

# 1)

1- Defina y diferencie programa concurrente, programa distribuido y programa paralelo.

## Programa Concurrente

- Una sola unidad de procesamiento, dedica una parte del tiempo a cada una de las tareas (concurrency sin paralelismo de hardware)

## Programa Distribuido

- Programa concurrente comunicado por mensajes
- Supone la ejecución sobre una arquitectura de memoria distribuida, aunque puede ejecutarse sobre una de memoria compartida o híbrida

## Programa Paralelo

- Varias unidades de procesamiento, cada una dedicándose a una parte de la tarea y donde todas están trabajando al mismo tiempo

# 2)

2- Marque al menos 2 similitudes y 2 diferencias entre los pasajes de mensajes sincrónicos y asincrónicos. Indicar cual es la principal ventaja de pasaje de mensajes sincrónicos respecto a pasaje de mensajes asincrónicos.

## Similitudes

- Usan canales unidireccionales
- Funcionan para envío de mensajes en memoria distribuida

## Diferencias

### PMS

- Comunicación tipo link
- Hay una sincronía entre el emisor y el receptor, donde ambos esperan tanto en el envío como en la recibida de los mensajes que el otro confirme

### PMA

- Comunicación tipo mailbox
- Es asincrónico, es decir, el que envía no necesita esperar una confirmación de que el mensaje fue recibido, aunque el que recibe espera la recepción de un mensaje

## Ventaja de PMS sobre PMA

- Permite sincronizar automáticamente a emisor y receptor, garantizando que el mensaje fue recibido y facilitando la coordinación entre procesos.
- Menor tamaño del buffer de mensajes (1)

### 3)

- 3- Analice qué tipo de mecanismos de pasaje de mensajes son más adecuados para resolver problemas de tipo Cliente/Servidor, Pares que interactúan, Filtros, y Productores y Consumidores. Justifique claramente su respuesta.

#### Cliente/Servidor

- RPC o Rendezvous
  - Ambos son bidireccionales y sincrónicos, para el caso donde el cliente necesita la respuesta del servidor
  - Es de los más adecuados
- PMS
  - Disminuye la concurrencia, pero es más adecuado para el caso donde el cliente necesita la respuesta del servidor
  - Más simple de implementar
  - Adecuado cuando el cliente debe esperar la respuesta para continuar
- PMA
  - Permite mayor concurrencia cuando el cliente no necesita la respuesta inmediatamente
  - Aumenta la concurrencia pero complejiza la lógica al recibir la respuesta más tarde

#### Pares que interactúan

- PMS
  - Permite sincronizar fácilmente en ambas direcciones
- Comunicación guardada
  - Es una solución eficiente para la interacción entre pares, permite a los procesos esperar selectivamente mensajes de diferentes fuentes, lo que facilita la programación de interacciones complejas
- PMA
  - Requiere mecanismos adicionales para garantizar la sincronización y evitar condiciones de carrera

#### Filtros

- PMA
  - Permite mayor concurrencia, ya que cada filtro puede trabajar en su parte del flujo de datos sin esperar a que los demás terminen
  - Los filtros forman un "flujo de datos continuo"

#### Productores y Consumidores

- PMA
  - Permite mayor concurrencia, ya que el productor puede seguir generando datos sin esperar a que sean consumidos
  - Permite desacoplar las velocidades del productor y consumidor

- PMS
  - Reduce la concurrencia, pero puede ser útil para controlar el flujo de datos entre el productor y el consumidor, asegurando que el productor no genere datos más rápido de lo que el consumidor los consume

## 4)

- 4- Indique por qué puede considerarse que existe una dualidad entre los mecanismos de monitores y pasaje de mensajes. Ejemplifique

Se puede considerar que existe una dualidad entre los mecanismos de monitores y pasaje de mensajes porque cada uno puede simular al otro y utilizarse para resolver los mismos problemas. La preferencia en el uso de monitores o pasaje de mensajes depende de la arquitectura

### Monitores

- Los recursos se comparten mediante memoria compartida
- La exclusión mutua es implícita, dado que hay un sólo proceso dentro del monitor en un momento dado
- La sincronización se hace con variables condición (wait/signal)
- El cliente invoca procedimientos del monitor

### Pasaje de mensajes

- Los recursos se comparten mediante el intercambio de mensajes
- La exclusión mutua es explícita dado que el receptor controla el acceso al recurso
- La sincronización se hace con send/receive
- El cliente envía un mensaje al servidor

### Ejemplo Cliente/Servidor

- En monitores, se tendría al servidor como un monitor, al cual se le hacen los pedidos y éste los retorna
- En pasaje de mensajes, el servidor sería un proceso, el cual recibe mensajes de pedidos y retorna los resultados

## 5)

- 5- ¿En qué consiste la comunicación guardada (introducida por CSP) y cuál es su utilidad? Describa cómo es la ejecución de sentencias de alternativa e iteración que contienen comunicaciones guardadas.

La comunicación guardada es un tipo de comunicación en donde la ejecución de una operación de comunicación se encuentra "protegida" por una condición (guarda)

Su utilidad recae en que permite coordinar interacciones complejas entre procesos sin tener busy waiting. Permite que un proceso espere de forma selectiva, distintos mensajes, eligiendo el que corresponda

En una sentencia de alternativa

```
if B1; comunicacion1 -> S1;  
[] B2; Comunicacion2 -> S2;  
fi
```

- Se evaluan todas las guardas
  - Si todas fallan, se termina el if
  - Si al menos una guarda tiene éxito, se elige una de ellas no determinísticamente
  - Si algunas guardas se bloquean (es decir, la guarda es verdadera pero la comunicación no puede ejecutarse inmediatamente), se espera hasta que alguna de ellas tengan éxito
- Luego de elegir la guarda exitosa, se ejecuta la sentencia de comunicacion
- Después se ejecuta la sentencia  $S_i$

En una sentencia de iteracion

- Igual que el if, pero sigue iterando hasta que todas las guardas fallen
- Si no tiene guarda y sólo tiene sentencia de comunicación, se asume que la guarda es verdadera

## 6)

- 6- Modifique la solución con mensajes sincrónicos de la Criba de Eratóstenes para encontrar los números primos detallada en teoría de modo que los procesos no terminen en deadlock.

No porfi

## 7)

- 7- Suponga que N procesos poseen inicialmente cada uno un valor. Se debe calcular la suma de todos los valores y al finalizar la computación todos deben conocer dicha suma.
- a) Analice (desde el punto de vista del número de mensajes y la performance global) las soluciones posibles con memoria distribuida para arquitecturas en estrella (centralizada), anillo circular, totalmente conectada, árbol y grilla bidimensional.
  - b) Escriba las soluciones para las arquitecturas mencionadas.

### a)

Centralizada

- Los mensajes al coordinador se envían casi al mismo tiempo
- El coordinador se ve saturado de mensajes donde tiene que hacer el receive para obtener los valores
- Se tarda más
- El número de mensajes es líneal

Anillo

- Solución lenta, cada proceso debe esperar recibir el mensaje del anterior, secuncializando el problema
- El número de mensajes es lineal

- Es la más fácil, pero tiene mucha comunicación, disminuyendo la velocidad

## Arbol

- Los procesos hoja envían sus valores a sus padres, quienes los suman y los envían a su vez a sus propios padres y así hasta la raiz
- El tiempo de ejecución es proporcional a la altura del arbol
- Es eficiente

## Grilla bidimensional

- La suma se calcula en dos etapas, primero se realiza la suma de los elementos de cada fila; luego, los resultados de cada fila se suman en una columna
- Finalmente, el resultado total se propaga a todos los procesos
- El tiempo de ejecución está dado por la dimensión de la grilla

**b)**

No porfi

**8)**

- 8- Marque similitudes y diferencias entre los mecanismos RPC y Rendezvous. Ejemplifique para la resolución de un problema a su elección.

## Similitudes

- Ambos son sincrónicos, la comunicación ocurre cuando ambas partes están listas
- Ambos simulan llamadas a procedimientos remotos
- Ambos tienen interfaces definidas, con un conjunto de operaciones exportadas
- Ambos trabajan sobre canales tipo link bidireccionales

## Diferencias

- Difieren en la manera de servir la invocación a operaciones
- RPC tiene módulos que exportan procedimientos y procesos que los llaman. Cada vez que se va a atender un llamado, se genera un nuevo proceso que resuelve la invocación al procedimiento, devuelve los resultados y luego finaliza
- Rendezvous tiene procesos que trabajan continuamente e invocan operaciones que resuelven otros procesos. Estos, en algún momento, resuelven el pedido y devuelven los resultados, continuando su ejecución
- RPC sólo brinda comunicación, debe implementarse la sincronización ya que comparten memoria dentro de un mismo módulo
- Rendezvous combina comunicación y sincronización. Los módulos son procesos

Ejemplifique? No porfi

## 9)

- 9- Describa sintéticamente las características de sincronización y comunicación de ADA. Indicar que diferencia hay entre la comunicación guardada de Rendezvous (general) con la provista por ADA.
- ADA se basa en procesos activos llamados Task
  - La comunicación y sincronización entre tareas se realiza mediante Rendezvous
  - Una tarea exporta uno o más Entries
  - Otra tarea realiza un entry call
  - La comunicación es sincrónica
  - La exclusión mutua está implícita dentro del accept, mientras se atiende una llamada, el cuerpo del accept se ejecuta de manera exclusiva

### Diferencias

- Rendezvous
  - Permite comunicación guardada, cada operación puede tener una guarda que controla si la comunicación puede realizarse
  - La elección entre comunicaciones es no determinística, basada en las guardas habilitadas
  - El proceso servidor se describe como un proceso que ejecuta sentencias guardadas
  - Comunicación y sincronización son expresadas a nivel del mecanismo de pasaje de mensajes
- ADA
  - Las guardas no se colocan directamente en cada entrada, la selección se realiza mediante un "select" con cláusulas "when"
  - La comunicación es determinada por el select
  - El servidor es una tarea con entries, la sincronización se realiza mediante el accept
  - Comunicación y sincronización están integradas como primitivas del lenguaje

En CSP el Rendezvous permite guardas directamente asociadas a las operaciones de comunicación, permitiendo mayor expresividad y no determinismo, mientras que en ADA el Rendezvous es una versión **estructurada**, donde las guardas sólo pueden expresarse en el "select ... when", haciendo el mecanismo más **restringido, seguro y legible**.

## 10)

- 10- Considere el problema de lectores/escritores. Desarrolle un proceso servidor para implementar el acceso a la base de datos, y muestre las interfaces de los lectores y escritores con el servidor. Los procesos deben interactuar: a) con mensajes asincrónicos; b) con mensajes sincrónicos; c) con RPC; d) con Rendezvous.

### a)

No porfi

### b)

No porfi

**c)**

No porfi

**d)**

No porfi