

## Práctica 4 – Pasaje de Mensajes

### CONSIDERACIONES PARA RESOLVER LOS EJERCICIOS DE PMA:

- Los canales son compartidos por todos los procesos.
- Cada canal es una cola de mensajes, por lo tanto el primer mensaje encolado es el primero en ser atendido.
- Por ser pasaje de mensajes asincrónico el *send* no bloquea al emisor.
- Se puede usar la sentencia *empty* para saber si hay algún mensaje en el canal, pero no se puede consultar por la cantidad de mensajes encolados.
- Se puede utilizar el *if/do* no determinístico donde cada opción es una condición booleana donde se puede preguntar por variables locales y/o por *empty* de canales.  
if (cond 1) → Acciones 1;  
□ (cond 2) → Acciones 2;  
....  
□ (cond N) → Acciones N;  
end if  
De todas las opciones cuya condición sea Verdadera elige una en forma no determinística y ejecuta las acciones correspondientes. Si ninguna es verdadera sale del if/do si hacer nada.
- Se debe tratar de evitar hacer *busy waiting* (sólo hacerlo si no hay otra opción).
- En todos los ejercicios el tiempo debe representarse con la función *delay*.

1. Supongamos que tenemos una *abuela* que tiene dos tipos de lápices para dibujar: **10 de colores y 15 negros**. Además tenemos tres clases de niños que quieren dibujar con los lápices: los que quieren usar sólo los lápices de colores (*tipo C*), los que usan sólo los lápices negros (*tipo N*), y los niños que usan cualquier tipo de lápiz (*tipo A*).
  - a) Implemente un código para cada clase de niño de manera que ejecute pedido de lápiz, lo use por 10 minutos y luego lo devuelva y además el proceso abuela encargada de asignar los lápices.
  - b) Modificar el ejercicio para que a los niños de tipo A se les puede asignar un lápiz sólo cuando: hay lápiz negro disponible y ningún pedido pendiente de tipo N, o si hay lápiz de color disponible y ningún pedido pendiente de tipo C.**Nota:** se deben modelar los procesos *niño* y el proceso *abuela*.
2. Se desea modelar el funcionamiento de un banco en el cual existen 5 cajas para realizar pagos. Existen **P personas** que desean pagar. Para esto cada una selecciona la caja donde hay menos personas esperando, una vez seleccionada espera a ser atendido. **Nota:** maximizando la concurrencia, deben usarse los valores actualizados del tamaño de las colas para seleccionar la caja con menos gente esperando.

3. Se debe modelar una casa de Comida Rápida, en el cual trabajan **2 cocineros** y **3 vendedores**. Además hay **C clientes** que dejan un pedido y quedan esperando a que se lo alcancen.

Los pedidos que hacen los clientes son tomados por cualquiera de los vendedores y se lo pasan a los cocineros para que realicen el plato. Cuando no hay pedidos para atender, los vendedores aprovechan para reponer un pack de bebidas de la heladera (tardan entre 1 y 3 minutos para hacer esto).

Repetidamente cada cocinero toma un pedido pendiente dejado por los vendedores, lo cocina y se lo entrega directamente al cliente correspondiente.

**Nota:** maximizar la concurrencia.

4. Se desea modelar una competencia de atletismo. Para eso existen dos tipos de procesos: **C corredores** y **un portero**. Los *corredores* deben esperar que se habilite la entrada a la pista, donde deben esperar que lleguen todos los corredores para comenzar. El *portero* es el encargado de habilitar la entrada a la pista.

a) Implementar usando un **coordinador**.

b) Implementar sin usar un coordinador.

**NOTAS:** el proceso *portero* NO puede contabilizar nada, su única función es habilitar la entrada a la pista; NO se puede suponer ningún orden en la llegada de los *corredores* al punto de partida.

5. Suponga que **N personas** llegan a la cola de un banco. Una vez que la persona se agrega en la cola no espera más de 15 minutos para su atención, si pasado ese tiempo no fue atendida se retira. Para atender a las personas existen **2 empleados** que van atendiendo de a una y por orden de llegada a las personas.

6. 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.

a) Implementar una solución utilizando un proceso intermedio entre cada persona y el empleado.

b) Implementar una solución sin utilizar un proceso intermedio entre cada persona y el empleado.

**CONSIDERACIONES PARA RESOLVER LOS EJERCICIOS DE PMS:**

- Los canales son punto a punto y no deben declararse.
- No se puede usar la sentencia *empty* para saber si hay algún mensaje en un canal.
- Tanto el envío como la recepción de mensajes es bloqueante.
- Sintaxis de las sentencias de envío y recepción:

*Envío:* nombreProcesoReceptor!port (datos a enviar)

*Recepción:* nombreProcesoEmisor?port (datos a recibir)

El port (o etiqueta) puede no ir. Se utiliza para diferenciar los tipos de mensajes que se podrían comunicarse entre dos procesos.

- En la sentencia de comunicación de recepción se puede usar el comodín \* si el origen es un proceso dentro de un arreglo de procesos. Ejemplo: Clientes[\*]?port(datos).
- Sintaxis de la Comunicación guardada:

*Guarda:* (condición booleana); sentencia de recepción → sentencia a realizar

Si no se especifica la condición booleana se considera verdadera (la condición booleana sólo puede hacer referencia a variables locales al proceso).

Cada guarda tiene tres posibles estados:

**Elegible:** la condición booleana es verdadera y la sentencia de comunicación se puede resolver inmediatamente.

**No elegible:** la condición booleana es falsa.

**Bloqueada:** la condición booleana es verdadera y la sentencia de comunicación no se puede resolver inmediatamente.

Sólo se puede usar dentro de un if o un **do** guardado:

El **IF** funciona de la siguiente manera: de todas las guardas **elegibles** se selecciona una en forma no determinística, se realiza la sentencia de comunicación correspondiente, y luego las acciones asociadas a esa guarda. Si todas las guardas tienen el estado de **no elegibles**, se sale sin hacer nada. Si no hay ninguna guarda elegible, pero algunas están en estado **bloqueado**, se queda esperando en el if hasta que alguna se vuelva elegible.

El **DO** funciona de la siguiente manera: sigue iterando de la misma manera que el **if** hasta que todas las guardas hasta que todas las guardas sean **no elegibles**.

7. En una estación de comunicaciones se cuenta con **10 radares** y una **unidad de procesamiento** que se encarga de procesar la información enviada por los radares. Cada radar repetidamente detecta señales de radio durante 15 segundos y le envía esos datos a la unidad de procesamiento para que los analice. Los radares no deben esperar a ser atendidos para continuar trabajando.

**Nota:** maximizar la concurrencia.

8. Supongamos que tenemos una **abuela** que tiene dos tipos de lápices para dibujar: **10 de colores** y **15 negros**. Además tenemos tres clases de niños que quieren dibujar con los lápices: los que quieren usar sólo los lápices de colores (**tipo C**), los que usan sólo los lápices negros (**tipo N**), y los niños que usan cualquier tipo de lápiz (**tipo A**).
- a) Implemente un código para cada clase de niño de manera que ejecute pedido de lápiz, lo use por 10 minutos y luego lo devuelva y además el proceso abuela encargada de asignar los lápices.
  - b) Modificar el ejercicio para que a los niños de tipo A se les puede asignar un lápiz sólo cuando: hay lápiz negro disponible y ningún pedido pendiente de tipo N, o si hay lápiz de color disponible y ningún pedido pendiente de tipo C.
- Nota:** se deben modelar los procesos **niño** y el proceso **abuela**.
9. Se debe modelar la atención en una panadería por parte de 3 empleados. Hay *C* clientes que ingresan al negocio para ser atendidos por cualquiera de los empleados, los cuales deben atenderse de acuerdo al orden de llegada.
- Nota:** maximizar la concurrencia.
10. 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.
- a) Implementar una solución utilizando un proceso intermedio entre cada persona y el empleado.
  - b) Implementar una solución sin utilizar un proceso intermedio entre cada persona y el empleado.