

# Programación Concurrente – 2015

## Práctica N°4

### Pasaje de Mensajes Sincrónico – Asíncrono

**Nota:** En todos los ejercicios el tiempo debe representarse con la función delay

#### CONSIDERACIONES PARA RESOLVER LOS EJERCICIOS:

##### 1- Para el pasaje de mensajes asíncrono:

- a. Sintaxis declaración:  
chan nombrecanal (tipoDato)  
chan nombrecanal[1..m](tipoDato)
- b. Sintaxis uso de canales  
Proceso uno[i:1..n]  
    Send nombreCanal(datos a enviar)  
    Receive nombreCanal (datos a recibir)  
    Send nombreCanal[i](datos a enviar)  
    Receive nombreCanal[i] (datos a recibir)  
    If (empty(nombreCanal)) then ....
- c. Los canales son globales.
- d. Cada canal es una cola de mensajes, por lo tanto el primer mensaje encolado es el primero en ser atendido.
- e. Por ser pasaje de mensajes asíncrono el send no bloquea al emisor.
- f. No se puede preguntar por la cantidad de mensajes encolados.
- g. Se puede utilizar el if/do no determinístico donde cada opción es una condición booleana donde se puede preguntar por el EMPTY de los 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.

## 2-Para el pasaje de mensajes sincrónico:

### a. Sintaxis declaración:

Los canales en PMS no se declaran porque directamente se nombra el proceso con el cual se quiere comunicar.

### b. Sintaxis uso de canales

Proceso uno[i:1..n]

nombreProcesoReceptor!port (datos a enviar)

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.

Ejemplo:

Si el Proceso1 se quiere comunicar con el Proceso2 entonces:

Proceso2!port(datos a enviar)

Si el Proceso1 se quiere comunicar con un proceso de tipo 2 (existe un arreglo de procesos 2) entonces:

ProcesoTipo2[j]!port(datos a enviar) {j es la posición en el arreglo de procesos de tipo 2 con el que me quiero comunicar}

Se puede usar el if/do guardado

If (Guarda 1) -> Acciones 1

□ (Guarda 2) -> Acciones 2

...

□ (Guarda 3) -> Acciones 3

End if

Cada Guarda de un if/do se compone de las siguientes partes:

cond.booleana; sent. de comunicación de recep

donde:

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

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).

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.

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 del if sin hacer nada. Si no hay ninguna guarda elegible, pero algunas estas en estado bloqueado, se queda esperando en el if hasta que alguna de esas se vuelva elegible.

El DO funciona de la siguiente manera: sigue iterando de la misma manera que el if hasta que todas las guardas son FALSAS. En ese momento sale del DO.

1. Supongamos que tenemos una abuela que tiene dos tipos de lápices para dibujar, de colores y lápices 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). 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. Se deben modelar al menos los procesos niño, el proceso abuela y los procesos lápices.
  - a) Implementar utilizando pasaje de mensajes asíncrono (PMA).
  - b) Implementar utilizando pasaje de mensajes síncrono (PMS).
  - c) Modificar el ejercicio original de la siguiente manera y resolverlo con PMS. 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.
2. Se desea modelar una competencia de atletismo con PMA. 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 proceso extra coordinador.
  - b. Implementar sin usar ningún proceso extra.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.
3. Se desea modelar con PMA 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.
4. Resolver con PMA. En un edificio existen 3 porteros y P personas. Las personas dejan reclamos en la oficina de los porteros que deben ser atendidos por cualquiera de ellos. Cada portero está continuamente trabajando, si hay algún reclamo pendiente lo atiende, sino realiza un recorrido por los pisos durante 10 minutos. NOTA: la persona no debe esperar a que el reclamo sea atendido, ni se le debe avisar que se resolvió.
5. Se debe modelar con PMA la descarga de cereales en una acopiadora. Para esto existen N camiones que deben descargar su carga. Los camiones se descargan de a uno por vez, y de acuerdo al orden de llegada. Una vez que el camión llegó, espera a lo sumo 2 hs. a que le avisan que es su turno para descargar, sino se retira sin realizar la descarga. Existe además un empleado que es el encargado de avisar a cada camión cuando es su turno. NOTAS: existe una función Descarga que es llamada por el camión que representa el tiempo que tarda en descargar su contenido el camión.
6. Se debe modelar una casa de Comida Rápida, en el cual trabajan DOS cocineros y TRES vendedores. Además hay C clientes que dejan un pedido y quedan esperando a que se lo alcancen. Los pedidos que dejan 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. Modelice utilizando PMA. Maximice la concurrencia.
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. Modelice utilizando PMS.
8. Resolver con PMS. 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. Implementar maximizando la concurrencia.