

# Comunicació de Processos

## Unitat 2.2 - Nel Banqué Torné

---

### 1. Introducció a la Comunicació entre Processos

La cooperació i comunicació entre processos és essencial per a la realització de tasques complexes en sistemes operatius. En entorns **multiprogramats** o **distribuïts**, diversos processos col·laboren per compartir recursos i intercanviar informació, permetent una execució eficient i coordinada.

#### 1.1 Exemple de pipeline

En un entorn de **pipeline**, la sortida d'un procés es converteix en l'entrada d'un altre procés, facilitant així la **comunicació** i el **processament de dades** en diferents etapes del flux de treball.

---

### 2. Exemples de Comunicació i Sincronització entre Processos

#### 2.1 Comunicació

El principal objectiu de la **comunicació entre processos** és l'intercanvi d'informació. Un exemple típic seria utilitzar la comanda `cat missatge`, on un procés llegeix dades des d'un fitxer i un altre procés les processa.

#### 2.2 Sincronització

La **sincronització de processos** implica la coordinació de l'accés als **recursos compartits** i l'ordenació de les tasques, de manera que no hi hagi interferència entre processos. Per exemple, en un sistema d'impressió, el **procés B** revisa periòdicament la cua d'impressió i imprimeix els fitxers, tot assegurant que les variables d'entrada i sortida siguin consistents.

---

### 3. Evitar les Condicions de Carrera

Una **condició de carrera** es produeix quan dos o més processos accedeixen simultàniament a un recurs compartit sense la deguda sincronització, cosa que pot conduir a resultats incorrectes o inconsistents. Les condicions per evitar-les són:

1. **Exclusió mútua:** Dos processos no poden estar simultàniament a la **secció crítica**.
  2. **No assumptes de velocitat:** No s'ha de suposar que els processos són ràpids o lents.
  3. **Cap bloqueig innecessari:** Un procés que no està a la secció crítica no pot bloquejar altres processos.
  4. **No esperes indefinides:** Cap procés ha d'esperar indefinidament per accedir a la secció crítica.
- 

### 4. Tipus de Mecanismes de Comunicació entre Processos

#### 4.1 Fitxers

L'ús de **fitxers** per a la comunicació entre processos implica operacions d'entrada i sortida a disc, les quals són més lentes que altres mètodes de comunicació basats en memòria.

#### 4.2 Pipes

Els **pipes** són canals de comunicació entre processos, especialment en la mateixa màquina. Funcionen com un flux **unidireccional**, i un procés escriu mentre l'altre llegeix. Les accions més comunes amb pipes són:

- **Lectura** amb `read()`.
- **Escriptura** amb `write()`.
- **Tancament** amb `close()`.
- **Eliminació** amb `unlink()`.

##### 4.2.1 Pipes amb nom (FIFOs)

Els **FIFOs** funcionen de manera similar als pipes, però inclouen un mecanisme de comunicació amb nom, permetent que múltiples processos accedeixin al mateix canal.

---

## 5. Sincronització de Dades Compartides

Quan diversos processos comparteixen dades, és essencial garantir la **consistència** d'aquestes dades. Això s'aconsegueix mitjançant:

- **Semàfors**: Mecanisme de control que limita l'accés a un recurs a un nombre determinat de processos.
- **Mutex**: Bloqueig mutu que assegura que només un procés pugui accedir a la secció crítica a la vegada.
- **Candau**: Similar a un mutex, però amb altres opcions de control més avançades.

---

## 6. Programació Paral·lela amb OpenMP

**OpenMP** és una API que facilita la **programació paral·lela**, permetent marcar parts del codi com a regions paral·leles que s'executaran de manera concurrent en diferents fils d'execució. Aquest enfocament permet aprofitar al màxim les arquitectures multicore i millorar el rendiment.

---

## 7. Conclusió

La **comunicació entre processos** en sistemes operatius es basa en una varietat de mecanismes, com ara pipes, FIFOs, cues de missatges, sockets i memòria compartida. Cada un d'aquests mecanismes té **avantatges** i **desavantatges** específics, que s'han d'utilitzar segons el context i les necessitats de la comunicació. A més, la **sincronització** correcta dels processos és fonamental per evitar condicions de carrera i assegurar que els recursos compartits siguin gestionats de manera coherent.