



TAREA 3 - Sistemas Operativos

Semaforos

Profesor: Viktor Tapia

Ayudantes: Fabian Saso - Juan Carlos Bustamante

16-11-2016

1. Ejercicio

1. Problema de Escritura y Lectura confluyente.

Se tiene un fichero de datos (archivo de texto) que es utilizado por varios **procesos**, de los cuales pueden ser tanto de lectura como de escritura.

El fichero es considerado como un recurso del sistema y por esto solo puede ser utilizado por **un** proceso a la vez, es decir, o bien un proceso está leyendo su información, o bien estará escribiendo en él. Ambas acciones **no** pueden ocurrir de manera simultanea y debe considerarse que un proceso escritor tiene preferencia **por sobre** un lector.

La mejor manera para poder satisfacer la necesidad es utilizando un **semáforo** el cual actuará de manera binaria, siendo 0 el equivalente a que el recurso esté siendo utilizado y 1 si aquel está disponible. Para esta tarea deberá considerar a un usuario lector y un escritor como dos procesos y al fichero como un recurso.

Implemente un algoritmo que resuelva la confluencia y aplique los semáforos necesarios.

Hint. Para que el problema esté bien resuelto se tiene que cumplir:

- No se puede hacer esperar a ningún proceso lector si ningún proceso escritor accede al recurso.
- Cuando un escritor debe realizar su tarea, debe ocurrir cuanto antes, sin que deban interferir los procesos lectores. (Para ello se establecen prioridades).

2. Los alumnos de ILI, Además deberán deben resolver el siguiente problema:

Cinco Filósofos se sientan alrededor de una mesa y pasan su vida cenando y pensando. Cada filósofo tiene un plato de fideos y un tenedor a la izquierda de su plato. Para comer los fideos son necesarios dos tenedores y cada filósofo sólo puede tomar los que están a su izquierda y derecha. Si cualquier filósofo coge un tenedor y el otro está ocupado, se quedará esperando, con el tenedor en la mano, hasta que pueda coger el otro tenedor, para luego empezar a comer.

Si dos filósofos adyacentes intentan tomar el mismo tenedor a una vez, se produce una condición de carrera: ambos compiten por tomar el mismo tenedor, y uno de ellos se queda sin comer. Si todos los filósofos cogen el tenedor que está a su derecha al mismo tiempo, entonces todos se quedarán esperando eternamente, porque alguien debe liberar el tenedor que les falta. Nadie lo hará porque todos se encuentran en la misma situación (esperando que alguno deje sus tenedores). Entonces los filósofos se morirán de hambre. Este bloqueo mutuo se denomina interbloqueo o deadlock.

El problema consiste en encontrar un algoritmos que permita que los filósofos nunca se mueran de hambre.



2. Consideraciones

- Como base de ayuda, los problemas descritos, son problemas clásicos de semáforos por lo que su implementación no debe ser problema.
- Además de realizar la implementación del algoritmo, deben realizar un informe en Latex que explique su solución y los pasos que el algoritmo realiza (máximo 2 planas).
- Debe ser Programada en lenguaje C o C++.

3. Entrega

- La tarea debe realizarse de manera individual.
- Para la los alumnos de **INF** la fecha de entrega es hasta el **Jueves 1 de Diciembre de 2016 a las 23:55 hrs**, mientras que los alumnos de **ILI** la fecha de entrega es hasta el **Domingo 4 de Diciembre de 2016 a las 23:55 hrs**.
- Por cada día de atraso se descuentan 20 puntos (máximo 3 días de atraso).
- La tarea debe subirse a Moodle con el formato **Tarea3_Apellido_Rol.tar.gz**. Dentro del tar debe ir una carpeta **con el mismo nombre** que contiene el código fuente del programa en c con su makefile, ademas del archivo .tex y .pdf. Además del README.txt con las instrucciones de compilación de ser necesario.
- Cualquier intento de copia, o que el algoritmo no tengan explicación dentro del informe, **será sancionado con nota 0** para todos los involucrados.