

# Bevezetés a párhuzamos programozási koncepciókba

Kacsuk Péter és Dózsa Gábor

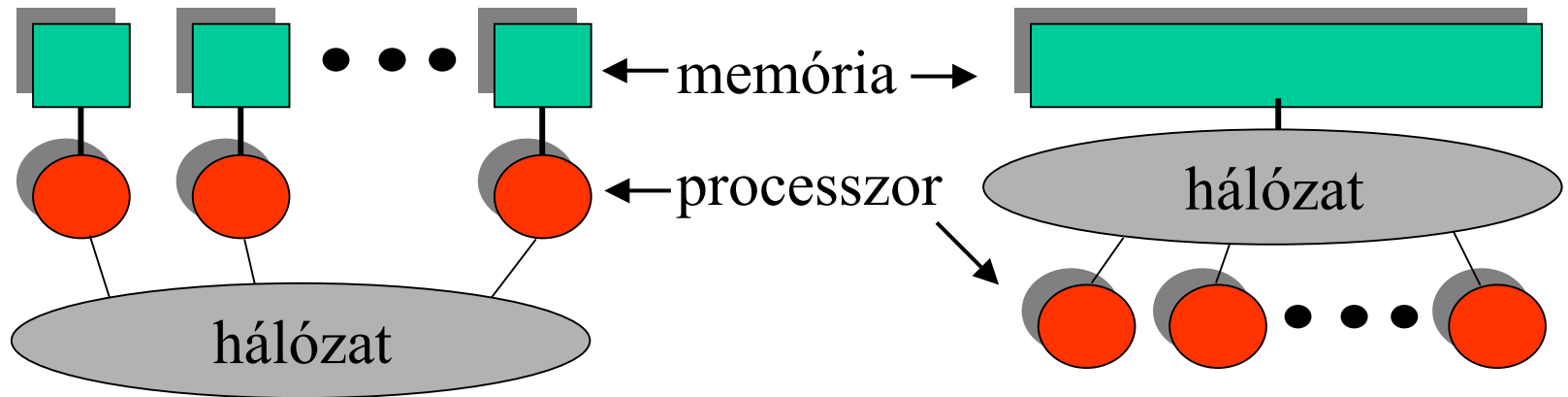
MTA SZTAKI

Párhuzamos és Elosztott Rendszerek  
Laboratórium

E-mail: [kacsuk@sztaki.hu](mailto:kacsuk@sztaki.hu)

Web: [www.lpds.sztaki.hu](http://www.lpds.sztaki.hu)

# Programozási modellek



**Üzenetküldés  
alapú  
kommunikáció**

**MPI, PVM**

**Közös memória  
alapú  
kommunikáció**

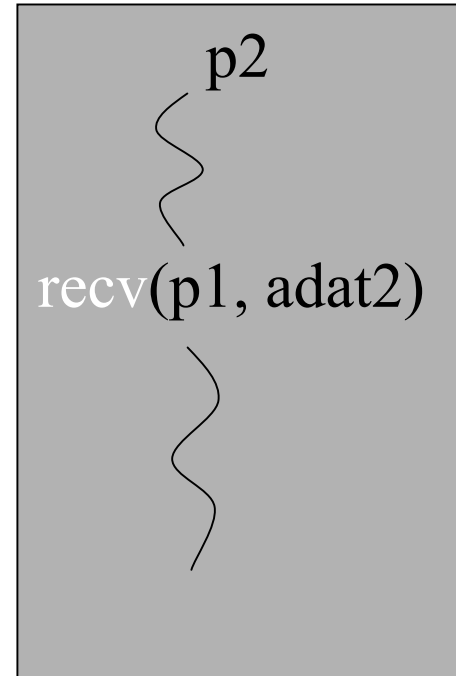
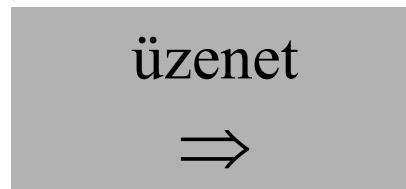
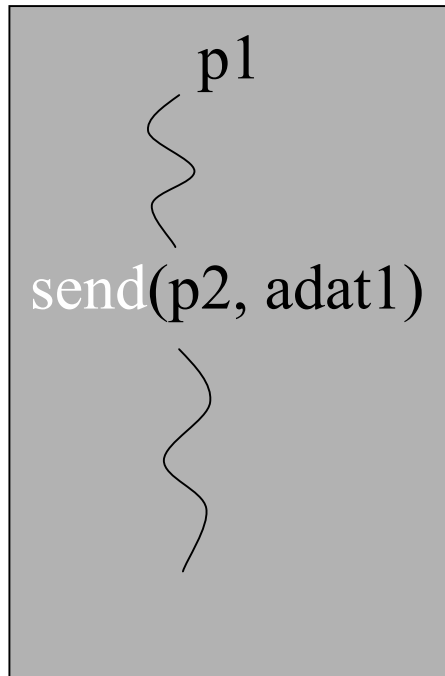
**OpenMP**

# Üzenetküldés alapú programok

Normál szekvenciális programozási nyelv (pl. Fortran, C, ...)

+

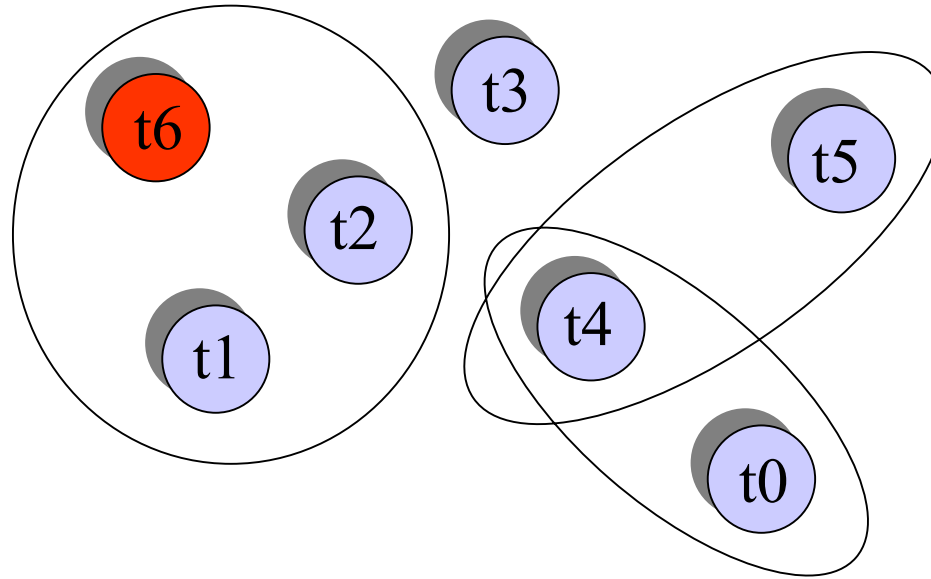
néhány speciális eljárás (rutin könyvtár) az üzenetek kezeléséhez.



# Üzenetküldő rendszerek

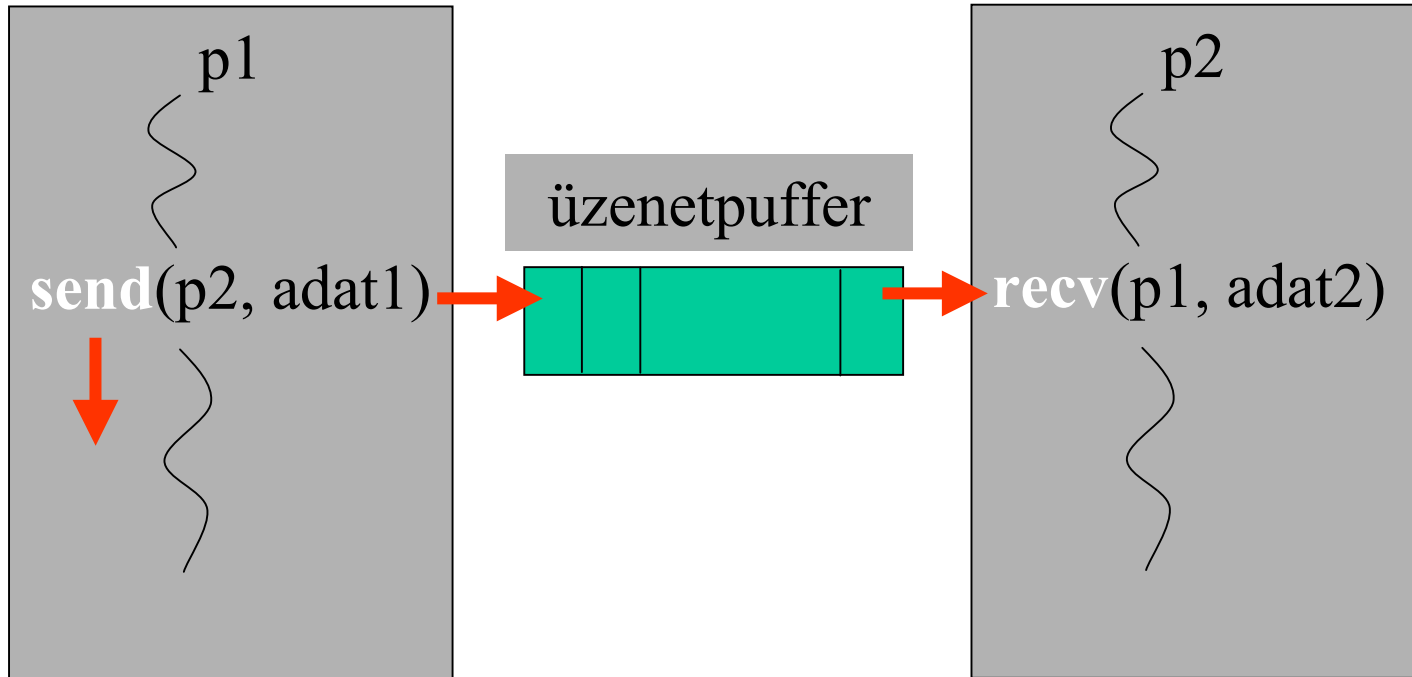
- **PVM : Parallel Virtual Machine**
  - üzenetküldő rendszer + egyszerű elosztott erőforrás kezelés (“virtuális gép” kialakítása)
  - Oak Ridge National Laboratory
- **MPI : Message-Passing Interface**
  - üzeneteküldő interfész szabvány (dokumentum!)
  - széles körű szakmai fórum alkotta meg
  - több implementáció (pl. MPICH, Sun HPC)

# PVM procesz modell

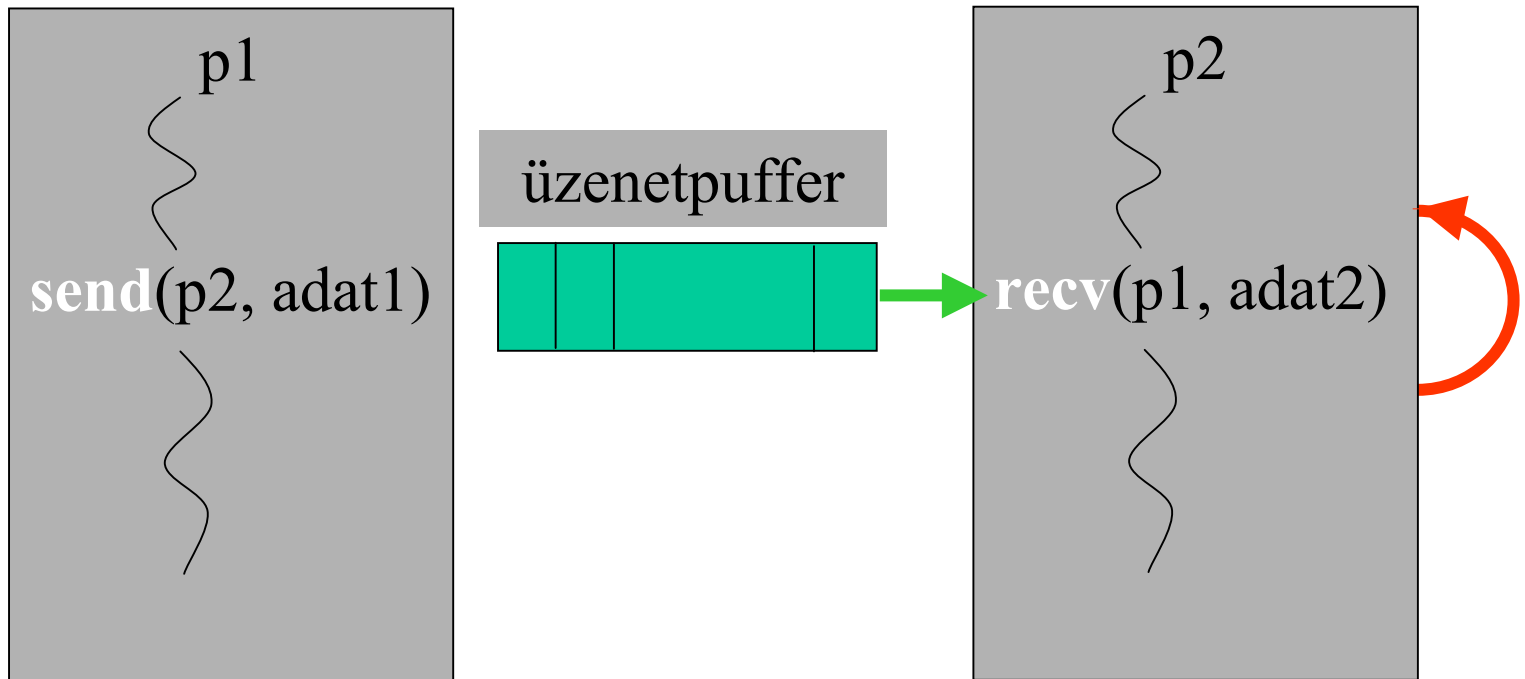


- Procesz : önálló vagy csoport(ok) tagja
- Procesz azonosítása: globálisan (“task id”)
  - **pvm\_mytid()** : “task id” lekérdezése
- Procesz dinamikusan hozható létre és csatlakoztatható csoporthoz

# Aszinkron üzenetküldés

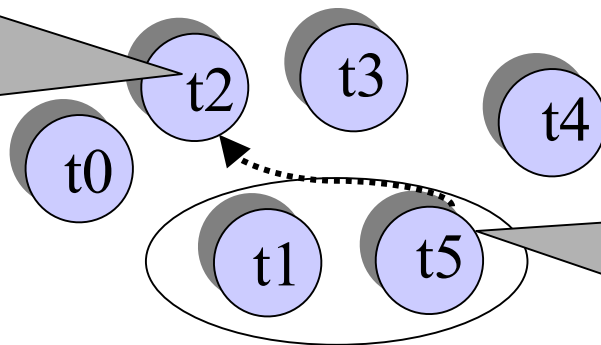


# Blokkolt üzenetvétel



# PVM: Pont-pont kommunikáció

```
pvm_recv(...)  
pvm_upk...(...)  
:  
pvm_upk...(...)
```



```
pvm_initsend(...)  
pvm_pk...(...)  
:  
pvm_pk...(...)  
pvm_send(...)
```

- Címzett azonosítása: **pvm\_send**

– célprocesz globális azonosítója (“task id”),

- Üzenet azonosítása:

– output puffer törlése: **pvm\_initsend**,

– adatok feltöltése a pufferbe: **pvm\_pk...**,

# PVM: Mi van benne ?

- **Kommunikáció és procesz kezelés**
  - pont-pont kommunikáció, kollektív kommunikáció, aszinkron küldés, blokkolt vétel
  - dinamikus procesz indítás / leállítás
- **Dinamikus virtuális gép kezelés**
  - virtuális párhuzamos gép kialakítása a felhasználó számára heterogén hardver környezetben
  - **pvm\_addhost(...), pvm\_delhost(...), ...**

# MPI processz modell



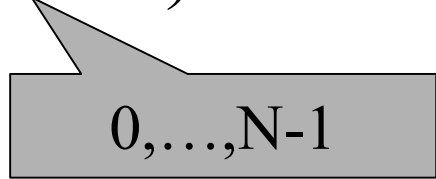
The diagram illustrates the MPI process model. It features a large outer oval labeled 'MPI\_COMM\_WORLD' which contains six smaller, overlapping ovals. Each of these smaller ovals contains a light blue circle with a grey shadow, representing a process. The overlapping nature of the smaller ovals indicates that processes can belong to multiple communication groups.

MPI\_COMM\_WORLD

- Kommunikátor : processz csoport + üzenet kontextus
- Processz azonosítás : (csoport, index)

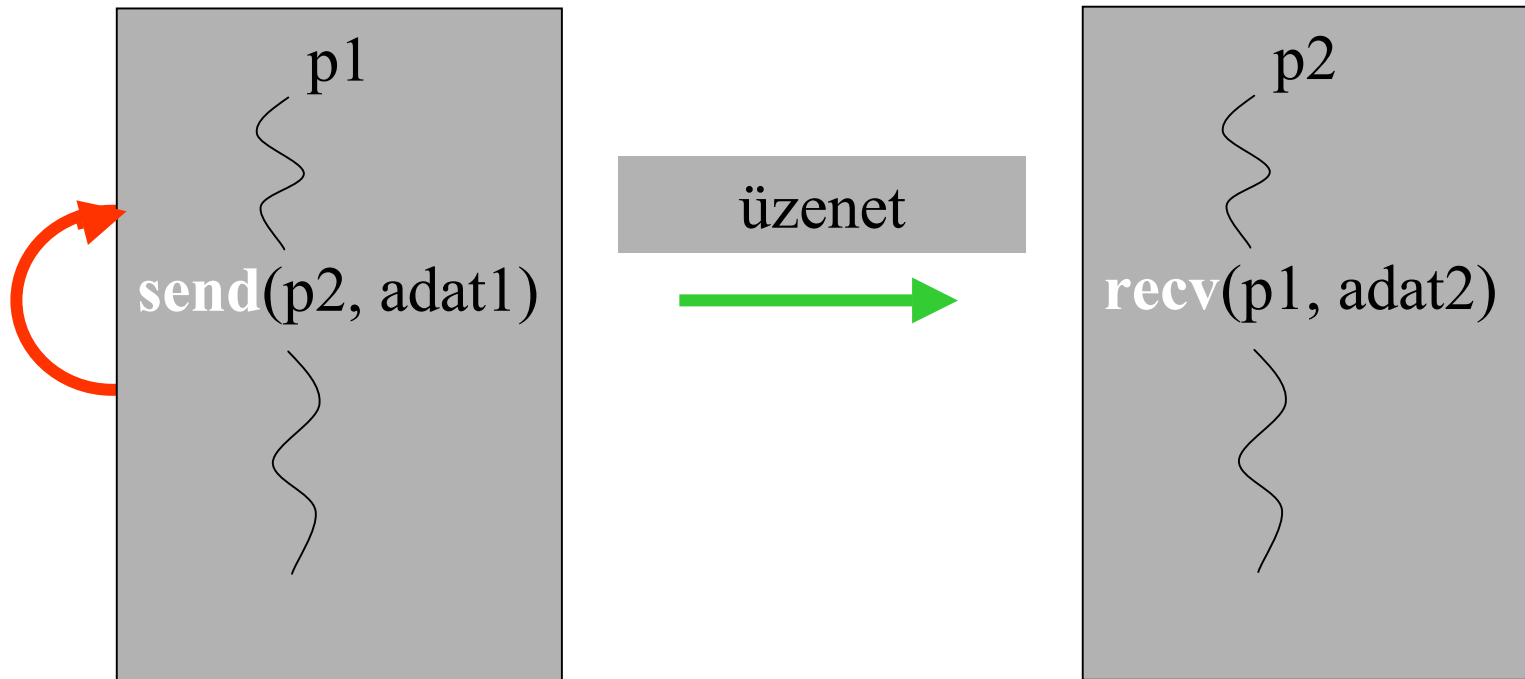


kommunikátor

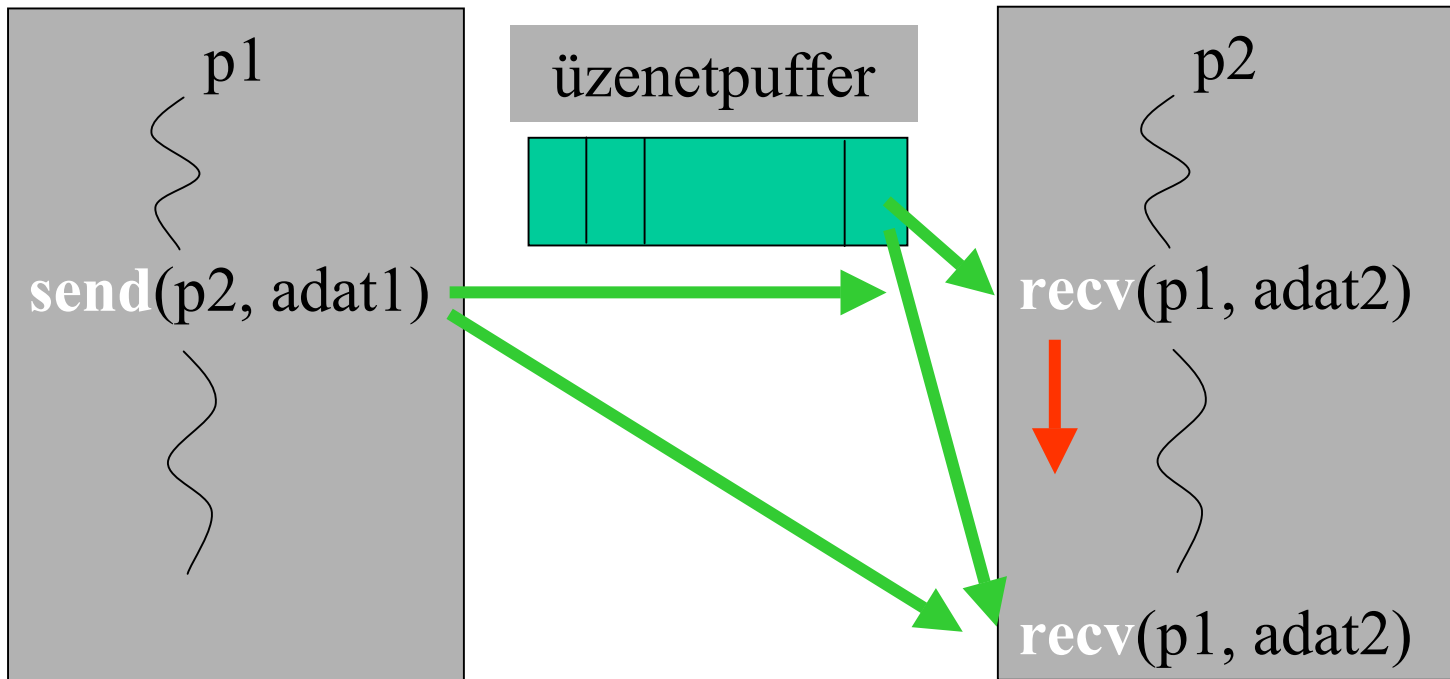


0,...,N-1

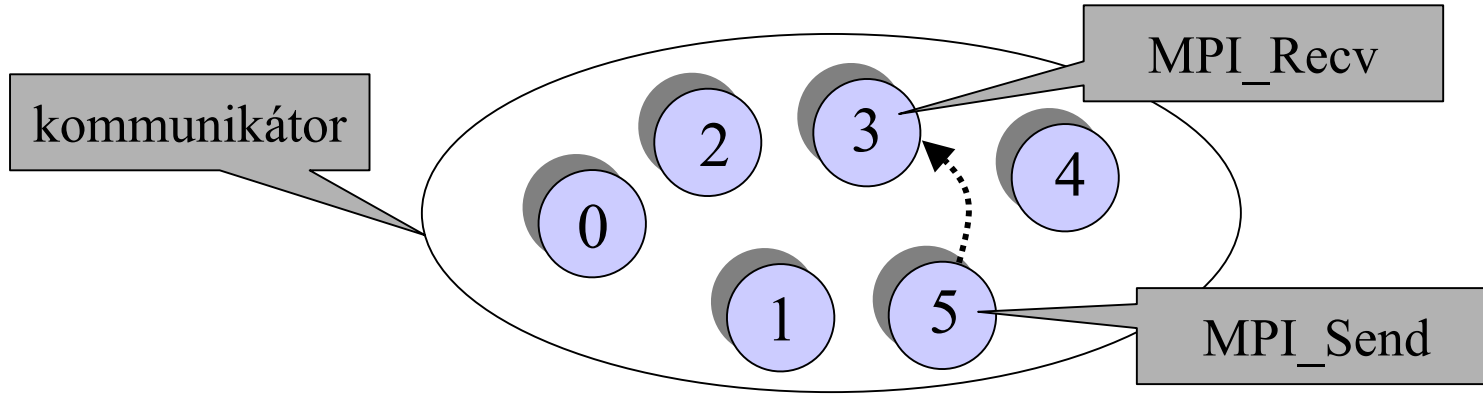
# Szinkron üzenetküldés



# Nem-blokkolt üzenetvétel



# MPI: Pont-pont kommunikáció



- **MPI\_Send** legfontosabb paramétere:

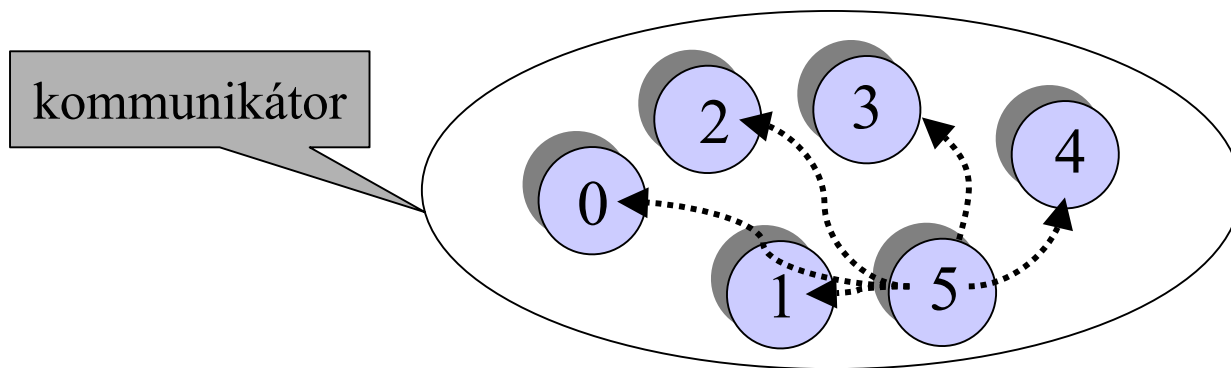
- adat puffer kezdő memória címe,
- adatelemek száma,
- adatelemek típusa (MPI\_Datatype),

üzenet  
azonosítása

- célprocesz indexe,
- kommunikátor (MPI\_Comm)

címzett  
azonosítása

# MPI: Kollektív kommunikáció



- *Minden* proceszben : **MPI\_Bcast**
- **MPI\_Bcast** paraméterei:

- adat puffer kezdő memória címe,
- adatelemek száma,
- adatelemek típusa (MPI\_Datatype),

üzenet  
azonosítása

címzettek  
azonosítása

- kommunikátor (MPI\_Comm)
- a gyökér procesz indexe,

küldő  
azonosítása

# MPI : Mi van benne ?

- MPI-1.1

- pont-pont kommunikáció, kollektív kommunikáció, topológiák (sokdimenziós rács ill. általános gráf), szinkron és aszinkron üzenetküldés, blokkolt és nem-blokkolt üzenetvétel
- Származtatott adattípusok
- ~ 120 rutint definiál

- MPI-2

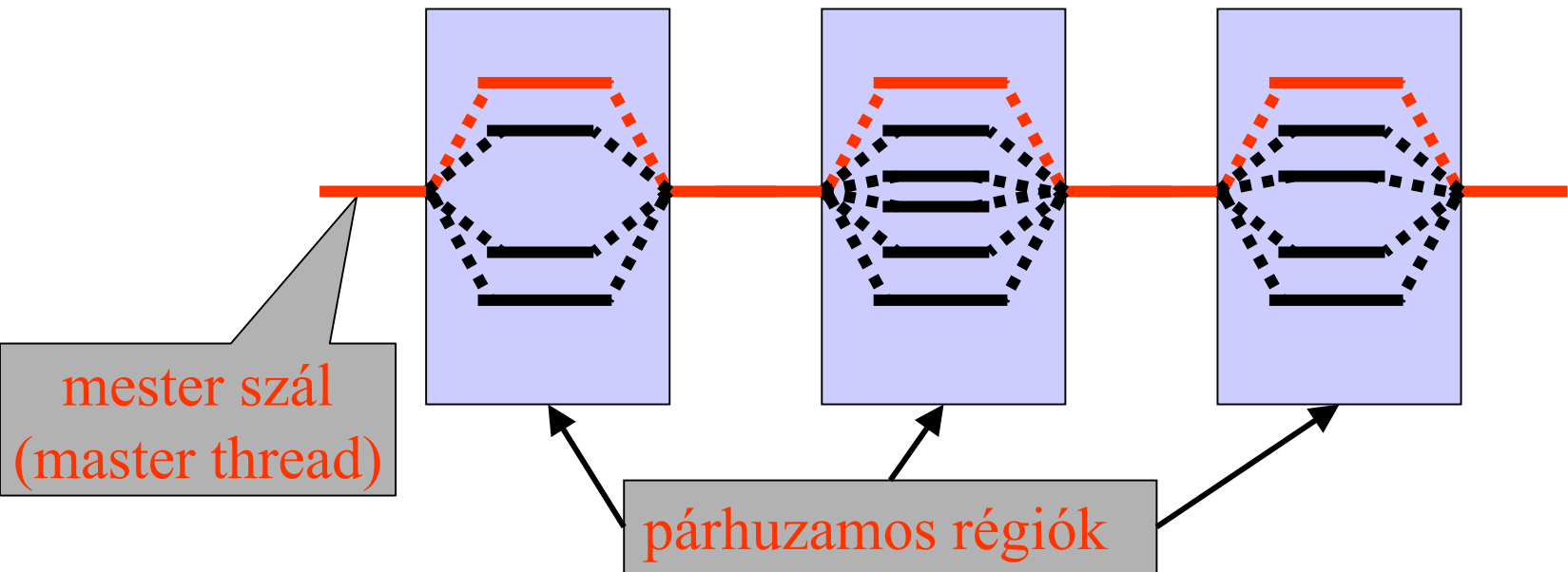
- processz kezelés (létrehozás, kilövés,...), párhuzamos IO
- több mint 100 újabb rutin

# OpenMP: Közös memóriás kommunikáció

- Cél: **szabvány** kidolgozása
- Hardver gyártók és kutatók együttes fejlesztése
- Programozói **interfész** specifikáció **sokszálas programok** készítéséhez
- Fordítási direktívák + rutin könyvtár
- Radikálisan megkönnyíti sokszálas Fortran, C és C++ programok írását

# OpenMP: Programozási modell

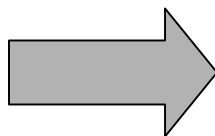
- Szétágazás-összevonás (“fork-join”) párhuzamosság
- Párhuzamosság inkrementális növelése



# OpenMP: Tipikus alkalmazás

- Ciklusok párhuzamosítása

```
void main()
{
    :
    :
    for(i=0; i<1000; i++)
        sok_szamolas(res[i])
    :
}
```



```
void main()
{
    :
    #pragma omp parallel for
    for(i=0; i<1000; i++)
        sok_szamolas(res[i])
    :
}
```

Szekvenciális program

Párhuzamos program

# Összehasonlítás

- **PVM**
  - dinamikus, heterogén környezet hatékony támogatása
  - hibatűrés és flexibilitás
- **MPI**
  - gazdag rutinkészlet komplex programokhoz
  - rendkívül hatékony implementációk
  - statikus és homogén
- **OpenMP**
  - inkrementális párhuzamosítás
  - gyakori skálázhatósági problémák (max. ~10-20 processzor)

*Köszönöm a figyelmüket*



További információ: [www.lpds.sztaki.hu](http://www.lpds.sztaki.hu)