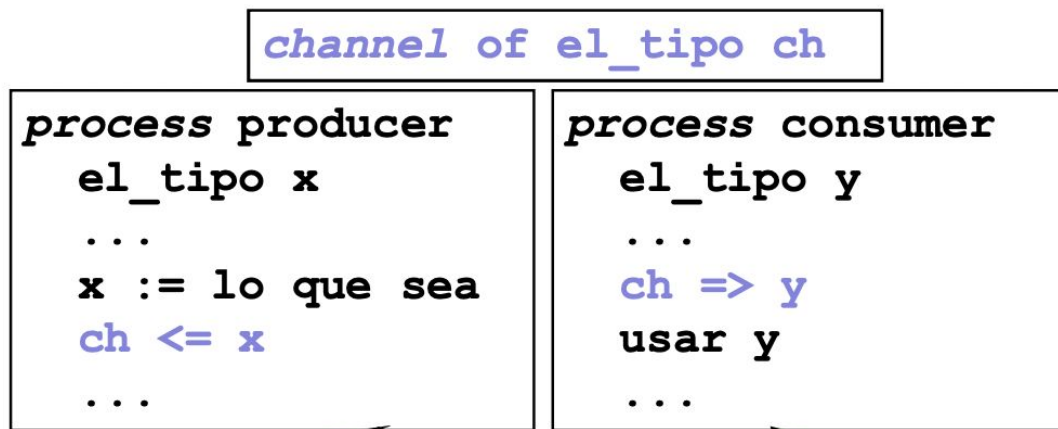


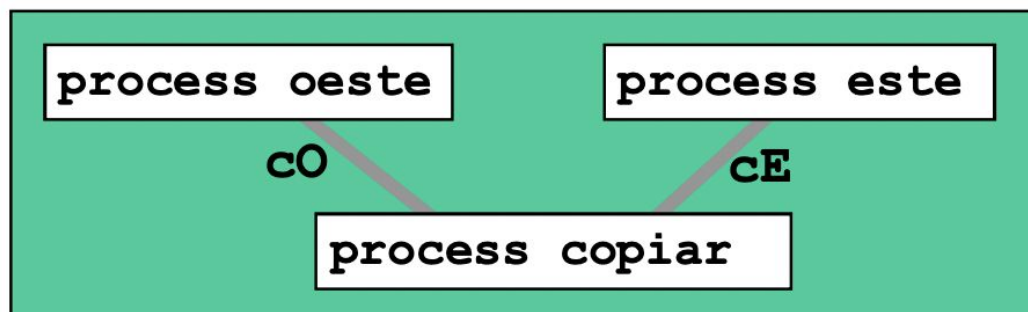
PROGRAMACIÓN DISTRIBUIDA. PROGRAMACIÓN MEDIANTE PASO SÍNCRONO DE MENSAJES.

CURSO 2021-2022

INTRODUCCIÓN

- Programación concurrente: varios procesos trabajan juntos en un mismo ordenador.
- Programación distribuida: los procesos se encuentran en distintos ordenadores. La comunicación entre estos procesos se va a llevar a cabo a **través de puertos**



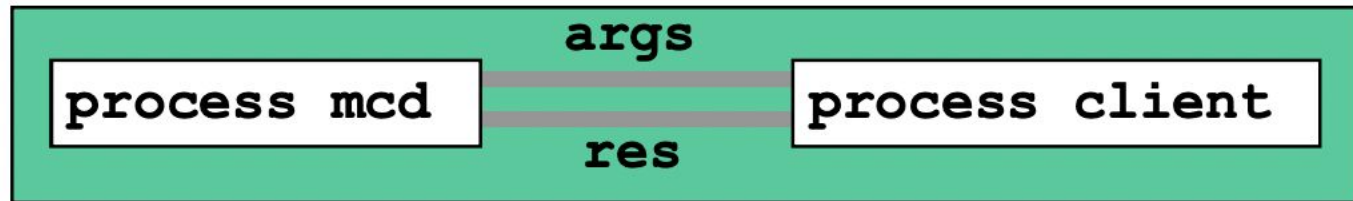


channel of character cE, cO

```
process oeste
character c
loop
   $cO \Rightarrow c$ 
  ...
```

```
process copiar
character c
loop
   $cE \Rightarrow c$ 
   $cO \Leftarrow c$ 
```

```
process este
character c
loop
  produce nuevo c
   $cE \Leftarrow c$ 
```



channel of (integer,integer) args
channel of integer res

```
process mcd
  integer x,y,m
  integer r := 1
  loop
    args => (x,y)
    ... --calcula mcd en m
    res <= m
```

```
process client
  integer a,b,m
  loop
    obtener a,b
    args <= a,b
    res => m
```



INTRODUCCIÓN

- ¿Qué pasa cuando hay **más de un cliente**?: ¡escuchar de forma selectiva!

```
channel of (integer, integer) args, args_2
channel of integer res, res_2
```

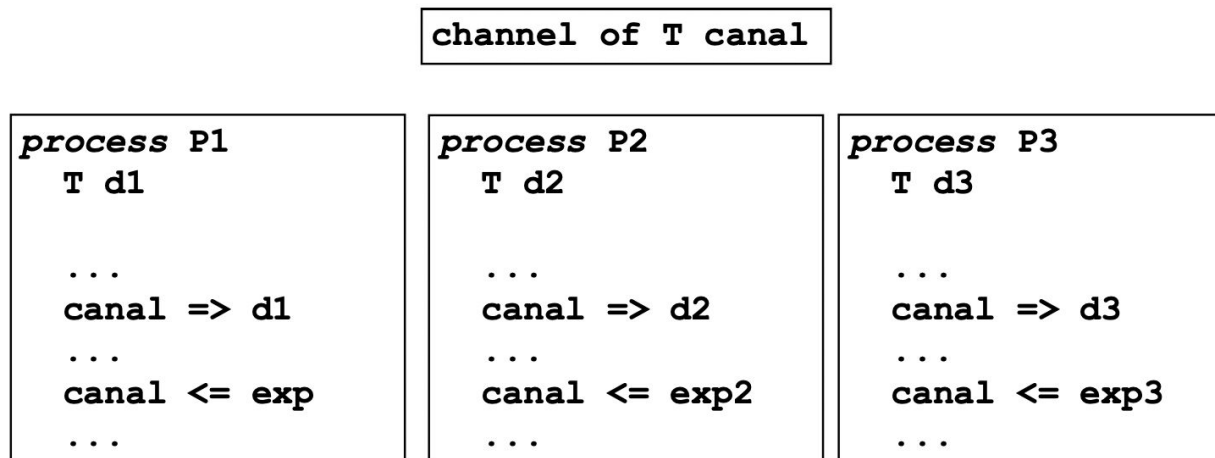
```
process mcd
  integer x, y, m, cl
  integer r := 1
  loop
    either args => (x, y)
      ... --calcula mcd en m
      res <= m
    or
      args_2 => (x, y)
      ... --calcula mcd en m
      res_2 <= m
```

```
process client
  integer a, b, m
  loop
    obtener a, b
    args <= a, b
    res => m
```

```
process client_2
  integer a, b, m
  loop
    obtener a, b
    args_2 <= a, b
    res_2 => m
```

RPC

- **Remote Procedure Call:** un cliente invoca servicios de un servidor que puede estar en otro procesador. La comunicación permite a un proceso influir en la ejecución de otro.
- **RMI** para Java.



SOCKETS EN JAVA

- Los **sockets** son un mecanismo que nos permite establecer un enlace entre dos programas que se ejecutan independientes el uno del otro (generalmente un programa cliente y un programa servidor).
- La librería **java.net** nos provee dos clases: **Socket** para implementar la conexión desde el lado del cliente y **ServerSocket** que nos permitirá manipular la conexión desde el lado del servidor.

SOCKETS EN JAVA

- El servidor estará a la espera de una conexión, en cuanto el cliente inicie enviará un mensaje de petición al servidor, éste le responderá, y una vez recibida la confirmación, el cliente enviará un par de mensajes y la conexión finalizará.

