

Práctica 3 – Monitores

CONSIDERACIONES PARA RESOLVER LOS EJERCICIOS:

- Los monitores utilizan el protocolo *signal and continue*.
- A una variable *condition* SÓLO pueden aplicársele las operaciones SIGNAL, SIGNALALL y WAIT.
- NO puede utilizarse el *wait con prioridades*.
- NO se puede utilizar ninguna operación que determine la cantidad de procesos encolados en una variable *condition* o si está vacía.
- La única forma de comunicar datos entre monitores o entre un proceso y un monitor es por medio de invocaciones al procedimiento del monitor del cual se quieren obtener (o enviar) los datos.
- No existen variables globales.
- En todos los ejercicios debe maximizarse la concurrencia.
- En todos los ejercicios debe aprovecharse al máximo la característica de exclusión mutua que brindan los monitores.
- Debe evitarse hacer *busy waiting*.
- En todos los ejercicios el tiempo debe representarse con la función *delay*.

1. Se dispone de un puente por el cual puede pasar un solo auto a la vez. Un auto pide permiso para pasar por el puente, cruza por el mismo y luego sigue su camino.

Monitor Puente

```

cond cola;
int cant = 0;

Procedure entrarPuente ()
    while ( cant > 0 ) wait (cola);
    cant = cant + 1;
end;

Procedure salirPuente ()
    cant = cant - 1;
    signal(cola);
end;

```

End Monitor;**Process Auto [a:1..M]**

```

Puente. entrarPuente (a);
“el auto cruza el puente”
Puente. salirPuente(a);

```

End Process;

- a. ¿El código funciona correctamente? Justifique su respuesta.
- b. ¿Se podría simplificar el programa? ¿Sin monitor? ¿Menos procedimientos? ¿Sin variable condition? En caso afirmativo, rescriba el código.
- c. ¿La solución original respeta el orden de llegada de los vehículos? Si rescribió el código en el punto b), ¿esa solución respeta el orden de llegada?

2. Existen N procesos que deben leer información de una base de datos, la cual es administrada por un motor que admite una cantidad limitada de consultas simultáneas.
 - a) Analice el problema y defina qué procesos, recursos y monitores/sincronizaciones serán necesarios/convenientes para resolverlo.
 - b) Implemente el acceso a la base por parte de los procesos, sabiendo que el motor de base de datos puede atender a lo sumo 5 consultas de lectura simultáneas.
3. Existen N personas que deben fotocopiar un documento. La fotocopidora sólo puede ser usada por una persona a la vez. Analice el problema y defina qué procesos, recursos y monitores serán necesarios/convenientes, además de las posibles sincronizaciones requeridas para resolver el problema. Luego, resuelva considerando las siguientes situaciones:
 - a) Implemente una solución suponiendo no importa el orden de uso. Existe una función *Fotocopiar()* que simula el uso de la fotocopidora.
 - 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 con la edad de cada persona (cuando la fotocopidora está libre la debe usar la persona de mayor edad entre las que estén esperando para usarla).
 - d) Modifique la solución de (a) para el caso en que se deba respetar estrictamente el orden dado por el identificador del proceso (la persona X no puede usar la fotocopidora hasta que no haya terminado de usarla la persona X-1).
 - e) Modifique la solución de (b) para el caso en que además haya un Empleado que le indica a cada persona cuando debe usar la fotocopidora.
 - f) Modificar la solución (e) para el caso en que sean 10 fotocopadoras. El empleado le indica a la persona cuál fotocopidora usar y cuándo hacerlo.
4. Existen N vehículos que deben pasar por un puente de acuerdo con el orden de llegada. Considere que el puente no soporta más de 50000kg y que cada vehículo cuenta con su propio peso (ningún vehículo supera el peso soportado por el puente).
5. En un corralón de materiales se deben atender a N clientes de acuerdo con el orden de llegada. Cuando un cliente es llamado para ser atendido, entrega una lista con los productos que comprará, y espera a que alguno de los empleados le entregue el comprobante de la compra realizada.
 - a) Resuelva considerando que el corralón tiene un único empleado.
 - b) Resuelva considerando que el corralón tiene E empleados ($E > 1$). Los empleados no deben terminar su ejecución.
 - c) Modifique la solución (b) considerando que los empleados deben terminar su ejecución cuando se hayan atendido todos los clientes.
6. Existe una comisión de 50 alumnos que deben realizar tareas de a pares, las cuales son corregidas por un JTP. Cuando los alumnos llegan, forman una fila. Una vez que están todos en fila, el JTP les asigna un número de grupo a cada uno. Para ello, suponga que existe una función *AsignarNroGrupo()* que retorna un número “aleatorio” del 1 al 25. Cuando un alumno

ha recibido su número de grupo, comienza a realizar su tarea. Al terminarla, el alumno le avisa al JTP y espera por su nota. Cuando los dos alumnos del grupo completaron la tarea, el JTP les asigna un puntaje (el primer grupo en terminar tendrá como nota 25, el segundo 24, y así sucesivamente hasta el último que tendrá nota 1). **Nota:** el JTP no guarda el número de grupo que le asigna a cada alumno.

7. 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:** mientras se reponen las botellas se debe permitir que otros corredores se encolen.
8. En un entrenamiento de fútbol hay 20 jugadores que forman 4 equipos (cada jugador conoce el equipo al cual pertenece llamando a la función *DarEquipo()*). Cuando un equipo está listo (han llegado los 5 jugadores que lo componen), debe enfrentarse a otro equipo que también esté listo (los dos primeros equipos en juntarse juegan en la cancha 1, y los otros dos equipos juegan en la cancha 2). Una vez que el equipo conoce la cancha en la que juega, sus jugadores se dirigen a ella. Cuando los 10 jugadores del partido llegaron a la cancha comienza el partido, juegan durante 50 minutos, y al terminar todos los jugadores del partido se retiran (no es necesario que se esperen para salir).
9. En un examen de la secundaria hay *un preceptor* y *una profesora* que deben tomar un examen escrito a 45 alumnos. El preceptor se encarga de darle el enunciado del examen a los alumnos cuando los 45 han llegado (es el mismo enunciado para todos). La profesora se encarga de ir corrigiendo los exámenes de acuerdo con el orden en que los alumnos van entregando. Cada alumno al llegar espera a que le den el enunciado, resuelve el examen, y al terminar lo deja para que la profesora lo corrija y le envíe la nota. **Nota:** maximizar la concurrencia; todos los procesos deben terminar su ejecución; suponga que la profesora tiene una función *corregirExamen* que recibe un examen y devuelve un entero con la nota.
10. En un parque hay un juego para ser usada por N personas de a una a la vez y de acuerdo al orden en que llegan para solicitar su uso. Además, hay *un empleado* encargado de desinfectar el juego durante 10 minutos antes de que una persona lo use. Cada persona al llegar espera hasta que el empleado le avisa que puede usar el juego, lo usa por un tiempo y luego lo devuelve. **Nota:** suponga que la persona tiene una función *Usar_juego* que simula el uso del juego; y el empleado una función *Desinfectar_Juego* que simula su trabajo. Todos los procesos deben terminar su ejecución.