

### Auxiliar 3

Prof.:Luis Mateu Aux.: Juan Manuel Barrios  
31 de agosto de 2007

#### 1. Diagrama de threads (Control 1 1994)

Se tiene la siguiente solución incorrecta para el problema de los lectores/escritores:

<pre>int readers= 0, writing= FALSE;  void enterRead() {     if (readers==0) {         while (writing)             ;         writing= TRUE;     }     readers++; }  void exitRead() {     readers--;     if (readers==0)         writing= FALSE; }</pre>	<pre>void enterWrite() {     while (writing)         ;     writing= TRUE; }  void exitWrite() {     writing= FALSE; }</pre>
--	---

Muestre mediante un diagrama de threads que un lector puede trabajar simultáneamente con un escritor. Detalle bien las invocaciones de procedimientos y los instantes en que se hacen los `if`, se incrementan variables, etc.

Propuesto: Corrija esta solución utilizando 2 semáforos (hint: utilice la estructura de la solución incorrecta).

#### 2. Equivalencia de mecanismos de sincronización

Implemente una interfaz para semáforos FIFO utilizando monitores. Para su implementación defina métodos análogos a `nWaitSem`, `nSignalSem` y `nMakeSem` y luego utilice un mecanismo de asignación de números por orden de llegada.

#### 3. Impresora compartida (Examen 1998)

Suponga que en *nSystem* se dispone de una impresora compartida por todas las tareas. Dos o más tareas pueden usar simultáneamente la impresora, pero en este caso las líneas de la impresora saldrían entremezcladas. En una aplicación se desea evitar el entremezclado de líneas haciendo que cada tarea solicite el acceso exclusivo a la impresora antes de ocuparla y además notifique cuando termina de utilizarla. Por lo tanto una tarea tendrá la siguiente forma:

```
tarea() {
    ...
    obtenerImpresora();
    ... /* utilizar impresora */
    devolverImpresora();
    ...
}
```

Además, por razones de ahorro de electricidad, se necesita que la impresora se coloque en modo de bajo consumo cada vez que transcurran 5 minutos sin ser utilizada por ninguna tarea.

Para colocar la impresora en modo de bajo consumo invoque el procedimiento `modoBajoConsumo()`. Para volver a usar la impresora cuando está en modo de bajo consumo, invoque el procedimiento `modoUsoNormal()`.

Implemente los procedimientos `obtenerImpresora()`, `devolverImpresora()` y un procedimiento `inicializarImpresora()` que se invoca al inicio de `nMain`.

El uso de *busy-waiting* o consultas periódicas para ver si se han cumplido los 5 minutos es equivalente a no responder la pregunta.