Sistemas Operativos
Práctico 4

Curso 2024

Objetivos

- Resolver problemas complejos con el uso de semáforos.
- Familiarizarse con el uso de monitores.

Duración

- 1 semana.

Ejercicio 1 (OpenFing) Se desea controlar el proceso de llenado y tapado de envases de la Compañía Algarrobo Cola. Existe una sola cinta transportadora y una máquina para cada acción. Se dispone de los siguientes procedimientos:

- **avanzo_cinta()** : Avanza la cinta un paso (muy pequeño).
- **llenar_botella()** : Llena la botella de líquido (finaliza cuando está llena).
- **pongo_tapa()** : Comienza la acción de tapado (finaliza cuando la tapa queda colocada).

Y de las siguientes funciones:

- **puedo_llenar()** : Devuelve TRUE si hay un envase bajo la estación de llenado.
- **puedo_tapar()** : Ídem para la estación de tapado.

Se pide:

(a) Resolver utilizando **semáforos**.

(b) Resolver utilizando **monitores**.

Nota: Las botellas vienen a intervalos irregulares y no siempre se está en situación de accionar las dos máquinas a la vez. Se desea conseguir los algoritmos más sencillos posibles que resuelvan correctamente el problema.

Ejercicio 2 (medio) Se tiene un controlador para entrar en una sala de servidores. A la sala entran a realizar mantenimiento tanto técnicos de software (TS) como de hardware (TH) y sólo pueden entrar 5 personas al mismo tiempo. Los TH tienen prioridad para entrar por sobre los TS, sin embargo dentro de cada grupo no existe ningún orden especificado.

Se pide: implemente utilizando **semáforos** el controlador descrito. Debe implementar las tareas TS y TH. Dispone de la función: `realizar_mantenimiento()`.

Ejercicio 3 (avanzado) Se desea modelar con **semáforos** las tareas VENDEDOR, CLIENTE y SUPERVISOR de una cafetería. Esta cafetería dispone de una caja, 8 vendedores y 2 supervisores. Los vendedores toman el pedido, cobran y elaboran el pedido de cada cliente. Tienen orden de no dejar la caja sola, por lo que para poder ir a elaborar el pedido debe haber otro vendedor en el área de la caja. Por razones de espacio no puede haber más de 2 vendedores en el área de caja a la vez, de los cuales solo uno de ellos puede estar atendiendo. Los vendedores solo toman pedidos cuando están en la caja. Los vendedores deben atender a los clientes lo antes posible, no se debe hacer esperar a un cliente si la caja está libre.

Los vendedores le avisarán al grupo de supervisores cada 10 pedidos cobrados por ese vendedor. Los supervisores estarán esperando ser avisados para llenar la planilla. El primer supervisor libre recibirá el número de vendedor y con ese dato llenará la planilla.

Se dispone de los siguientes procedimientos:

- **recibir_pedido()**, **cobrar_pedido()**, **elaborar_pedido()**. Ejecutados por el vendedor.
- **enviar_pedido_pagar_y_recibir_pedido()**. Ejecutado por el cliente, retorna cuando el vendedor termina de elaborar el pedido.
- **llenar_planilla(int nro_vendedor)**. Ejecutado por el supervisor actualiza la planilla de ventas.

Aclaraciones:

- No se permite la implementación de tareas auxiliares.
- Todos los procedimientos disponibles pueden demorar tiempos variables.

Ejercicio 4 (medio) Se desea modelar una piscina pública. Dadas las restricciones por COVID solo se permite hasta 25 personas bañándose simultáneamente. Luego de que está llena, las personas que llegan a continuación deben esperar afuera hasta que se libere un lugar. Cada cierto tiempo un empleado debe aspirar la basura del fondo de la piscina y para ello la misma debe estar vacía. Las personas que se bañan tienen prioridad para entrar sobre el empleado que debe esperar a que nadie quiera usar la piscina.

Se pide: Implementar las personas y el empleado usando **semáforos**. Se dispone de las siguientes funciones auxiliares:

- **bañarse()**: Ejecutada por las personas para usar la piscina
- **limpiar()**: Ejecutada por el empleado para aspirar el fondo de la piscina

Ejercicio 5 (avanzado) Se desea modelar una construcción de un edificio. En la misma participan 4 obreros los cuales levantan paredes obteniendo ladrillos desde un pallet que contiene 300 ladrillos. Los obreros toman de a 3 ladrillos, los colocan en la pared y vuelven a tomar ladrillos nuevos.

Cuando un obrero detecta que no hay suficientes ladrillos le avisa al operador de la grúa para que se coloque un nuevo pallet y espera a que haya uno nuevo para tomar los ladrillos. El pallet solo se podrá cambiar mientras no haya obreros sacando ladrillos.

Cada cierto tiempo el capataz de la obra revisará los ladrillos consumidos. Para eso deberá tener acceso exclusivo al pallet y tendrá prioridad sobre los empleados y la grúa.

Se pide: Implementar usando **semáforos** a los obreros, el operador de la grúa y el capataz. No se pueden usar tareas auxiliares.

Se dispone de las siguientes funciones auxiliares:

- **tomar_ladrillos()**: Ejecutada por los obreros para sacar 3 ladrillos del pallet
- **colocar_ladrillos()**: Ejecutada por los obreros para agregar los ladrillos a la pared
- **cambiar_pallet()**: Ejecutada por el operador de la grúa para cambiar el pallet
- **contar_ladrillos()**: Ejecutada por el capataz para revisar los ladrillos consumidos
- **otras_tareas()**: Ejecutada por el capataz cuando no revisa los ladrillos

Ejercicio 6 (avanzado) Se desea modelar un depósito de una compañía de juguetes. El depósito cuenta con 2 sectores independientes uno de los cuales se usa para guardar juguetes (sector 1) y el otro para guardar fuegos artificiales (sector 2).

En el depósito trabajan 10 empleados los cuales reciben pedidos de artículos para sacar del depósito los cuales deben retirar y colocar en un camión que se encuentra estacionado en la puerta. En el sector de juguetes puede haber hasta cinco trabajadores simultáneamente mientras que en el sector de fuegos artificiales solo puede entrar uno por vez.

El camión solo puede contener a lo sumo 100 artículos por lo que si está lleno los empleados deben esperar a que se retire el camión y venga otro. Mientras no está lleno puede haber varios empleados guardando artículos simultáneamente en el camión. El primer empleado que detecta que el camión

está lleno debe avisarle al conductor para que se lleve el camión. El mismo conductor volverá luego con un camión vacío y le avisará a los empleados que estén esperando por guardar artículos que ya pueden hacerlo.

Cada cierto tiempo el supervisor realiza un control de stock sobre el depósito (de a un sector por vez) y requiere acceso exclusivo al sector que está controlando. En el sector de juguetes tiene prioridad sobre los empleados mientras que en el de fuegos artificiales no hay prioridad. De haber empleados esperando por la llegada de un nuevo camión el supervisor puede solicitarle a uno de ellos que lo ayude con el trabajo de supervisión.

Se pide: Modelar los empleados, el conductor y el supervisor usando **semáforos**.

Se dispone de las siguiente funciones que son ejecutadas por los empleados:

- **recibir_pedido():integer** (retorna si corresponde al sector 1 o 2)
- **retirar_pedido()**
- **guardar_pedido()**
- **ayudar_supervisor()**

Se dispone de la siguiente función que será ejecutada por el conductor:

- **cambiar_camión()**

Se dispone de las siguientes funciones que serán ejecutadas por el supervisor:

- **que_supervisar():integer** (retorna si corresponde al sector 1 o 2)
- **elegir_ayudante(array[1..10] of boolean):integer** (recibe que trabajadores están esperando y elije uno)
- **supervisar()**
- **otras_tareas()**

Ejercicio 7 (medio) Se quiere desarrollar un software que permita distribuir el trabajo entre dos grupos de TI, desarrollo y verificación. Los integrantes del grupo de desarrollo implementan módulos de software y en la medida que culminan dejan los módulos para su verificación. El grupo de verificación valida, en la medida que tenga módulos disponible, los módulos. Por reglas de la empresa no se puede tener más de 20 módulos disponibles para verificar. En caso de llegar a esta situación, el jefe pasará todos los módulos disponibles a un centro dedicado a la temática y esta es su única tarea. El software además, debe llevar los indicadores de producción, cuántos módulos se implementaron, cuántos se validaron positivamente, cuántos negativamente y cuantos se pasaron al centro especializado.

Se dispone de los siguientes procedimientos y funciones:

- **implementar()** : Función de los desarrolladores, devuelve un módulo.
- **MLPV(m)** : Procedimiento que inserta el módulo `m` en la estructura de módulos listos para verificar.
- **verificar(m)** : Función que verifica el módulo `m`. Devuelve verdadero si se valido el módulo y falso en caso contrario.
- **obtener()** : Función que devuelve el primer módulo disponible para verificar. Si no hay módulos la función falla.
- **a_centro(m)** : Procedimiento invocado por el jefe que entrega al centro especializado el módulo `m` a verificar.

Se pide: implemente con **monitores** la realidad anterior. Debe incluir los procedimientos Desarrollador, Tester y Jefe. Además debe implementar una función que retorne cada uno de los indicadores en el orden definido en la letra.

Ejercicio 8 (OpenFing) Una tribu de N caníbales come de una gran marmita común con capacidad para 6 comensales simultáneos. Cuando un comensal quiere comer, come de la marmita, a menos que no haya suficiente comida para él. Si no hay suficiente comida en la marmita, el caníbal despierta al cocinero y espera a que el cocinero haya rellenado la marmita con la carne de los misioneros capturados (no debe haber notificaciones repetidas). Para rellenar la marmita el cocinero debe esperar a que todos los comensales que se encuentran actualmente comiendo terminen. El cocinero, por su parte, vuelve a dormir cuando ha rellenado la marmita. Consideraciones:

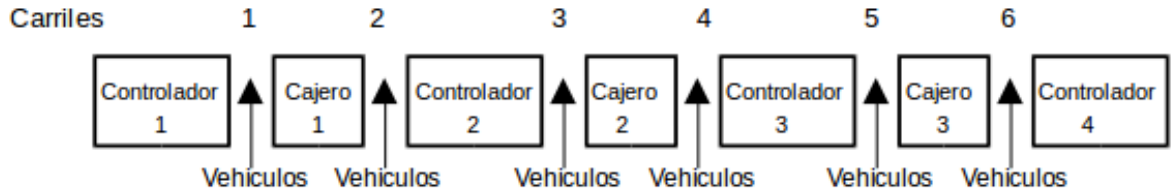
- No podrán entrar nuevos comensales a la marmita cuando el cocinero está rellenando o esperando para rellenar.
- Se supone que la marmita llena dispone de comida para más de seis caníbales.

Se dispone de las siguientes funciones:

- **Hay_suficiente_comida():boolean**
  Esta función es ejecutada por los comensales y retorna si hay suficiente comida en la marmita. No puede ser ejecutada por dos o más comensales a la vez.
- **Rellenar()**
  Esta función es ejecutada por el cocinero.
- **Comer()**
  Es ejecutado por los comensales.
- **Ocio()**
  Ejecutada por los caníbales cuando no están comiendo.

Se pide: Implementar los procedimientos caníbal y cocinero utilizando **monitores**. Especifique la semántica de los mismos en caso de ser necesario.

Ejercicio 9 (avanzado) Se desea modelar un peaje de una carretera de una sola vía. El peaje además de cobrar el ticket controla los documentos de los autos y camiones. Los puestos están distribuidos de esta manera:



Las cajas y los controladores atenderán para los dos carriles que tienen a su costado, salvo las de las puntas que solamente atienden autos del carril que tienen al lado. Una vez que un cajero atendió a un vehículo, no podrá atender a otro hasta que el controlador termine de controlar dicho vehículo, y viceversa. Por otro lado los camiones solo podrán pasar por los carriles de las puntas. Un inspector controlara cada cierto tiempo a estos controladores (los de las puntas), los que no podrán atender ningún vehículo mientras están siendo controlados y viceversa.

El control por parte del inspector a estos controladores se hará con los 2 a la vez, es decir que ninguno de los 2 deberá estar controlando vehículos, debiendo obtener primero la atención del controlador 4 antes que el controlador 1. Por razones de ordenamiento y comodidad, los vehículos siempre pedirán ser atendidos primero por el puesto de su izquierda y luego por el de la derecha.

Se pide: Modelar este sistema usando **monitores**. Se dispone de las siguientes funciones o procedimientos:

- **pagar()** — paga o cobra al vehículo al cajero.
- **controlar_documentacion()** — control de documentación del controlador al vehículo.
- **inspeccion()** — control del inspector a los controladores de las puntas.
- **dame_carril(tipo_vehículo:Integer)** — retorna a que carril debe ir el vehículo, dependiendo si es auto o camión, el cual será ejecutado por el vehículo.
- **que_soy()** — retorna 0 si es un auto o 1 si es un camión.

Nota: Se podrá utilizar como máximo una tarea auxiliar.