

# *Parallel programming laboratory*

Imre Szeberényi

BME IIT

<szebi@iit.bme.hu>



MŰEGYETEM 1782

# *Burglar – cops*

---

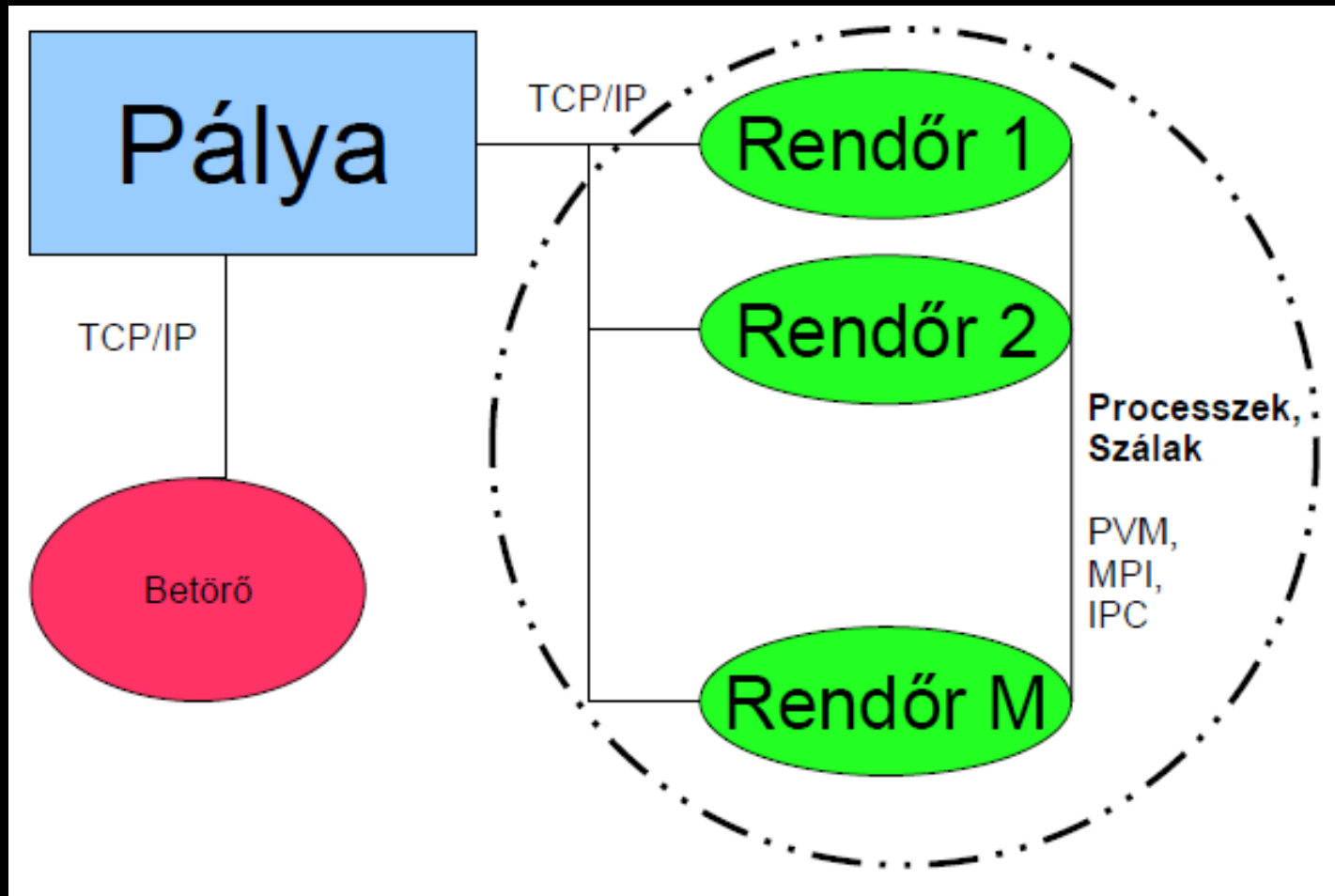
- Cops are chasing a burglar on the street (labyrinths).
- The cops are communicating with each other or with police centre on mobile.
- Every moving actor (cop, robber) is a thread or a process that communicates with each other.
- Chasing is over when the robbers can not move.

# *Labyrinth server*

---

- The server runs on the local Windows machine and is responsible for displaying it.
- It is listening to TCP port 15623. It performs simple commands or. display.
- The clients are running on the server para.
- The connection is made through the TCP tunnel through firewalls.

# Architecture



# Communication

- Simple text messages in both directions ending with endline ('\n').

## Initialization:

Message	Answer	Meaning
M		Burglar step in
	X,Y	Burglar initial position
N		Cop step in
	X,Y	Cop initial position

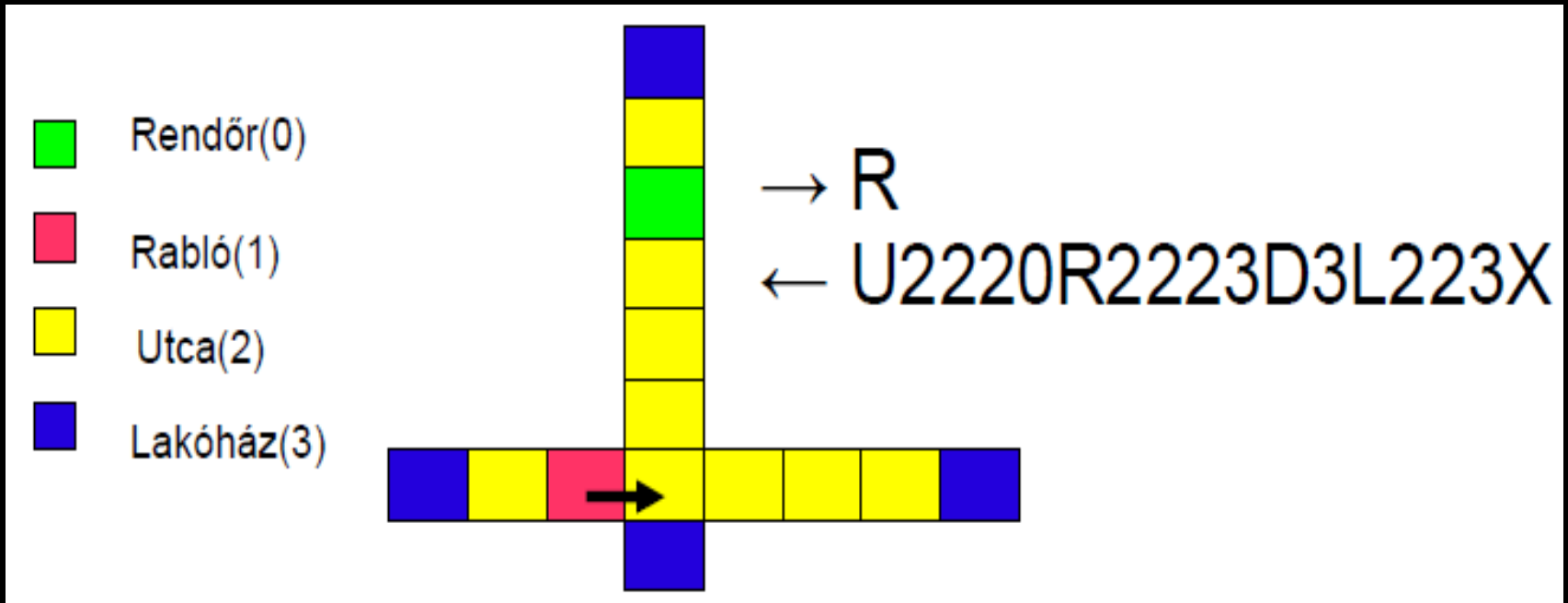
# Communication/2

## Step command (burglar/cop)

Message	Meaning
U,D,L,R	Step (up, down, left, right)
Message	Meaning
F	Stepping is not possible
C	Game over
<i>view</i>	Successful step. <i>view</i> is the view what is viewable path from the new position.

# Communication/3

view := U#{#} D#{#} L#{#} R#{#}X



# *Helper class: player*

```
class Player {
public:
    typedef enum {up, down, left, right}
    direction;
    typedef enum {burglar, cop} player_t;
    Player(char* host, int port, player_t cop);
    char *move(direction d);
    int getx(){ return x; }
    int gety(){ return y; }
    player_t gettyp() { return me; }
    void setx(int x) { this->x = x; }
    void sety(int y) { this->y = y; }
private:
    ...
}
```

# *Tunnel kialakítása*

- A megjelenítő szerver a 15623-on hallgat
- A klienseket (rabló, rendőr) futtató gépen egyedi port kell, hogy többen is dolgozhassanak egyszerre.
- Javaslat 15000+UID:
  - bash esetén:
  - `let RABLO_PORT=15000+`id -u``  
(példa: test.sh)
- Tunnel:
  - `RABLO_PORT -> megjelenítő:15623`

# Putty configuration

Session->SSH->Tunnels

✓ Local ports accept connection from other hosts

Source port:

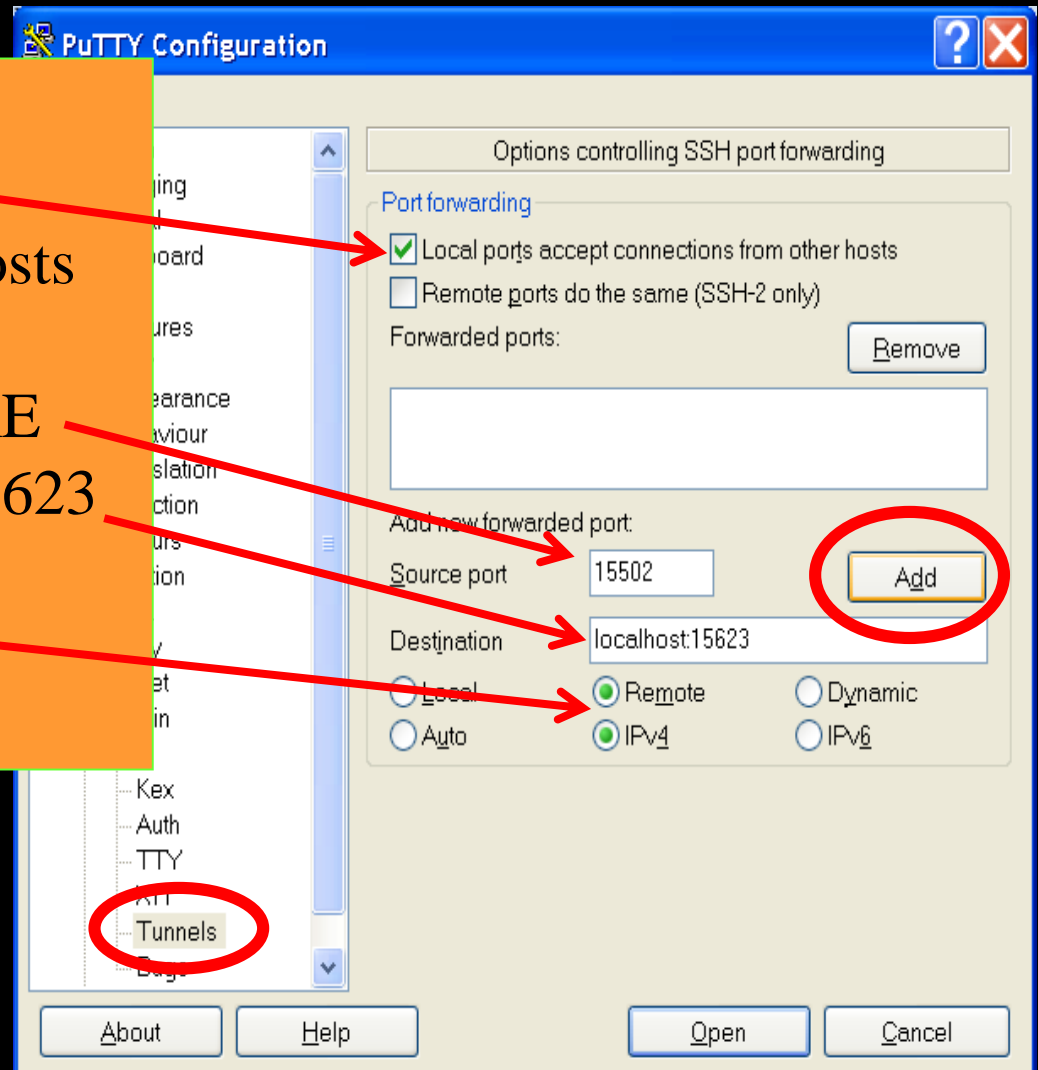
RABLO\_PORT\_ERTEKE

Destination: localhost:15623

✓ Remote

✓ Ipv4

Add



# Teszt

Unpacking the sources:

server -> WinX

client -> para

WinX:

server.bat

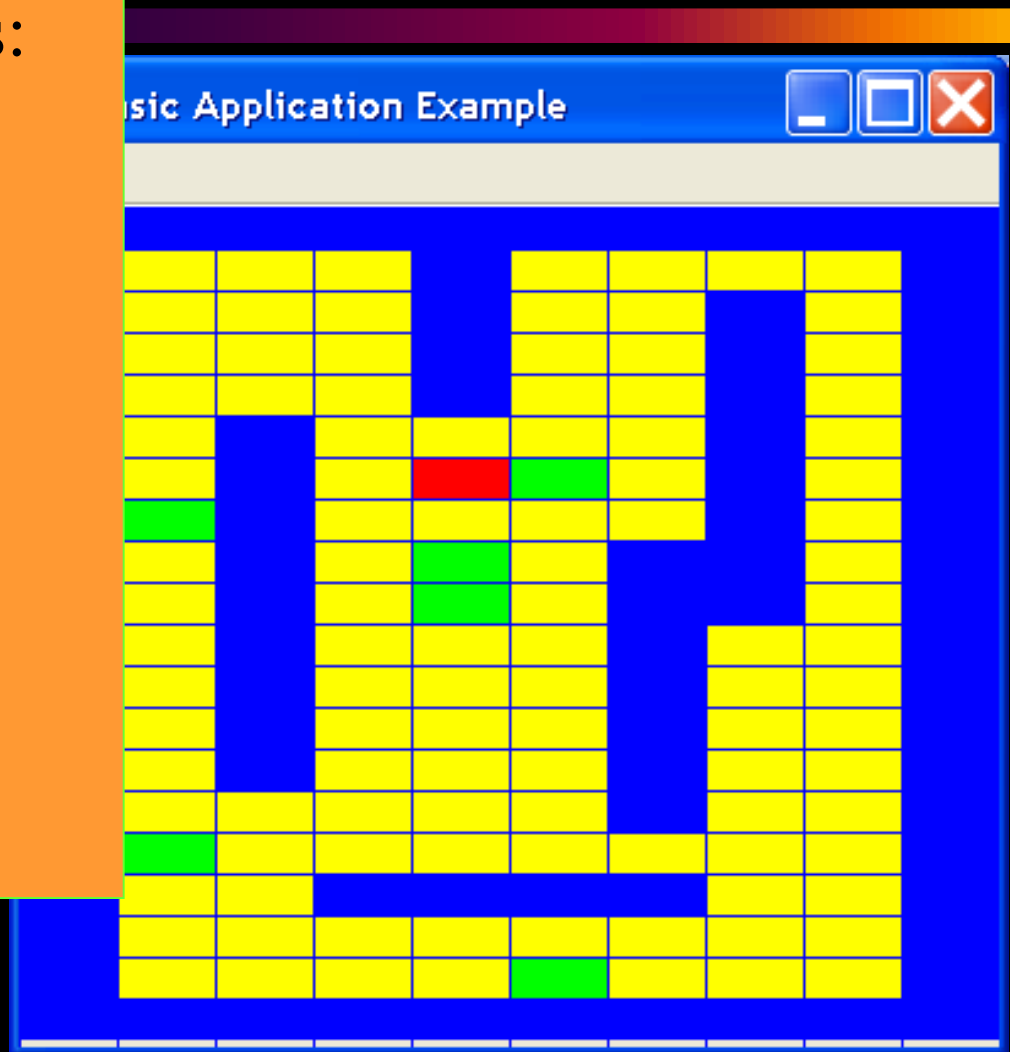
Para:

make

./test.sh,

or

./testmpi.sh



# *Suggestion*

---

- Simple solution:
  - If the cop saw the burglar, broadcasts a message with the exact coordinates.
  - The example program forget to tell the position.

# *message sending/receiving/testing*

```
MPI::Init(argc, argv); // proc. inic
...
MPI::COMM_WORLD.Send(uzenet, hossz, tipus,
    kinek, mit); // blocking send
...
MPI::COMM_WORLD.Recv(buff, hossz, tipus, kitol,
    mit); // blocking receive
...
if (MPI::COMM_WORLD.Iprobe(MPI::ANY_SOURCE,
    TAG_RABLO, statusz)) // test
```

# Assignment



## Helper materials:

- ~szebi/para/MPI
- ~szebi/para/rablo
  - server – viewer program (jar)
  - client – example client programs
  - tests
- Task to be submitted :
  - short description: pdf, or txt
  - sources,
  - makefile