

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. 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.
2. 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.
3. Existen N personas que deben fotocopiar un documento cada una. Resolver cada ítem usando semáforos:
 - a) Implemente una solución suponiendo que existe una única fotocopiadora compartida por todas las personas, y las mismas la deben usar de a una persona a la vez, sin importar el orden. Existe una función *Fotocopiar()* llamada por la persona que simula el uso de la fotocopiadora. Sólo se deben usar los procesos que representan a las *Personas*.
 - b) Modifique la solución de (a) para el caso en que se deba respetar el orden de llegada.
 - c) Modifique la solución de (b) para el caso en que se deba dar prioridad de acuerdo a la edad de cada persona (cuando la fotocopiadora está libre la debe usar la persona de mayor edad entre las que estén esperando para usarla).
 - d) Modifique la solución de (b) para el caso en que además haya un *Empleado* que le indica a cada persona cuando debe usar la fotocopiadora.
 - e) Modificar la solución (e) para el caso en que sean 10 fotocopiadoras. El *empleado* le indica a la persona cuando puede usar una fotocopiadora, y cual debe usar.
4. Se debe simular una maratón con **C corredores** donde en la llegada hay UNA máquina expendedoras de agua con capacidad para 20 botellas. Además, existe **un repositor** encargado de reponer las botellas de la máquina. Cuando los **C corredores** han llegado al inicio comienza la carrera. Cuando un corredor termina la carrera se dirigen a la máquina expendedora, espera su turno (respetando el orden de llegada), saca una botella y se retira. Si encuentra la máquina sin botellas, le avisa al *repositor* para que cargue nuevamente la máquina con 20 botellas; espera a que se haga la recarga; saca una botella y se retira. **Nota:** maximizar la concurrencia; mientras se reponen las botellas se debe permitir que otros corredores se encolen.

5. Una empresa de turismo posee **4 combis** con capacidad para 25 personas cada una y **UN vendedor** que vende los pasajes a los clientes de acuerdo al orden de llegada. Hay **C clientes** que al llegar intentan comprar un pasaje para una combi en particular (el cliente conoce este dato); si aún hay lugar en la combi seleccionada se le da el pasaje y se dirige hacia la combi; en caso contrario se retira. Cada combi espera a que suban los 25 pasajeros, luego realiza el viaje, y cuando llega al destino deja bajar a todos los pasajeros. **Nota:** maximizar la concurrencia; suponga que para cada combi al menos 25 clientes intentarán comprar pasaje.
6. A una cerealera van T camiones a descargarse trigo y M camiones a descargar maíz. Sólo hay lugar para que 7 camiones a la vez descarguen, pero no pueden ser más de 5 del mismo tipo de cereal. **Nota:** sólo se pueden usar procesos que representen a los camiones.
7. En un examen final hay P alumnos y 3 profesores. Cuando todos los alumnos han llegado comienza el examen. Cada alumno resuelve su examen, lo entrega y espera a que alguno de los profesores lo corrija y le indique la nota. Los profesores corrigen los exámenes respetando el orden en que los alumnos van entregando. **Nota:** todos los procesos deben terminar.
8. 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 se queda esperando el puntaje del grupo. Cuando 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).
9. Simular la atención en una Terminal de Micros que posee 3 puestos para hisopar a **150 pasajeros**. En cada puesto hay **una Enfermera** que atiende a los pasajeros de acuerdo al orden de llegada al mismo. Cuando llega un pasajero se dirige al puesto que tenga menos gente esperando. Espera a que la enfermera correspondiente lo llame para hisoparlo, y luego se retira. **Nota:** suponer que existe una función *Hisopar()* que simula la atención del pasajero por parte de la enfermera correspondiente.
10. En una empresa de software hay **3 programadores** que deben arreglar errores informados por **N clientes**. Los clientes continuamente están trabajando, y cuando encuentran un error envían un reporte a la empresa para que lo corrija (no tienen que esperar a que se resuelva). Los programadores resuelven los reclamos de acuerdo al orden de llegada, y si no hay reclamos pendientes trabajan durante una hora en otros programas. **Nota:** los procesos no deben terminar (trabajan en un loop infinito); suponga que hay una función *ResolverError* que simula que un programador está resolviendo un reporte de un cliente, y otra *Programar* que simula que está trabajando en otro programa.