

Programarea in retea

- OOB-

Lenuta Alboaie
adria@info.uaic.ro

Cuprins

- Trimiterea si receptarea datelor in regim *out-of-band*

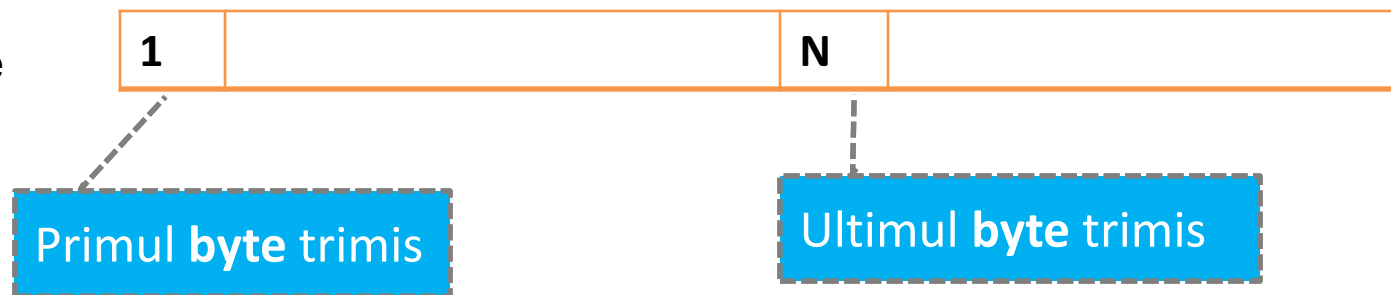
Trimiterea si receptarea datelor in regim out-of-band

- Ideea: in timpul unei conexiuni cind sunt transmise date (“**in-band data**”), si daca la un capat se intimpla “ceva” acesta va dori sa transmita rapid celuilalt *peer* in regim de prioritate, o notificare (“**out-of-band data**”)
- Mecanismul de realizare
 - Se utilizeaza bitul URG setat in antetul TCP
 - Antetul TCP contine un camp indicind locatia datelor urgente ce trebuie trimise
 - **Trimiterea datelor OOB:**
 - Pentru a expedia un octet urgent intr-un flux de date putem utiliza **send()**:
send (sd, buff, 1, **MSG_OOB**);

Trimiterea si receptarea datelor in regim out-of-band

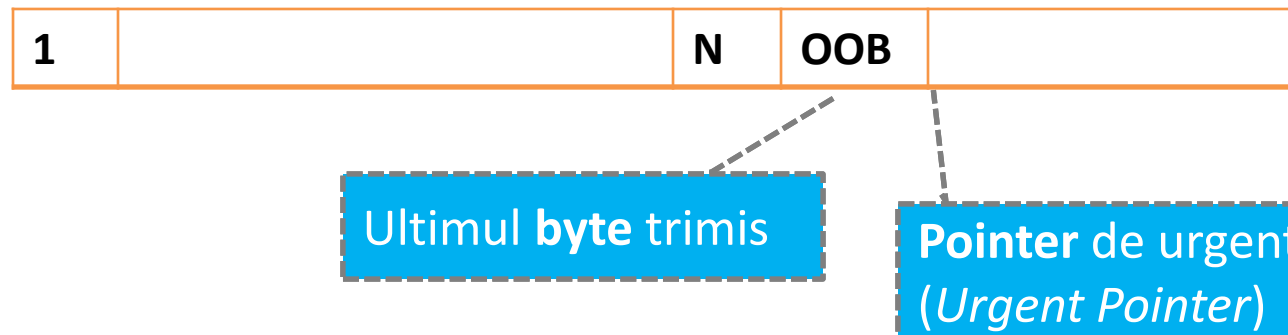
Trimiterea unui octet OOB

Buffer-ul
de trimitere
TCP



send (sd, "a", 1, **MSG_OOB**) Transmiterea unui
byte OOB

Buffer-ul
de trimitere
TCP



Trimiterea si receptarea datelor in regim out-of-band

Receptionarea datelor OOB:

- Se genereaza semnalul **SIGURG**
- Apelul **select()** va modifica lista descriptorilor de exceptie

Citirea datelor OOB:

- Daca socketul nu are asociata optiunea **SO_OOBINLINE**, mesajul OOB este plasat intr-un buffer special (*out-of-band buffer*); citirea bufferului se poate realiza cu **recv()** sau **recvmsg()** setind **MSG_OOB**
- Daca socket-ul are asociata optiunea **SO_OOBINLINE**, mesajul OOB este plasat in *buffer*-ul normal de primire;
- Procesul va sti ca a ajuns la *acel octet* in functie de valoarea *out of-band-mark* asociata conexiunii

Trimiterea si receptarea datelor in regim out-of-band

Primitiva `socketmark()`

- Cind se primesc date OOB, se face asocierea cu *out-of-band-mark*, reprezentind pozitia datelor OOB in streamul de date trimis de emitator
- `socketmark()` asigura ca procesul receptor sa determine daca conexiunea are sau nu asociata marca *out-of-band*

```
#include <sys/socket.h>
```

```
int socketmark (int sockfd) ;
```

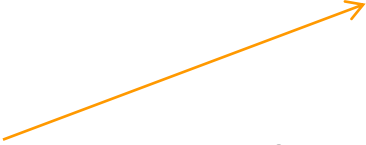
Returneaza: 1 – este *out-of-band mark*,

0 – nu este *out-of-band mark*

–1 on error

Trimiterea si receptarea datelor in regim out-of-band

Primitiva `socketatmark()` – observatii

- *out-of-band mark* se aplica indiferent daca procesul receptor primeste datele in mod *out-of-band inline* (socket-ul cu optiunea `SO_OOBINLINE`) sau in mod *out-of-band* (flagul `MSG_OOB`)
 - Este implementata folosindu-se `ioctl()` si `SIOCATMARK`
int value;
error = `ioctl(tcp_socket, ioctl_type, &value);`
- 

Trimiterea si receptarea datelor in regim out-of-band

Primitiva `socketmark()` – discutii & exemplu

- Daca s-a primit date in mod *OOB inline* `socketmark()` intoarce `true` daca urmatorul octet ce poate fi citit a fost trimis cu flagul `MSG_OOB`
- Daca *socket*-ul nu are asociata optiunea `SO_OOBINLINE`, `socketmark()` intoarce *true* daca urmatorul octet ce poate fi citit este primul octet care a fost trimis dupa data OOB
- Operatia de citire se opreste in functie de *out of-band-mark*
 - Exemplu:

Daca sunt 100 de octeti in buffer dar doar 5 octeti pina la *out of-band-mark*, chiar daca procesul citește 100 de octeti el va primi initial doar primii 5 octeti

Trimiterea si receptarea datelor in regim out-of-band

Erori posibile

- Se asteapta citirea de date OOB, dar ele nu au fost inca trimise – se returneaza **EINVAL**
- Procesul a fost notificat ca va primi date OOB (via **select()** sau **SIGURG**), el incearca sa le citeasca dar ele inca nu au ajuns – se returneaza **EWOULDBLOCK**
- Se incearca sa se citeasca un acelasi octet **OOB** de mai multe ori – se returneaza **EINVAL**
- Daca procesul are setat optiunea **SO_OOBINLINE**, dar incearca sa citeasca cu flagul **MSG_OOB** – se returneaza **EINVAL**

Trimiterea si receptarea datelor in regim out-of-band

- Utilizari:
 - Modalitate de comunicare a celuiilalt punct terminal a unei conditii de exceptie chiar si in cazul cind controlul fluxului a oprit emitatorul
 - Pentru a detecta timpuriu erori de comunicare intre client si server (*heart-beat*)

Trimiterea si receptarea datelor in regim out-of-band

Tratare **OOB** prin **SIGURG**:

```
if (listen (sd, 5) == -1)  { ... }
...
while (1)
{
    ... client = accept (sd, (struct sockaddr *) &from, &len);
    signal(SIGURG, urgHandler);
    fcntl(client, F_SETOWN, getpid()); /*setarea proprietarului socketului conectat */
    for(;;)  {
        if((n=read(client,msg,sizeof(msg)-1))==0)
        {
            printf("Am primit EOF\n");
            break;
        }
        else
        {
            msg[n]=0;
            printf("Am citit %d octeti: %s\n",n, msg);
        }
    } ... } //while
void urgHandler(int signnr)
{
    int n;          char buff[100];
    printf("SIGURG e primit\n");
    n=recv(client,buff, sizeof(buff)-1, MSG_OOB);
    buff[n]='\0';
    printf("Am citit %d octet OOB %s\n",n, buff);
}
```

Trimiterea si receptarea datelor in regim out-of-band

Tratare **OOB** folosind primitiva **socketmark()**

DEMO

- **Emitatorul** trimite 4 octeti de date normale, apoi un octet OOB si inca un octet de date normale
- **Receptorul** nu foloseste SIGURG sau select(); el apeleaza primitiva **socketmark()**

Rezumat

- Trimiterea si receptarea datelor in regim out-of-band

Bibliografie

- UNIX Network Programming: The sockets networking API, W. Richard Stevens, Bill Fenner, Andrew M. Rudoff
- The Illustrated Network: How TCP/IP Works in a Modern Network (The Morgan Kaufmann Series in Networking), Walter Goralski



Intrebari?