

EJERCICIOS TEMA 4 PROGRAMACION CONCURRENTE

FECHA REALIZACIÓN:

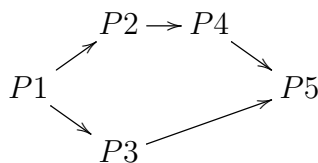
NOMBRE:

NOTA:

1. Un grafo de precedencia es un grafo acíclico dirigido. Los nodos representan tareas y los arcos indican el orden en el que las tareas se deben ejecutar. En particular, una tarea se puede ejecutar tan pronto como sus predecesoras terminen. Asume que las tareas son procesos y que cada proceso tiene el siguiente perfil:

P:: esperar a los predecesores, si tiene
cuerpo de P
señalizar a los sucesores, si tiene

- a) Usando semáforos, muestra como sincronizar 5 procesos cuyo orden de ejecución permitido viene determinado por el siguiente grafo de precedencia:



Minimiza el número de semáforos que utilizas, y no impongas restricciones que no estén especificadas en el grafo.

- b) Describe como sincronizar procesos, dado un grafo arbitrario de precedencia. En particular, diseña un método general para asignar semáforos a aristas o procesos y para usarlos. No intentes que el número de semáforos sea mínimo.

2. Considera el siguiente programa:

```
int x=0, y=0, z=0;
Sem lock1=1, lock2=1;
process foo{
    z=z+2;
    P(lock1);
    x=x+2;
    P(lock2);
    V(lock1);
    y=y+1;
    V(lock2);}
process bar{
    P(lock2);
    x=x+1;
    P(lock1);
    V(lock1);
    y=y+2;
    V(lock2);
    z=z+1;}
```

- Asume que el planificador es débilmente justo. El programa se bloquea siempre, a veces, nunca? Explícalo.
- Cúales son los posibles valores finales para x, y, z? Incluye los estados de bloqueo si los hay.

3. Desarrolla una implementación utilizando semáforos de las siguientes dos primitivas que proporciona el kernel de UNIX:
 - sleep: bloquear al proceso ejecutando
 - wakeup: despertar a *todos* los procesos dormidos
4. Hay dos tipos de procesos M and W que intentan entrar a un baño. Sólo se permite la entrada a procesos de tipo M cuando no hay ningún proceso de tipo W dentro, y viceversa. Se pide:
 - solución no justa
 - solución justa que alterna el uso del baño entre M y W