

Computadors

Paral·lelisme

(2/2)

Grau en Ciència i Enginyeria de Dades

Facultat d'Informàtica de Barcelona (FIB)
Universitat Politècnica de Catalunya (UPC)

Creative Commons License

Aquest document utilitza Creative Commons Attribution 3.0 Unported License



Els detalls d'aquesta licència es poden trobar a <https://creativecommons.org/licenses/by-nc-nd/4.0>

Index

- Introducció
- Models de Programació
 - OpenMP
- **Altres mètodes de paral·lelisme a nivell de thread/procés**

Paral·lelisme: Punt de vista del Hardware

- Nombre d'instruccions executades per cicle

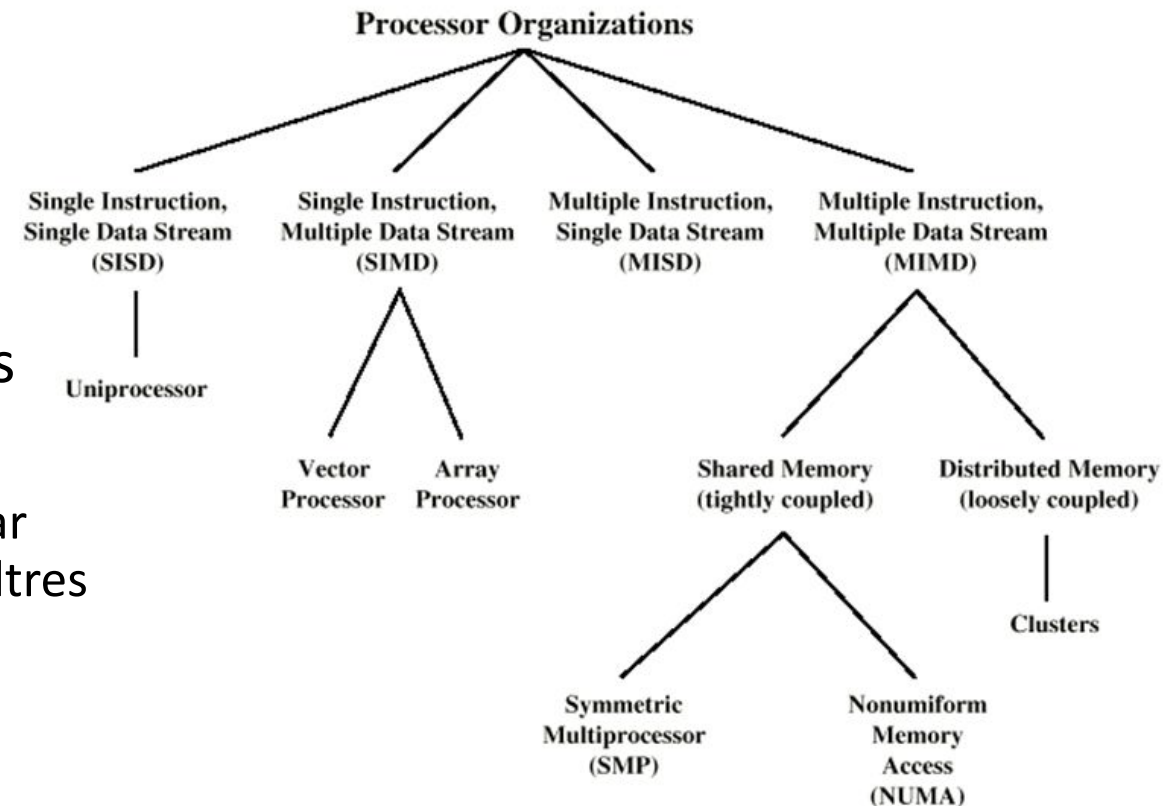
- ILP vs TLP

- Limitacions de recursos

- E.g.: #processadors, #cores, #HwThreads
- Limitacions Estructurals
 - Una instrucció planificada no es pot executar perquè el recurs hw està sent utilitzat per altres

- Taxonomia de Flynn

- https://en.wikipedia.org/wiki/Flynn's_taxonomy



Paral·lelisme: Punt de vista del Software

- L'execució del codi està inherentment limitat per les dependències
 - Dependència de dades
 - Un valor de l'instrucció depèn del resultat d'una instrucció anterior
 - Dependència de control
 - El camí del fluxe d'execució depèn del resultat d'una instrucció anterior
- Suport del compiler / biblioteca
 - Optimitzacions en temps de compilació
 - Model de programació paral·lela

Paral·lelisme: Punt de vista del Software

- Nivells de Paral·lelisme
 - Granularitat gruixuda (milers d'instruccions o programes independents; ~segons)
 - Parts d'un programa, Tasks
 - Baixa càrrega addicional per comunicació i sincronització, però difícil per balancejar la càrrega de treball
 - Granularitat mitjana (<10000 instruccions; ~milisegons)
 - Procediments, Rotines
 - Granularitat fina (10s-1000s instruccions; ~microsegons)
 - Bucles, grups d'instruccions
 - Assistit pel compilador (difícil de detectar pels programadors) i models de programació paral·lela
 - Adequat per memòria compartida i fàcil per balancejar càrrega de treball
- **Decisions del disseny paral·lel poden tenir conseqüències importants!!!!**
 - **Tenir en compte els extra-costos de sincronització i gestió paral·lela (fork-join; cues)**

Paral·lelisme: Punt de vista del SO

- Gestió de processos i threads
 - User-Level Threads vs Kernel-Level Threads
- Polítiques de planificació
 - Basat en afinitat en recursos Hw, en prioritats, etc
- Balanceig de càrrega de treball entre recursos Hw
 - Gestió de contenció de recursos compartits
- Capacitat de comunicació i sincronització
 - **Inter-process communication (IPC)**

Models de Paral·lelisme segons Memòria

- Shared-Memory vs Distributed-Memory Computing

- Shared-memory

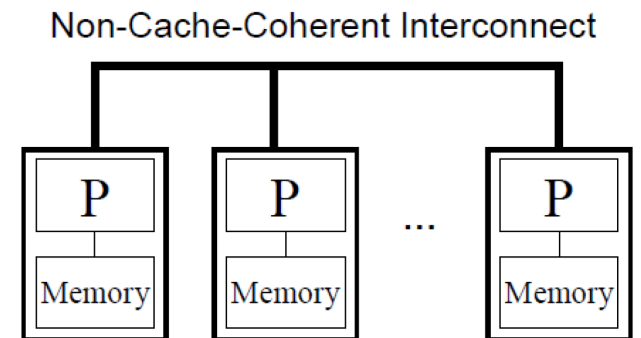
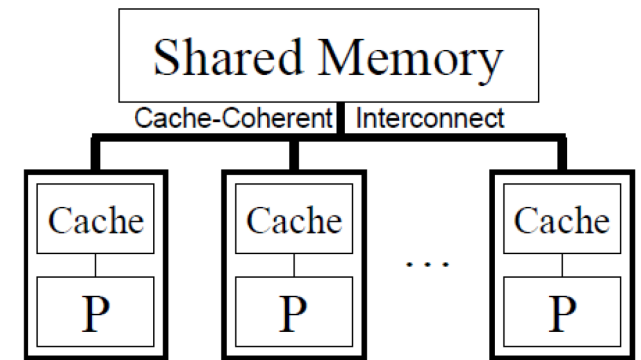
- Adequat per multithreaded
 - E.g.: **OpenMP – Open MultiProcessing**

- Distributed-memory

- Adequat per càrregues de treball computacional
 - E.g.: **MPI – Message Passing Interface**

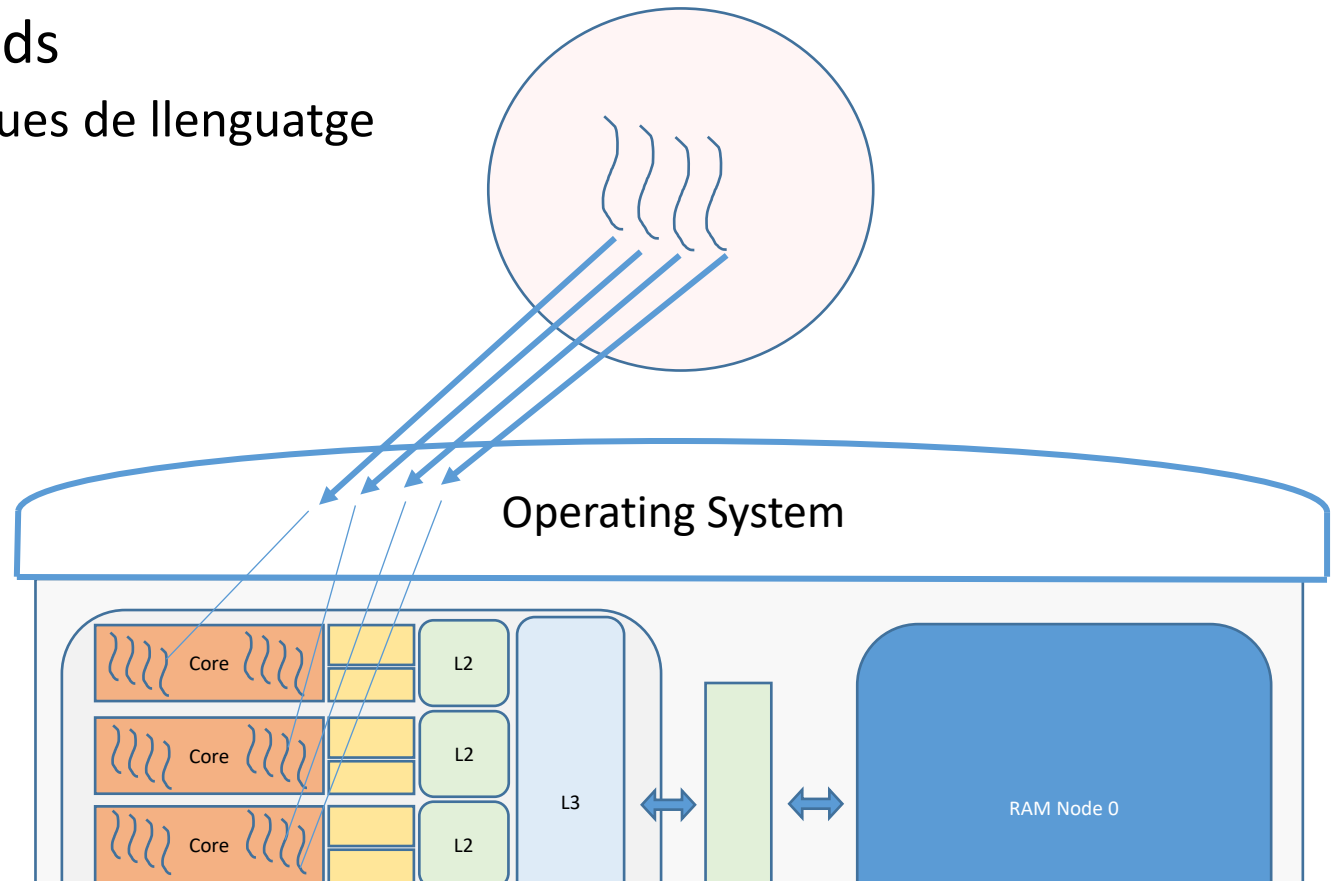
- Tècniques de programació híbrida

- Combinació de les dues



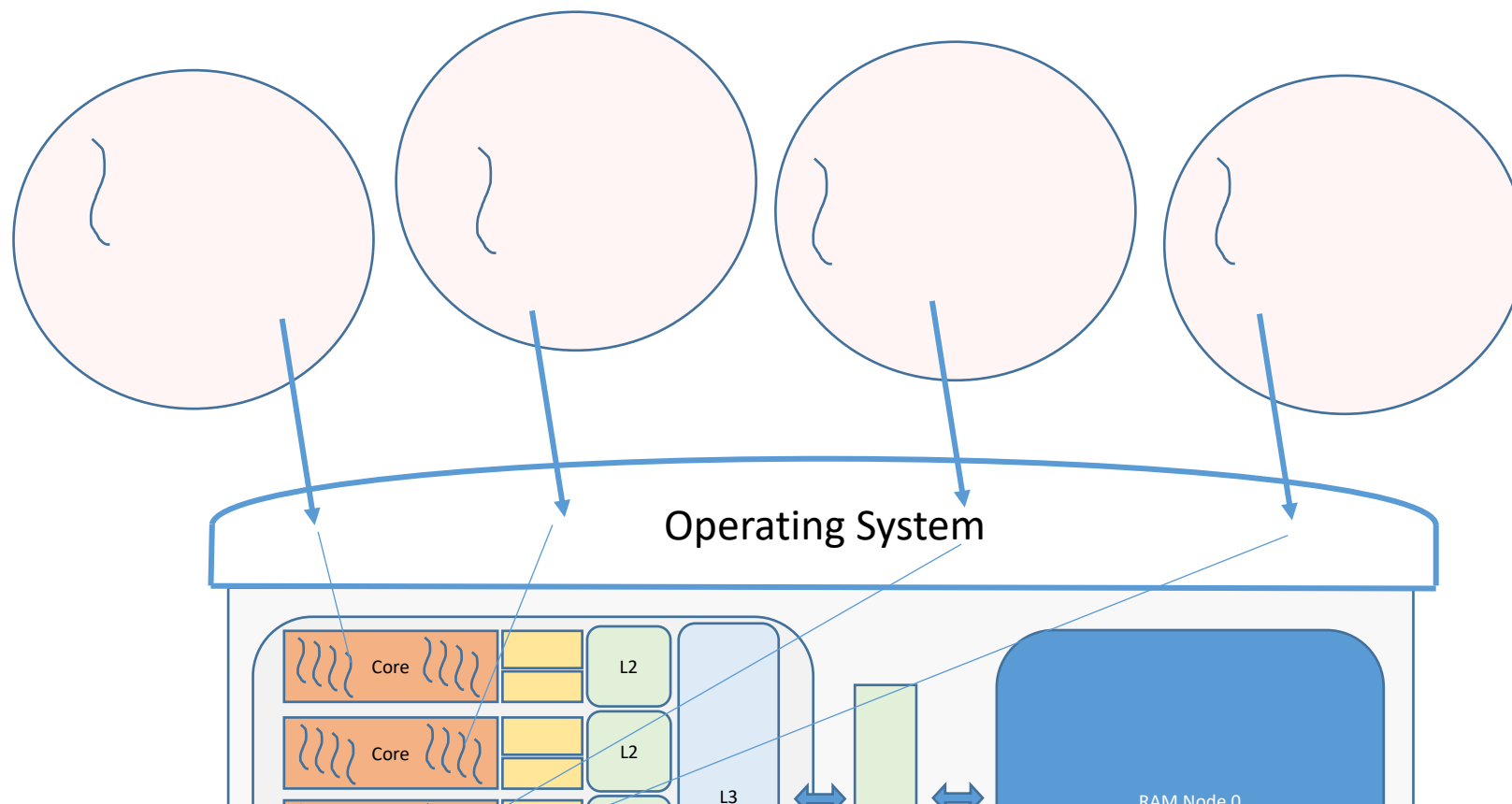
Utilització de recursos Hw (Threads)

- Mètode Multi-threaded
 - Permet explotar el paral·lisme dins de cada procés
 - Suport de gestió de threads
 - Per part del SO i biblioteques de llenguatge



Utilització dels recursos Hw (**Processors**)

- Mètode multi-procés
 - Sense memòria compartida. Sense complexitat en la gestió de memòria



Inter-Process Communication (IPC)

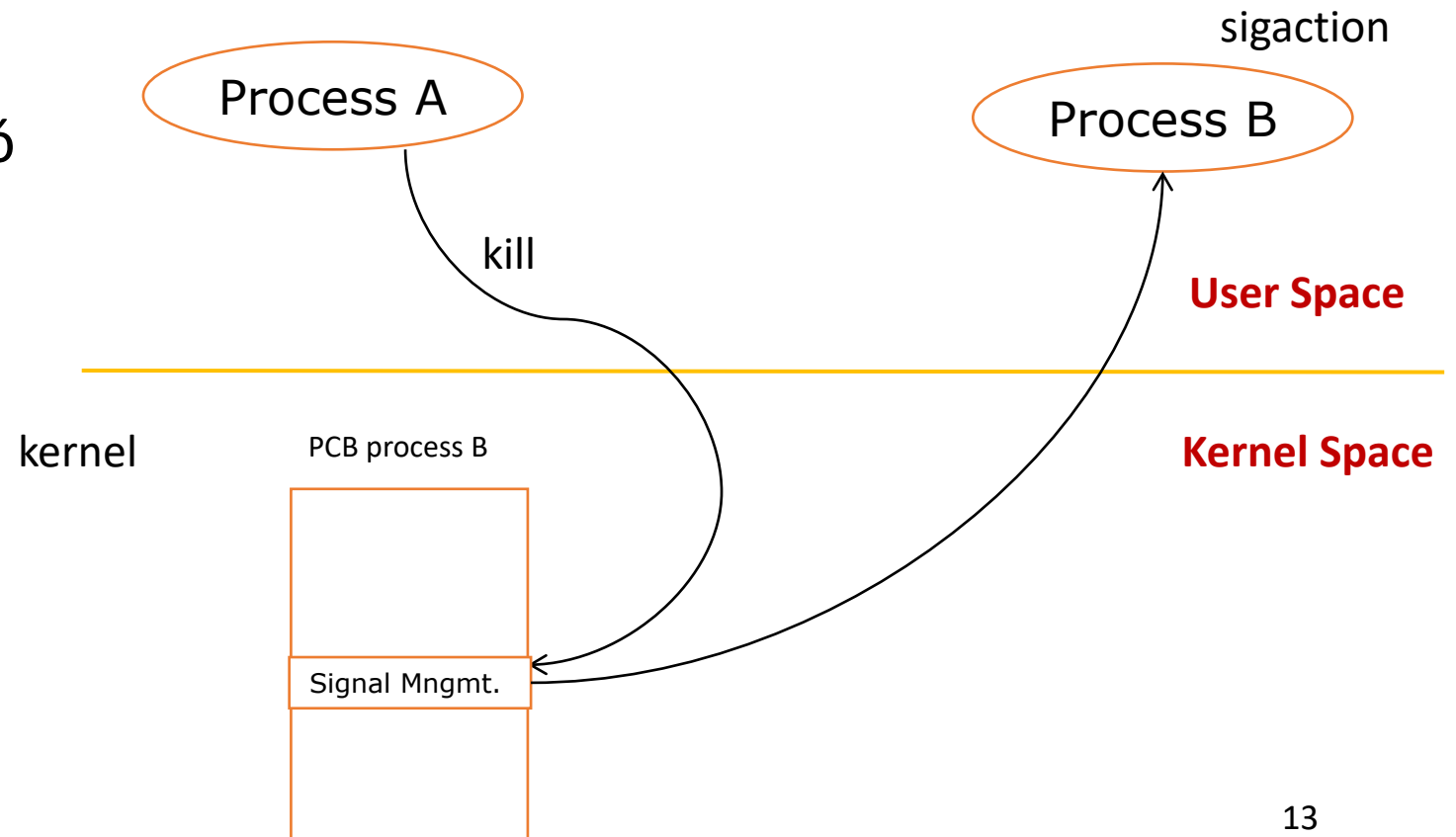
- Processos/threads que cooperen entre si necessiten comunicació i sincronització
 - Intercanvi de dades / Notificació d'esdeveniments
 - Comunicació bloquejant (síncrona) / no-bloquejant (asíncrona)
 - Comunicació directe / indirecte
- Els processos poden executar-se en un únic ordinador o en variïis
- Mètode d'IPC basat en ...
 - Latència, ampla de banda, tipus d'intercanvi de dades
- **Els errors són difícils de trobar i resoldre!!!!**

Inter-Process Communication (IPC)

- Mecanismes de comunicació
 - Signals
 - Pipes/named pipes
 - Shared Memory
 - Memory-Mapped files
 - Sockets
 - Message Passing libraries
 - Others...

Inter-Process Communication (IPC)

- Signals
 - Notificació d'esdeveniment
 - SENSE intercanvi de dades
 - Canvi en el fluxe d'execució
 - Accions programables
- Exemple:
 - tecles Ctrl + C



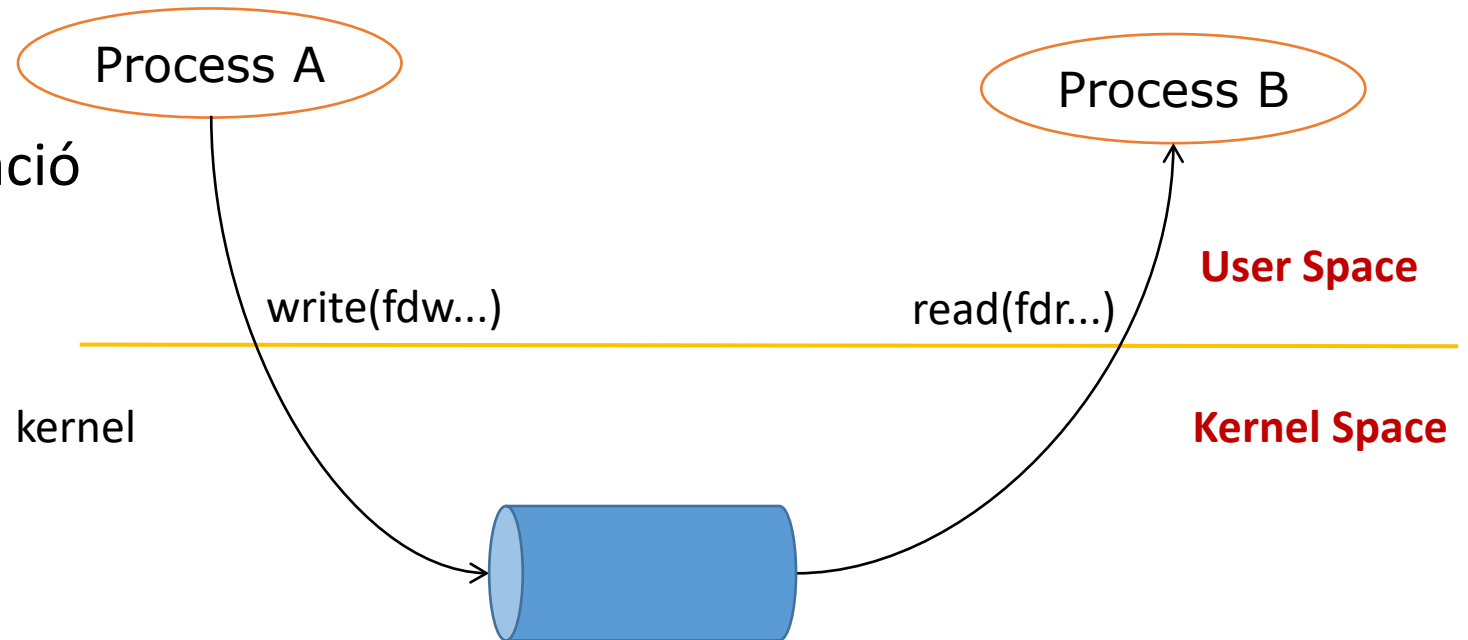
Inter-Process Communication (IPC)

- Pipes/Named Pipes

- Memòria FIFO
- Comunicació N:M
- Mecanismes de sincronització

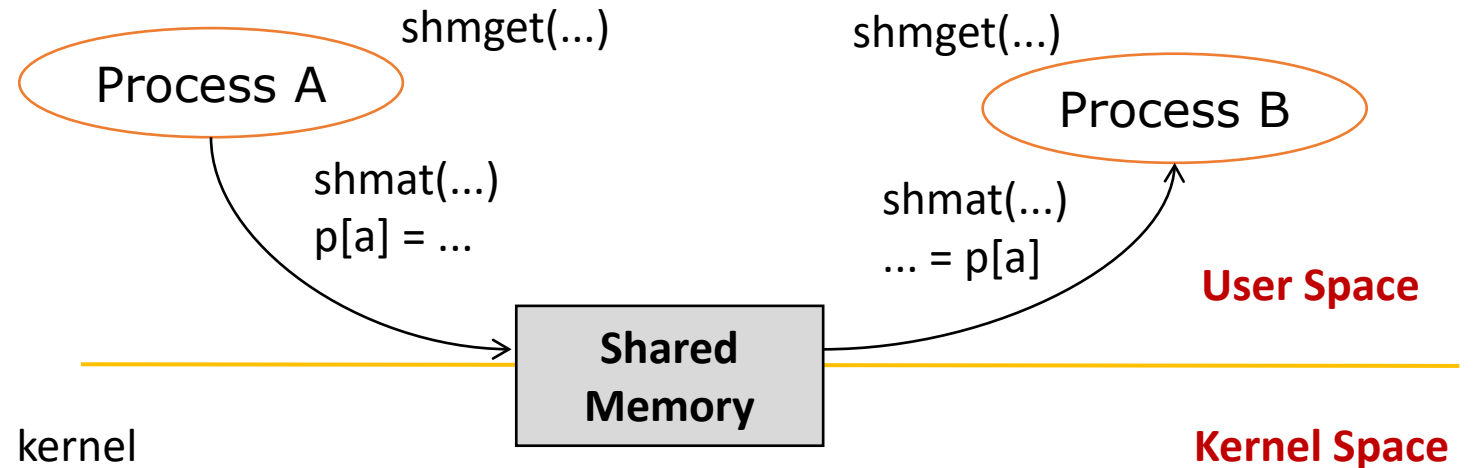
- Exemple:

- Prog1 | Prog2
- Línia de comandes:
 - “#>mknod PIPE p”



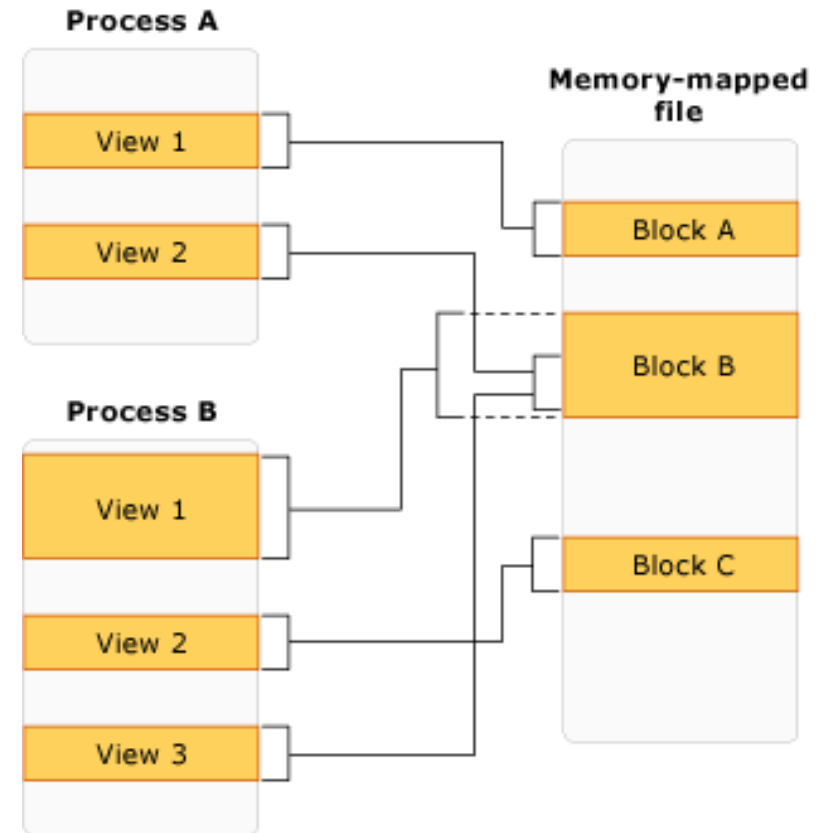
Inter-Process Communication (IPC)

- Memòria compartida
 - Varis processos comparteixen un espai de memòria
 - Molt ràpid
 - Necessita sincronització explícita



Inter-Process Communication (IPC)

- Fitxers mapejats en memòria
 - Un fitxer es carrega en memòria virtual
 - Pot ser compartit entre processos

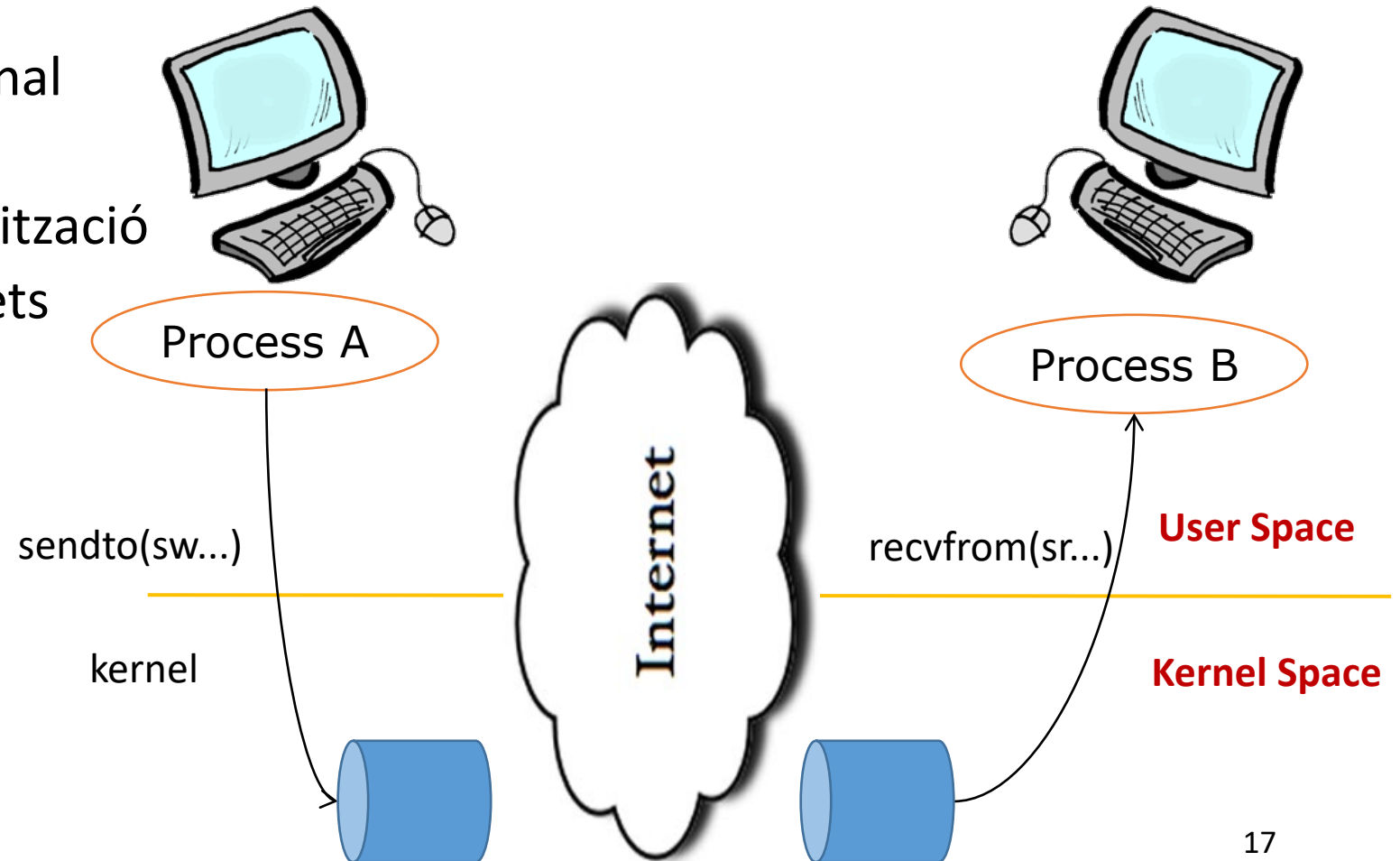


<https://docs.microsoft.com/en-us/dotnet/standard/io/memory-mapped-files>

Inter-Process Communication (IPC)

- Sockets

- Comunicació bidireccional entre processos
- Mecanismes de sincronització
- Diferents tipus de sockets

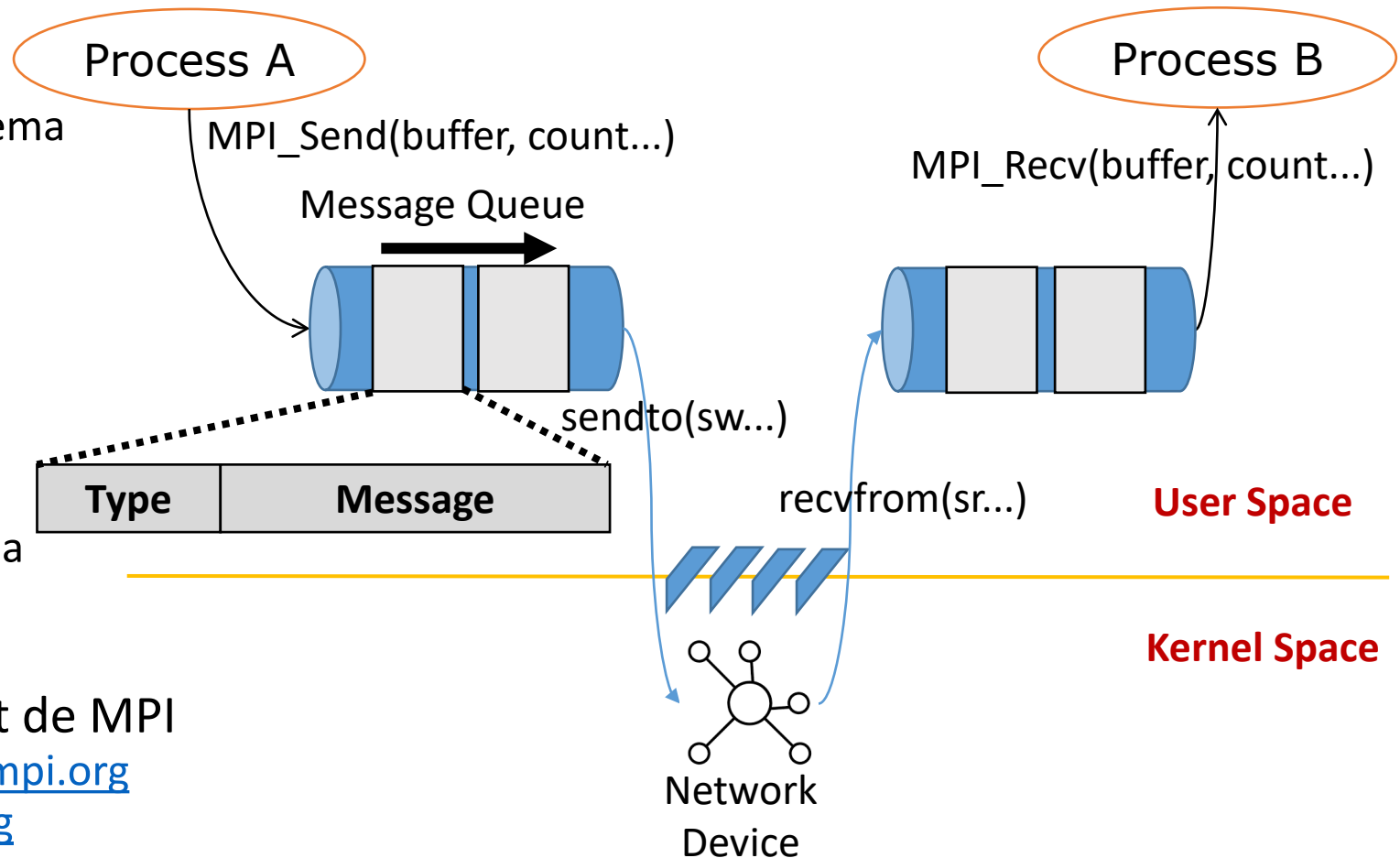


Inter-Process Communication (IPC)

- Message Passing
 - Implementat amb crides a sistema
 - Overhead
 - Cues de missatges
 - Missatges estructurats

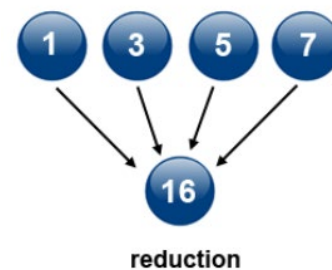
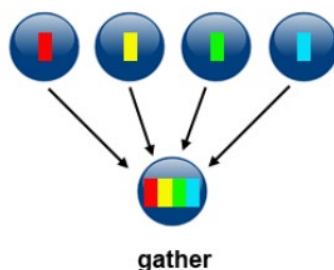
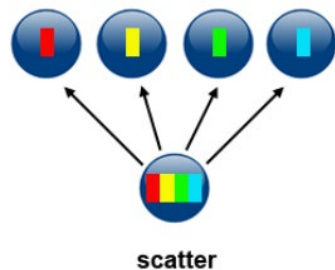
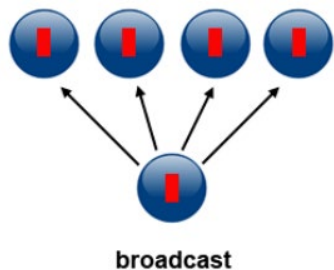
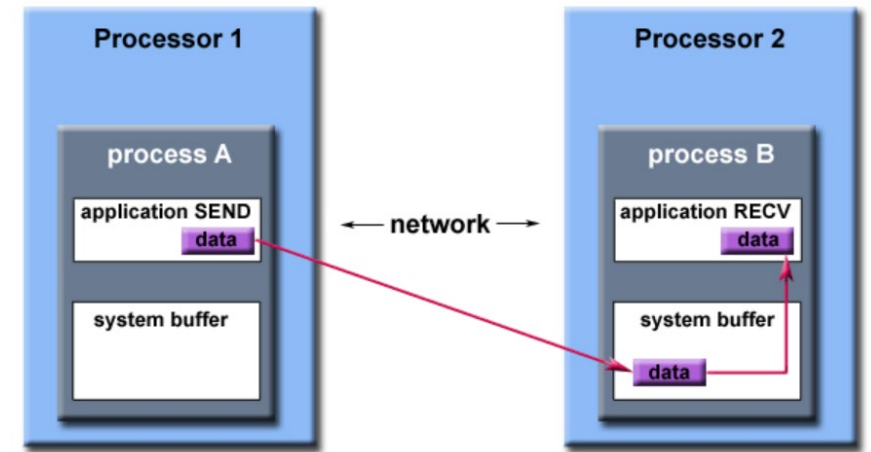
- **MPI**
 - Message Passing Interface
 - Orientat a computació paral·lela

- Implementacions de codi obert de MPI
 - OpenMPI, <https://www.open-mpi.org>
 - MPICH, <https://www.mpich.org>



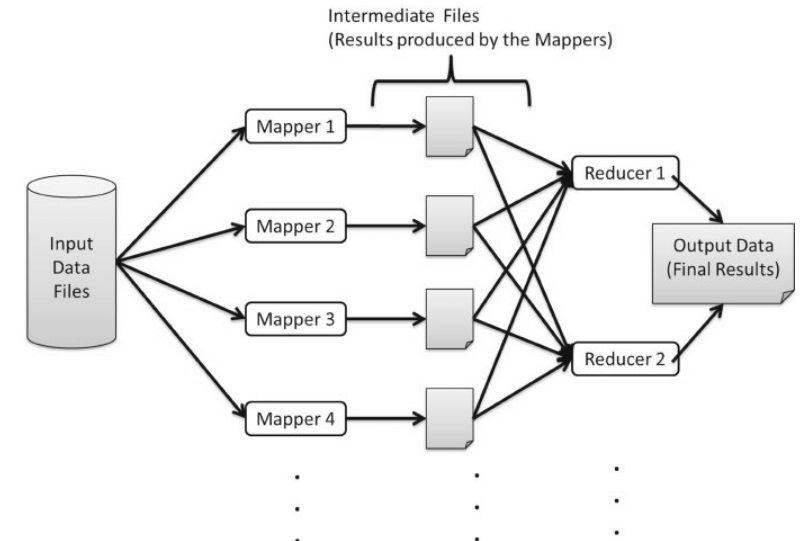
Comunicació basada en missatges

- Idea bàsica: dividir la feina en diferents tasques comunicades entre si
 - Bloquejant vs no-bloquejant
- Diferents tipus de comunicació
 - Comunicació Punt-a-Punt
 - Missatges entre dues tasques MPI
 - Comunicació col·lectiva
 - Tots els processos d'un determinat àmbit



Altres Models de Programació Paral·lela...

- CUDA (per [GPGPU](#))
 - Arquitectura i model de programació paral·lela de propòsit general de NVIDIA
 - La GPU accelera aplicacions a més de gràfics
 - Distribuir el codi en centenars de cores i milers de threads
 - E.g.: 1 Tesla: 240 cores; 128 threads per core → 30720 threads en total
- MapReduce
 - Adequat per processament paral·lel de dades
 - Les dades es divideixen per tot un cluster en un sistema de fitxers distribuït
 - Moure la computació a les dades i realitzar el processament en paral·lel en els diferents nodes



Bibliografia

- Computer Systems – A Programmer's perspective (3rd Edition)
 - Randal E. Bryant, David R. O'Hallaron, Person Education Limited, 2016
 - https://discovery.upc.edu/permalink/34CSUC_UPC/11q3oqt/alma991004062589706711
 - Several Chapters
- OpenMPI
 - <https://www.open-mpi.org/>
- OpenMP
 - <https://www.openmp.org>
 - OpenMP 5 reference guide C/C++ summary
 - <https://www.openmp.org/wp-content/uploads/OpenMPRef-5.0-111802-web.pdf>