómo se deben asociar los productores de oxígeno con los productores de hidrógeno.



Programa los procedimientos combineOxi y combineHidro. Ud. debe formar la molécula de agua apenas disponga de 2 átomos de hidrógeno h1 y h2 y uno de oxígeno o. Para ello invoque el procedimiento:

```
H2O makeH2O(Hidrogen *h1, Hidrogen *h2, Oxigen *o);
```

Tanto la invocación de combineOxi como las 2 invocaciones de combineHidro deben retornar la molécula de agua construida a partir de los átomos que se suministraron en esas 3 invocaciones.

Para la sincronización Ud. debe usar los monitores de nSystem.

Observación: Está permitido que los productores de su solución sufran de hambruna. Es decir, no es necesario que atienda a los productores de oxígeno o los productores de hidrógeno en orden FIFO. Por ejemplo, en el diagrama de threads sería válido producir la primera molécula de agua a partir del átomo suministrado por el thread O2 (y no O1). A modo de referencia la versión secuencial de esta aplicación sería:

```
for(;;) {
    Oxigen *o= produceOxi();
    Hidrogen *h1= produceHidro();
    Hidrogen *h2= produceHidro();
    H2O h2o= makeH2O(h1, h2, o);
    use1(h2o);
    use2(h2o);
}
```