Rezolvare

1.

```
class Ftp_srv_parser {
public:
  enum Req_type {req_pwd, req_cwd, req_port, req_retr};
  Ftp_srv_parser(int sd);
  ~Ftp_srv_parser();
  bool read_cmd(Req_type& req, char*& param);
private:
  int f_sd; // socket id
  enum {buf_size=32768};
  char f_buf[buf_size]; //buffer pt. citire
  char* f_pos; // pozitia curenta in buffer
  char* f_end; // ptr. dupa ultimul caracter citit
  char next_ch(); // citeste un caracter,
    // returneaza 0 la end-of-file
};

Ftp_srv_parser(int sd)
{
  f_sd=sd;
  f_pos=f_end=f_buf;
}

Ftp_srv_parser::~Ftp_srv_parser()
{
}

char
Ftp_srv_parser::next_ch()
{
  if(f_pos==f_end){
    int r=read(f_sd, f_buf, buf_size);
    if(r==0){
      return 0;
    }
    f_end=f_buf+r;
    f_pos=f_buf;
```

```
  }
  return *(f_pos++);
}

bool
Ftp_srv_parser::read_cmd(Req_type& req, char*& param)
{
  char buf[buf_size];
  int i;
  char c;
  do {
    c=next_ch();
    buf[i]=c;
    ++i;
  } while(c>' ');
  if(c==0){
    return false;
  }
  buf[i]=0;
  if(strcmp(buf, "WD"==0){
    param=req_pwd;
  }
  if(strcmp(buf, "WD"==0){
    param=req_cwd;
  }
  if(strcmp(buf, "ORT"==0){
    param=req_port;
  }
  if(strcmp(buf, "ETR"==0){
    param=req_retr;
  }
  while(c==' '){
    c=next_ch();
  }
  if(c>' '){
    i=0;
    do {
      buf[i]=c;
      c=next_ch();
      ++i;
    } while(c>' ');
```

```
    buf[i]=0;
    arg=new char[i];
    strcpy(arg, buf);
  }
  if(c=='\r') {
    c=next_ch();
  }
  return true;
}
```

2. Funcţia de citire de pe conexiune fiind blocantă, avem nevoie de câte un fir de execuţie pentru citirea de pe fiecare conexiune, plus un fir pentru citirea de la utilizator.

Deoarece scrierea se poate bloca dacă cititorul conexiunii nu citeşte, avem nevoie cel puţin de câte un fir pentru scrierea pe fiecare conexiune.

Presupunând n parteneri, vom construi $2n + 1$ fire:

- n fire, asociate câte unei conexiuni; fiecare fir execută o buclă în care citeşte un mesaj de pe conexiune şi îl afişează. Este nevoie de sincronizare (prin mutex) la scrierea către utilizator, dar în rest aceste fire nu au nevoie să comunice cu alte fire;
- un fir care citeşte de la utilizator. Pentru fiecare mesaj citit, îl pasează prin câte o zonă tampon fiecărui fir de scriere;
- n fire, câte unul pentru fiecare conexiune, pentru scrierea pe conexiune. Fiecare dintre acestea scrie mesajele din zona tampon.

3.a)

- Masina 1: IP=192.168.0.34/28, default gateway=192.168.0.33
- Masina 2:
  - interfaţa 1 (spre 1): IP=192.168.0.33/28
  - interfaţa 2 (spre 3,4): IP=192.168.0.19/28
  - tabela de dirijare:
    1. 192.168.0.32/28 → interfaţa 1
    2. 192.168.0.16/28 → interfaţa 2
    3. default → gateway 192.168.0.17
- Maşina 3: IP=192.168.0.18/28, default gateway=192.168.0.17

- Maşina 4:
  - intarfaţa 1 (spre 2,3): IP=192.168.0.17/28
  - interfaţa 2 (spre 5,6): IP=192.168.0.3/28
  - tabela de dirijare:
    1. 192.168.0.16/28 → interfaţa 1
    2. 192.168.0.0/28 → interfaţa 2
    3. 192.168.0.32/28 → gateway 192.168.0.19
    4. default → gateway 192.168.0.1
- Maşina 5: IP=192.168.0.2/28, default gateway=192.168.0.1
- Maşina 6:
  - intarfaţa 1 (spre 5,6): IP=192.168.0.1/28
  - interfaţa 2 (spre exterior): IP=193.231.20.34/27
  - tabela de dirijare:
    1. 192.168.0.0/28 → interfaţa 1
    2. 193.231.20.0/27 → interfaţa 2
    3. 192.168.0.16/28 → gateway 192.168.0.3
    4. 192.168.0.32/28 → gateway 192.168.0.3
    5. default → gateway 193.231.20.34

Tabelele de distanţe ale fiecărui nod sunt:

Pasul 1:

Nod 1	Nod 2	Nod 3	Nod 4
$2 \to 1$	$1 \to 1$	$1 \to \infty$	$1 \to 1$
$3 \to \infty$	$3 \to 1$	$2 \to 1$	$2 \to 1$
$4 \to 1$	$4 \to 1$	$4 \to 1$	$3 \to 1$

Pasul 2:

Nod 1	Nod 2	Nod 3	Nod 4
$2 \to 1$	$1 \to 1$	$1 \to$ (prin 2)	$1 \to 1$
$3 \to 2$ (prin 2)	$3 \to 1$	$2 \to 1$	$2 \to 1$
$4 \to 1$	$4 \to 1$	$4 \to 1$	$3 \to 1$

Stabilizarea apare începând cu pasul 2. După căderea legăturii 1–2: Pasul 1:

Nod 1	Nod 2	Nod 3	Nod 4
$2 \rightarrow 2$ (prin 4)	$1 \rightarrow 2$ (prin 4)	$1 \rightarrow$ (prin 2)	$1 \rightarrow 1$
$3 \rightarrow 2$ (prin 4)	$3 \rightarrow 1$	$2 \rightarrow 1$	$2 \rightarrow 1$
$4 \rightarrow 1$	$4 \rightarrow 1$	$4 \rightarrow 1$	$3 \rightarrow 1$

Pasul 2:

Nod 1	Nod 2	Nod 3	Nod 4
$2 \rightarrow 2$ (prin 4)	$1 \rightarrow 2$ (prin 4)	$1 \rightarrow$ (prin 4)	$1 \rightarrow 1$
$3 \rightarrow 2$ (prin 4)	$3 \rightarrow 1$	$2 \rightarrow 1$	$2 \rightarrow 1$
$4 \rightarrow 1$	$4 \rightarrow 1$	$4 \rightarrow 1$	$3 \rightarrow 1$

Stabilizarea apare din nou incepând cu pasul 2.

4. Fiecare proces îşi va numerota mesajele emise. Fiecare mesaj va conţine, pe lângă textul mesajului, următoarele informaţii:

- numărul de ordine al mesajului;
- cele mai recente numere de ordine ale mesajelor primite de la fiecare dintre ceilalţi participanţi (dacă mesajele sunt numerotate de la 1, se poate pune 0 pentru a semnifica faptul că încă nu s-a primit nici un mesaj de la acel partener).

Fiecare procs va ţine în aşteptare un mesaj primit inainte de-a-l afişa până la primirea şi afişarea tuturor mesajelor referite de el.