

# PROTOCOALE DE COMUNICAȚIE : LABORATOR 2

## Folosirea unei legaturi de date pentru transmiterea unui fisier

Responsabil: Alecsandru PĂTRAȘCU

### Cuprins

### Cerinte laborator

Se cere sa se implementeze un protocol pentru transmisia de fisiere pornind de la scheletul de cod pus la dispozitie.

Legatura de date dintre transmitator si receptor este simulata prin intermediul executabilului `link`. Mai precis, se simuleaza doua legaturi:

- legatura de date pentru transmiterea datelor (transmitator  $\rightarrow$  receptor)
- legatura de date pentru confirmari (receptor  $\rightarrow$  transmitator)

Ambele legaturi de date nu pierd si nu corup mesaje. Confirmarile se transmit instantaneu.

Protocolul implementat trebuie sa fie de tipul STOP AND WAIT. Astfel, transmitatorul trimite un mesaj cu date, asteapta confirmarea primirii mesajului de la receptor si trimite urmatorul mesaj. Receptorul asteapta mesaje cu date de la transmitator si trimite mesaje de confirmare.

Detalii de implementare:

- transmitatorul primeste ca argument numele fisierului care va fi transmis
- receptorul nu cunoaste numele fisierului care va fi transmis si nici dimensiunea - aceste informatii trebuie primite de la transmitator
- transmitatorul trimite continutul fisierului catre receptor
- receptorul asteapta datele de la transmitator si le scrie intr-un fisier

### Pasi rezolvare

1. Stabilirea tipurilor de mesaje necesare
2. Implementarea transmisiei unui singur fisier
3. *BONUS*: implementare transmisie simultana pentru mai multe fisiere

**Nota:** nu trebuie sa fie modificata structura definita in `lib.h` - se vor utiliza doar campurile `len` si `payload`. Va trebui sa fie definita o noua structura, utila pentru a gestiona tipurile de mesaje stabilite la (1), care sa fie scrisa in `payload`.

### Software disponibil

1. Simulator legatura de date - executabilul `link` - generat in urma comenzii `make`
2. Schelet de cod pentru transmitator si receptor

## 3. API simulator:

- `int send_message(msg* m)`
  - parametru: mesajul care va trimis
  - rezultat: numarul de octeti transferati (in caz de succes) sau -1 in caz de eroare
- `int recv_message(msg* m)`
  - parametru: adresa la care se memoreaza datele primite
  - numarul de octeti receptionati(in caz de succes) sau -1 in caz de eroare

4. Compilare schelet de cod: `make`**Utilitare necesare**

- Generare fisiere de o anumita dimensiune: `dd`
- Verificare corectitudine transmisie: `diff`