

Nonblocking IO

Veerle Ongenae

Blokkeren

- Actie oproepen
 - Kan niet meteen starten
 - Proces in slaapmode totdat actie kan starten

aanroepen die blokkeren :

- invoer (recv)
 - Geen data, wachten op invoer
- uitvoer (send)
 - Indien buffer vol, wachten tot er ruimte vrij is
- inkomende verbindingen (accept)
- maken verbinding (connect)

→ EWOULDBLOCK of EINPROGRESS (bij connect)

sockets nonblocking maken

- optie instellen met fcntl :

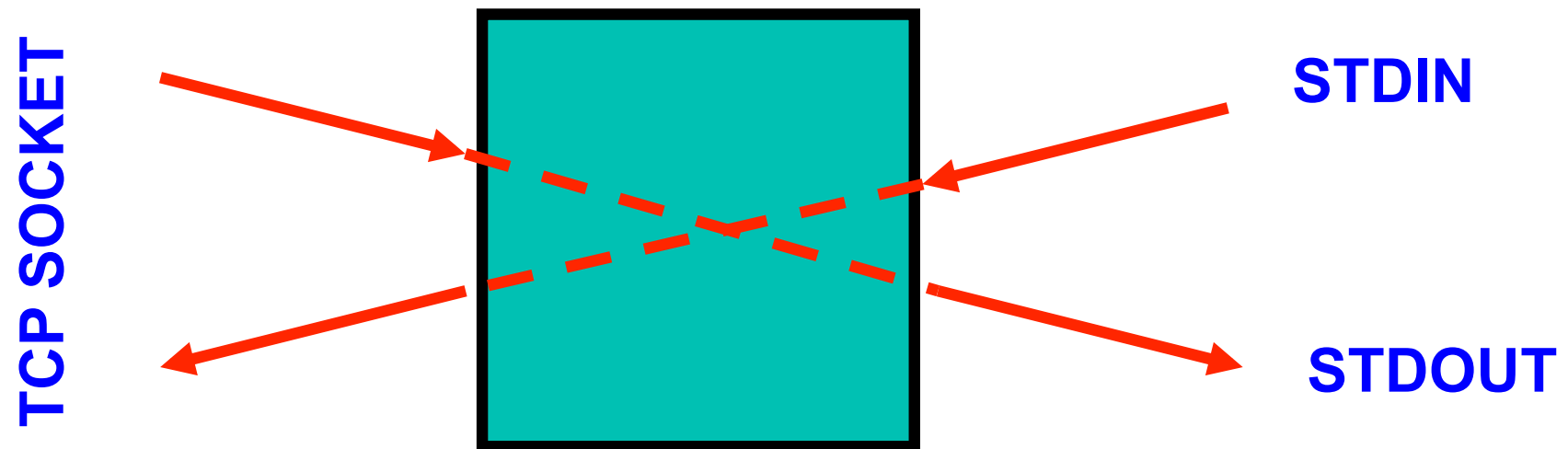
```
val = fcntl(sockfd, F_GETFL, 0);  
fcntl(sockfd, F_SETFL, val | O_NONBLOCK);
```

- Zelf invoer en uitvoer beheren
 - Leesbuffer to (invoer --> socket)
 - Schrijfbuffer from (socket --> uitvoer)

Voorbeeld

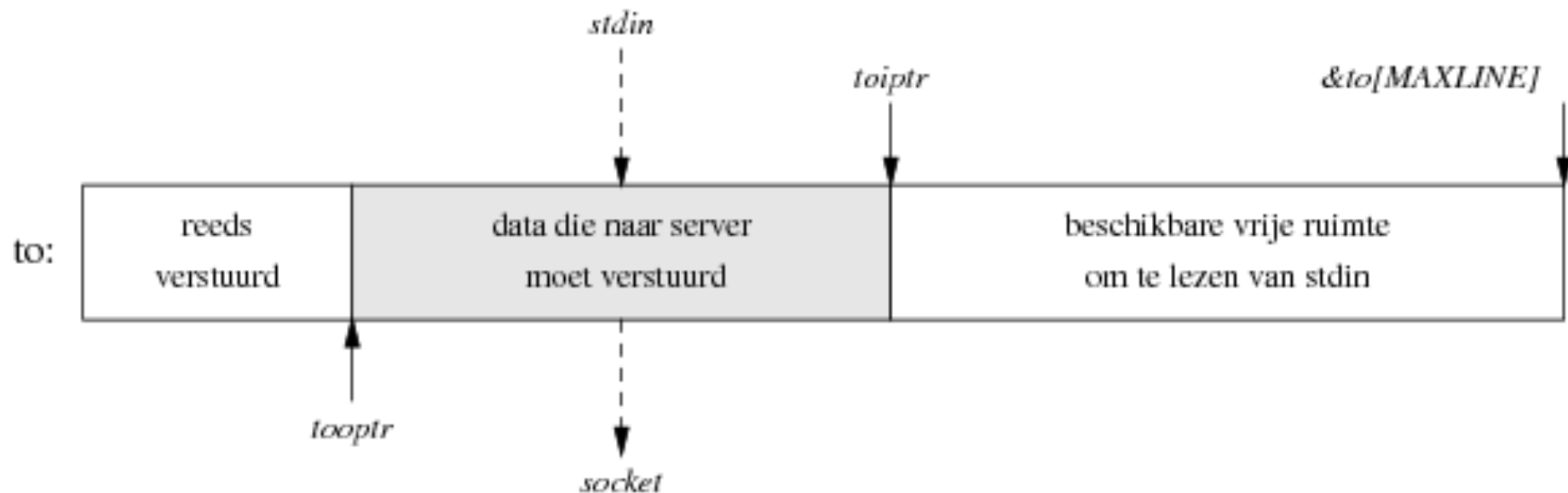
- Vb. Echoclient
 - Invoer toetsenbord → versturen met socket
 - Invoer socket → tonen op scherm

Generische TCP Client



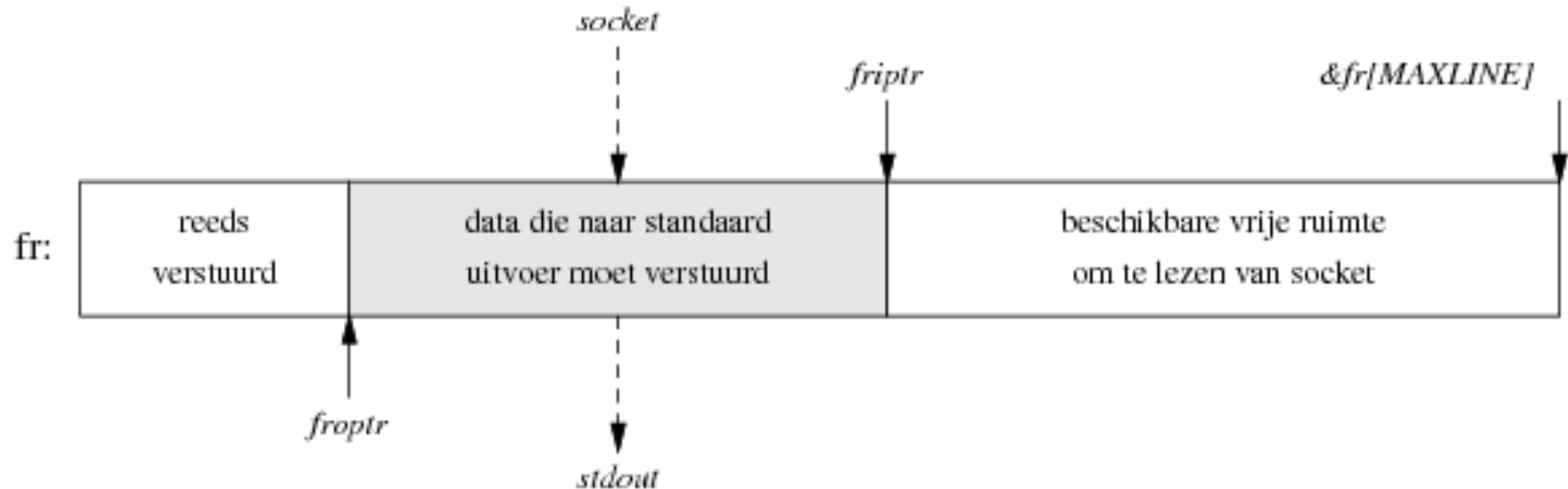
Nonblocking met buffer

- Toetsenbord → socket



Nonblocking met buffer

- socket → scherm





Deel 1 : init

- Argumenten inlezen
- Socket aanmaken
- Connectie met server
- Verschillende kanalen non-blocking maken
- Buffers aanmaken
- fd_set's aanmaken voor lezen en schrijven
- select uitvoeren

Deel 2 : testen invoer?

- Invoer?
- Lezen
 - Tot maximaal einde buffer
 - Pointer verschuiven



Deel 3: testen uitvoer?

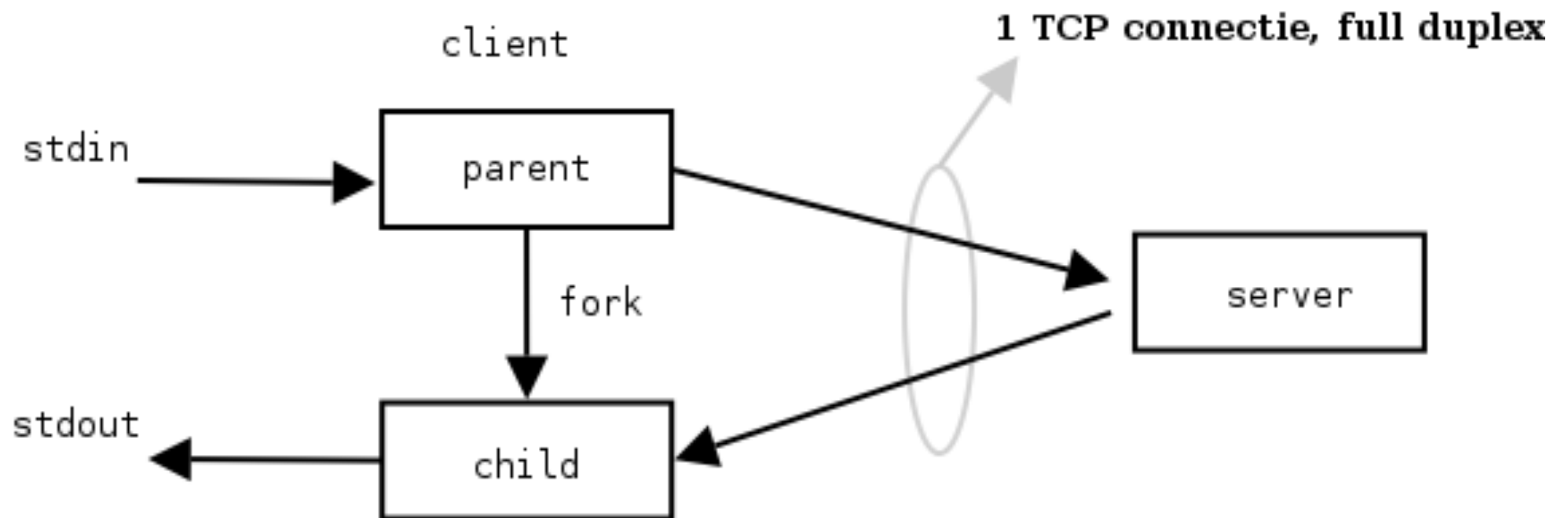
- Nog informatie in buffer?
- Kanaal beschikbaar?
- Schrijf naar kanaal
- Verschuif pointer
- Indien alles geschreven
 - Beide pointers naar begin



Nonblocking met kind

- Ongewenst blokkeren van applicatie verhelpen m.b.v. kindprocessen
- Eenvoudige implementatie
- Minder efficiënt

Nonblocking met kind





Code voorbeeld echo client

- `echo_client_kind.cpp`