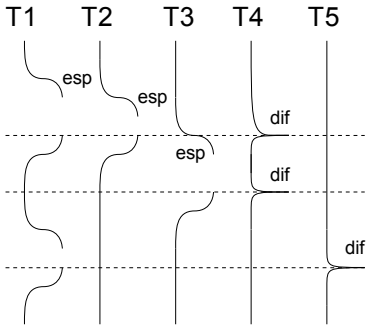
CC41B Sistemas Operativos
Control 1 – Semestre Primavera 2008
Prof.: Luis Mateu

Pregunta 1

Para los procedimientos `esperar` y `difundir` se especifica que cuando cualquier thread invoca `esperar`, ese thread debe quedar bloqueado hasta que algún thread invoque `difundir`, retornando la información suministrada por `difundir`. Si la invocación de `esperar` ocurre simultáneamente con la invocación de `difundir` (es decir, si hay algún traslape en las dos invocaciones), el thread puede continuar de inmediato o bien puede bloquearse hasta la siguiente invocación de `difundir`. Si ocurren 2 invocaciones simultáneas de `difundir`, `esperar` puede retornar la información suministrada en cualquiera de esas 2 invocaciones de `difundir`.



Se propone la siguiente implementación:

```
int cont= 0;
Info *info;

void difundir(Info *infoP) {
  cont++;
  info= infoP;
}
```

```
Info *esperar() {
  int micont= cont;
  while (micont==cont)
    ;
  return info;
}
```

i. Haga un diagrama de threads que muestre que la solución propuesta es inconsistente. Por ejemplo un thread podría retornar información equivocada. (Considere que nunca se produce el desborde de la variable `cont`.)

ii. Indique si es posible corregir esta solución haciendo una pequeña modificación de forma tal que ya no se produzcan inconsistencias.

iii. Critique esta solución desde el punto de vista de la eficiencia.

iv. Escriba una solución consistente y *eficiente* de `esperar` y `difundir` usando como herramienta de sincronización los semáforos de nSystem. Considere que estos semáforos garantizan que los nWaitSem serán atendidos en orden FIFO.

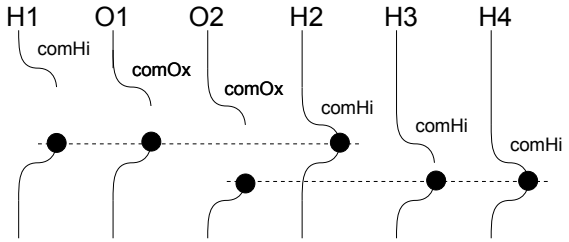
Pregunta 2

En el programa de más abajo, 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 (H_2O).

```
void oxigen() {
 for(;;) {
  Oxigen *o= produceOxi();
  H2O *h2o= combineOxi(o);
  use1(h2o);
 }
}
```

```
void hidrogen() {
 for(;;) {
  Hidrogen *h= produceHidro();
  H2O *h2o= combineHidro(h);
  use2(h2o);
 }
}
```

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



Programe 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 sumnistraron 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).