

## Práctica 2 – Semáforos

### CONSIDERACIONES PARA RESOLVER LOS EJERCICIOS:

- Los semáforos deben estar declarados en todos los ejercicios.
- Los semáforos deben estar inicializados en todos los ejercicios.
- No se puede utilizar ninguna sentencia para *setear* o *ver* el valor de un semáforo.
- Debe evitarse hacer *busy waiting* en todos los ejercicios.
- En todos los ejercicios el tiempo debe representarse con la función *delay*.

1. Existen N personas que deben ser chequeadas por un detector de metales antes de poder ingresar al avión.
  - a. Implemente una solución que modele el acceso de las personas a un detector (es decir si el detector está libre la persona lo puede utilizar caso contrario debe esperar).
  - b. Modifique su solución para el caso que haya tres detectores.
2. Un sistema operativo mantiene 5 instancias de un recurso almacenadas en una cola, cuando un proceso necesita usar una instancia del recurso la saca de la cola, la usa y cuando termina de usarla la vuelve a depositar.
3. Suponga que existe una BD que puede ser accedida por 6 usuarios como máximo al mismo tiempo. Además los usuarios se clasifican como usuarios de prioridad alta y usuarios de prioridad baja. Por último la BD tiene la siguiente restricción:
  - no puede haber más de 4 usuarios con prioridad alta al mismo tiempo usando la BD.
  - no puede haber más de 5 usuarios con prioridad baja al mismo tiempo usando la BD.
 Indique si la solución presentada es la más adecuada. Justifique la respuesta.

#### Var

```
sem: semaphore := 6;
alta: semaphore := 4;
baja: semaphore := 5;
```

#### Process Usuario-Alta [I:1..L]::

```
{ P(sem);
  P(alta);
  //usa la BD
  V(sem);
  V(alta);
}
```

#### Process Usuario-Baja [I:1..K]::

```
{ P(sem);
  P(baja);
  //usa la BD
  V(sem);
  V(baja);
}
```

4. Se tiene un curso con 40 alumnos, la maestra entrega una tarea distinta a cada alumno, luego cada alumno realiza su tarea y se la entrega a la maestra para que la corrija, esta revisa la tarea y si está bien le avisa al alumno que puede irse, si la tarea está mal le indica los errores, el alumno corregirá esos errores y volverá a entregarle la tarea a la maestra para que realice la corrección nuevamente, esto se repite hasta que la tarea no tenga errores.
5. Suponga que se tiene un curso con 50 alumnos. Cada alumno elige una de las 10 tareas para realizar entre todos. Una vez que todos los alumnos eligieron su tarea comienzan a realizarla. Cada vez que un alumno termina su tarea le avisa al profesor y si todos los alumnos que tenían la misma tarea terminaron el profesor les otorga un puntaje que representa el orden en que se terminó esa tarea.  
*Nota:* Para elegir la tarea suponga que existe una función ***elegir*** que le asigna una tarea a un alumno (esta función asignará 10 tareas diferentes entre 50 alumnos, es decir, que 5 alumnos tendrán la tarea 1, otros 5 la tarea 2 y así sucesivamente para las 10 tareas).
6. A una empresa llegan E empleados y por día hay T tareas para hacer ( $T > E$ ), una vez que todos los empleados llegaron empezaron a trabajar. Mientras haya tareas para hacer los empleados tomaran una y la realizarán. Cada empleado puede tardar distinto tiempo en realizar cada tarea. Al finalizar el día se le da un premio al empleado que más tareas realizó.
7. Existe una casa de comida rápida que es atendida por 1 empleado. Cuando una persona llega se pone en la cola y espera a lo sumo 10 minutos a que el empleado lo atienda. Pasado ese tiempo se retira sin realizar la compra.
8. Hay una fábrica con M operarios en donde se deben realizar N tareas (siendo  $M = N \times 5$ ). Cada tarea se realiza de a grupos de 5 operarios, ni bien llegan a la fábrica se juntan de a 5 en el orden en que llegaron y cuando se ha formado el grupo se le da la tarea correspondiente empezando de la tarea uno hasta la enésima. Una vez que los operarios del grupo tienen la tarea asignada producen elementos hasta que hayan realizado exactamente X entre los operarios del grupo. Una vez que terminaron de producir los X elementos, se juntan los 5 operarios del grupo y se retiran.  
*Nota:* cada operario puede hacer 0, 1 o más elementos de una tarea. El tiempo que cada operario tarda en hacer cada elemento es diferente y random. Maximice la concurrencia.
9. En un curso hay dos profesores que toman examen en forma oral, el profesor A llama a los alumnos de acuerdo al orden de llegada, mientras que el profesor B llama a cualquier alumno (que haya llegado). Existen N alumnos que llegan y se quedan esperando hasta ser llamados para rendir, luego de que uno de los dos profesores lo atiende, se va. Indicar si la siguiente solución realizada con semáforo resuelve lo pedido. Justificar la respuesta.

<pre> string estado[N] = ([N], "Esperando" ) queue colaA, colaB sem lleoA, lleoB = 0 sem esperando[N] = ([N], 0) sem mutex[N] = ([N], 1) sem mutexA, mutexB = 1 </pre>	
<p><b>Profesor A::</b></p> <pre> { int idAlumno   while (true)     { P(lleoA)       P(mutexA)       idAlumno = pop(colaA)       V(mutexA)       P(mutex[idAlumno])       If (estado[idAlumno] == "Esperando")         estado[idAlumno] = "A"         V(mutex[idAlumno])         V(esperando[idAlumno])         //Se toma el examen//         V(esperando[idAlumno])       else         V(mutex[idAlumno])     } } </pre>	<p><b>Profesor B::</b></p> <pre> { int idAlumno   while (true)     { P(lleoB)       P(mutexB)       idAlumno = popAleatorio(colaB)       V(mutex(B))       P(mutex[idAlumno])       If (estado[idAlumno] == "Esperando")         estado[idAlumno] = "B"         V(mutex[idAlumno])         V(esperando[idAlumno])         //Se toma el examen//         V(esperando[idAlumno])       else         V(mutex[idAlumno])     } } </pre>
<p><b>Alumno[i: 1..N]</b></p> <pre> { P(mutexA)   push(colaA, i)   V(mutexA)   P(mutexB)   push(colaB, i)   V(mutexB)   P(esperando[i])   if (estado[i] == "A")     //Interactúa con el Prof A//   else     //Interactua con el Prof B//   P(esperando[i]) } </pre>	