

# Laboratorio di Reti di Calcolatori

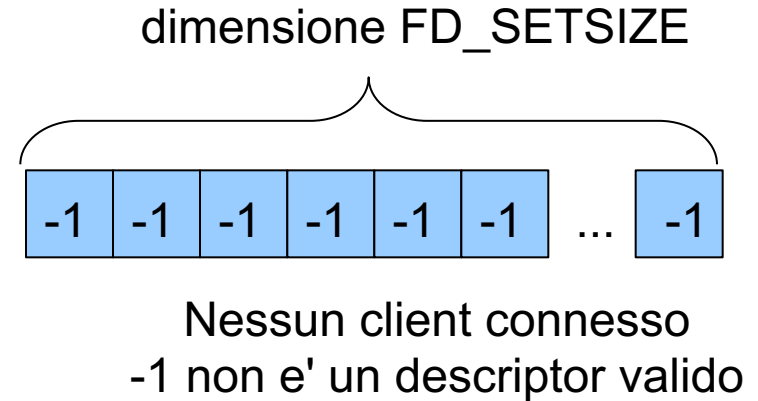
## Lezione 8

# Server basato su I/O Multiplex

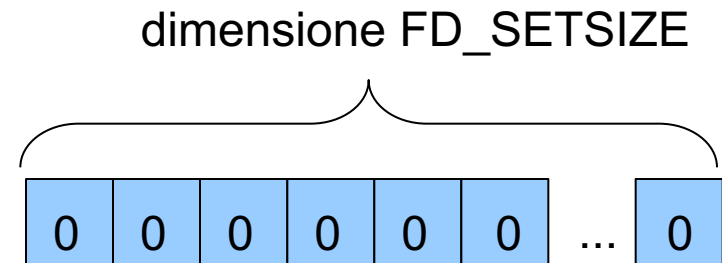
- Per realizzare un server è possibile utilizzare l'I/O Multiplex
- Un unico processo iterativo gestisce il socket che accetta nuove connessioni e tutti i socket connessi
- Il server memorizza:
  - L'insieme di descrittori di tipo `fd_set` da monitorare con `select`
  - Un array di interi che memorizza i socket descriptor relativi alle connessioni con i client

# Strutture dati Server

Array di descrittori dei  
socket connessi con i client  
`connsd[ ]`

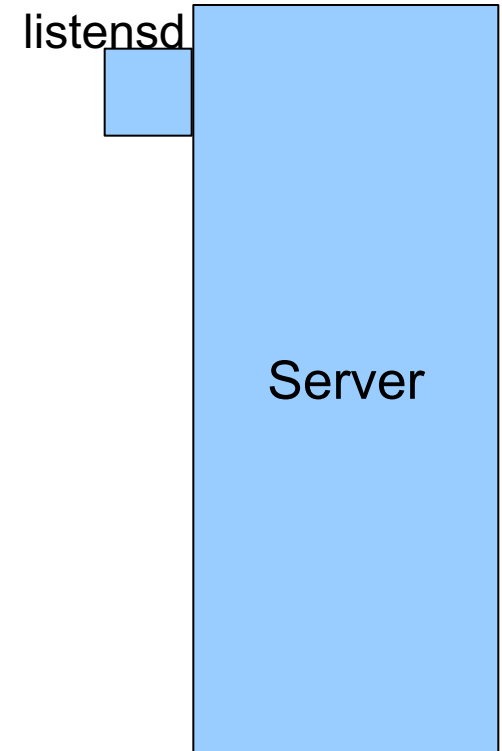


`fd_set` dei descrittori  
da monitorare in lettura



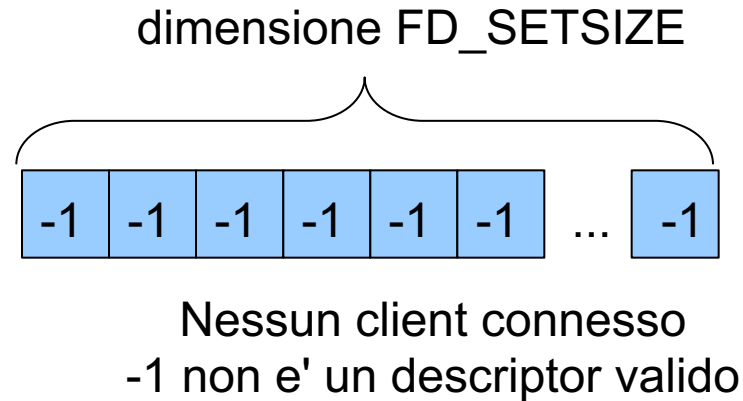
# Server basato su I/O Multiplex

Il server in origine ha solo il socket in ascolto  
Escludendo standard in out ed err il primo descrittore libero per il socket in ascolto è 3

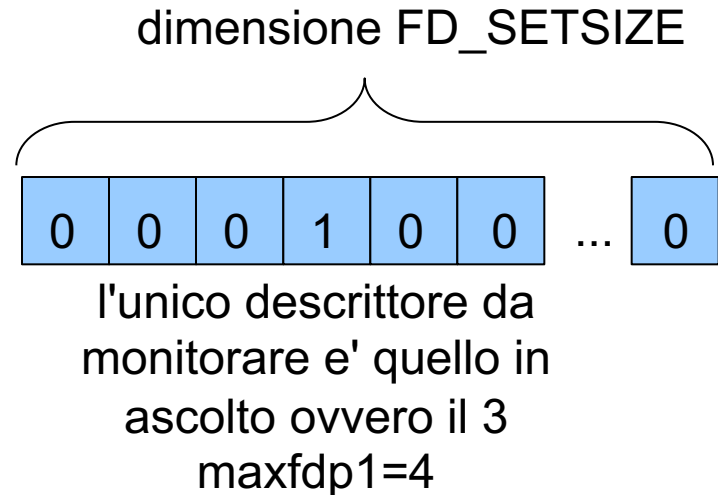


# Strutture dati Server

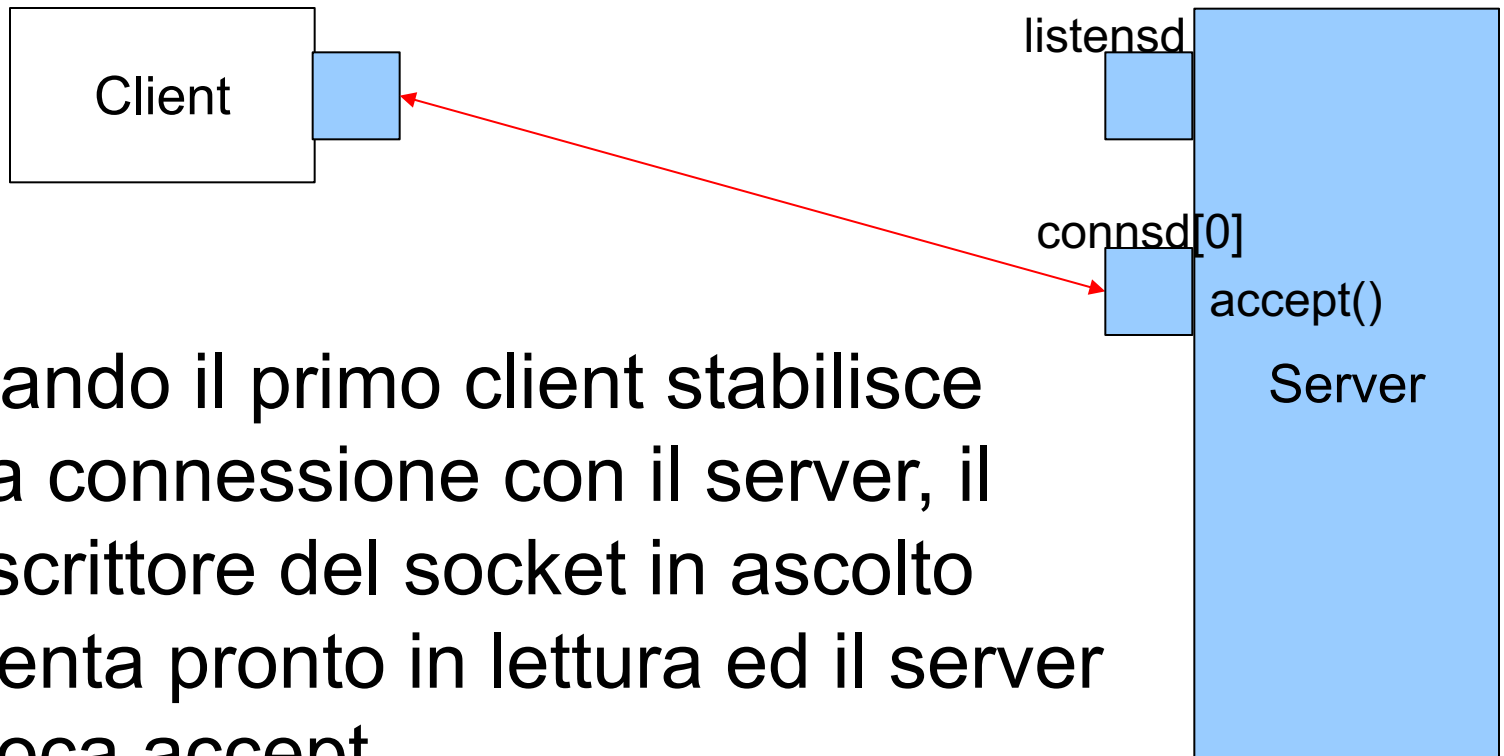
Array di descrittori dei  
socket connessi con i client  
`connsd[ ]`



`fd_set` dei descrittori  
da monitorare in lettura



# Server basato su I/O Multiplex

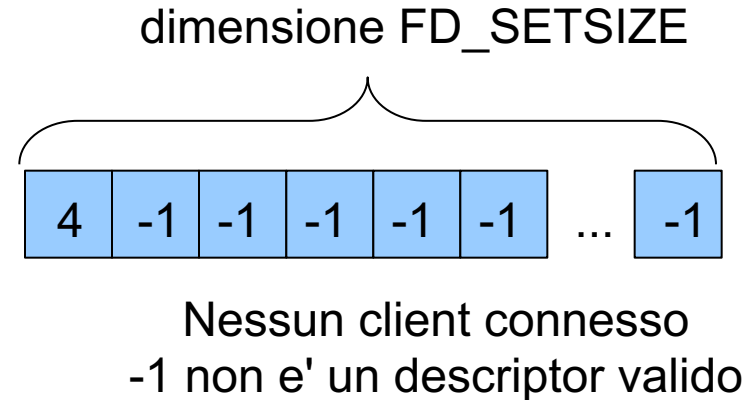


Quando il primo client stabilisce una connessione con il server, il descrittore del socket in ascolto diventa pronto in lettura ed il server invoca `accept`

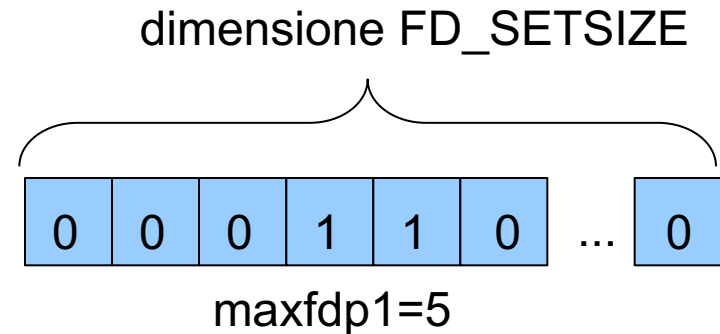
Si supponga che Il socket descriptor restituito da `accept` sia 4

# Strutture dati Server

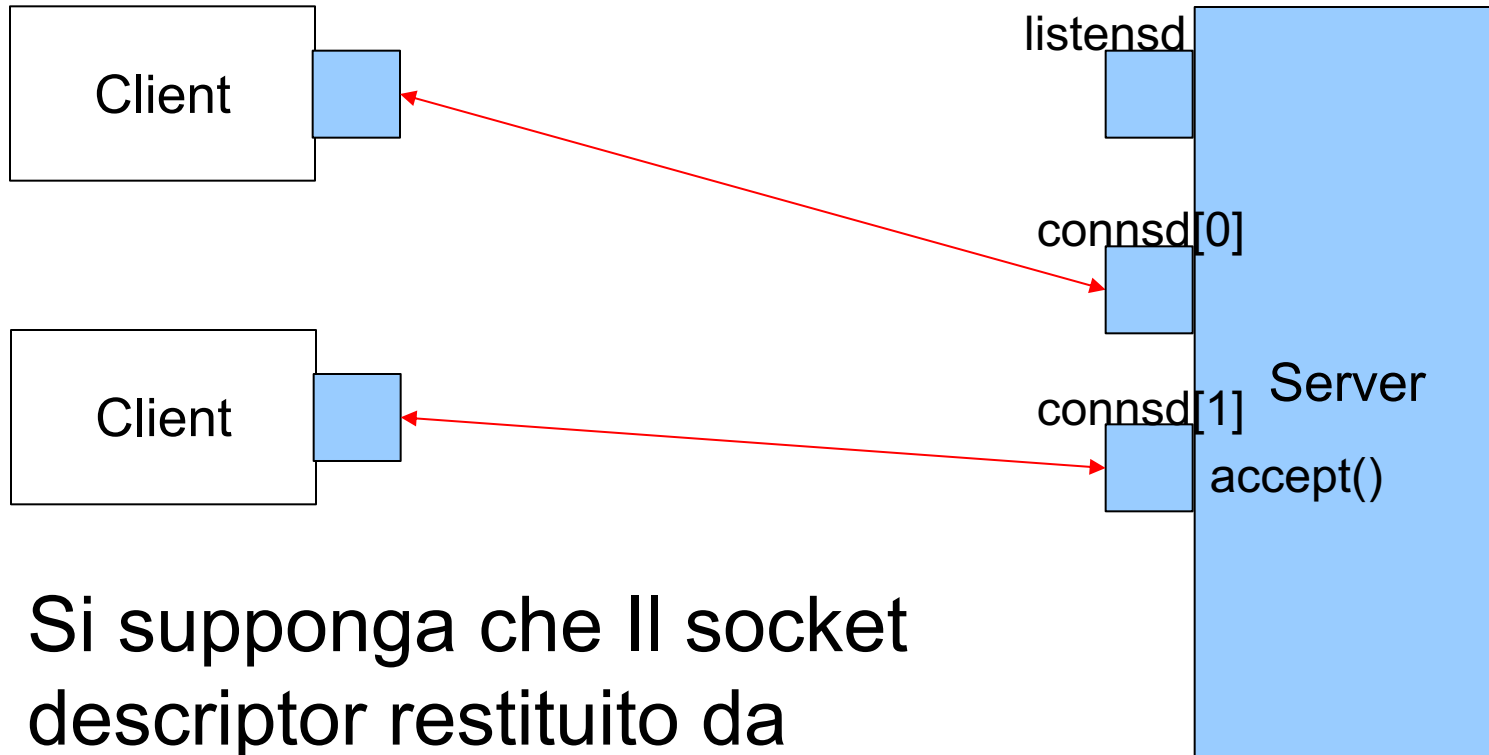
Array di descrittori dei socket connessi con i client  
`connsd[ ]`



`fd_set` dei descrittori da monitorare in lettura



# Server basato su I/O Multiplex

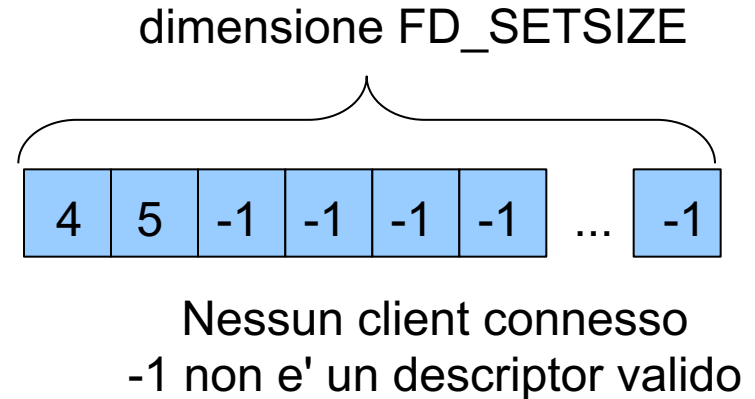


Si supponga che Il socket descriptor restituito da `accept` dopo la connessione del secondo client sia 5

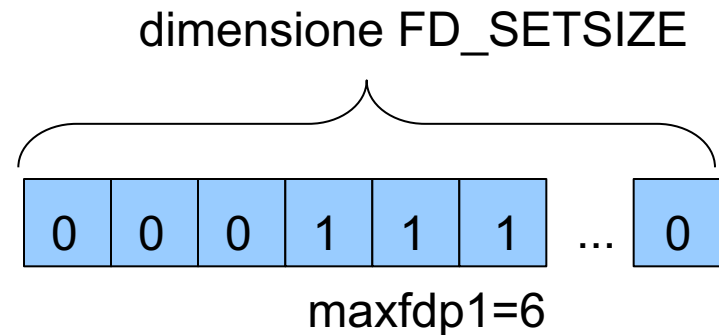


# Strutture dati Server

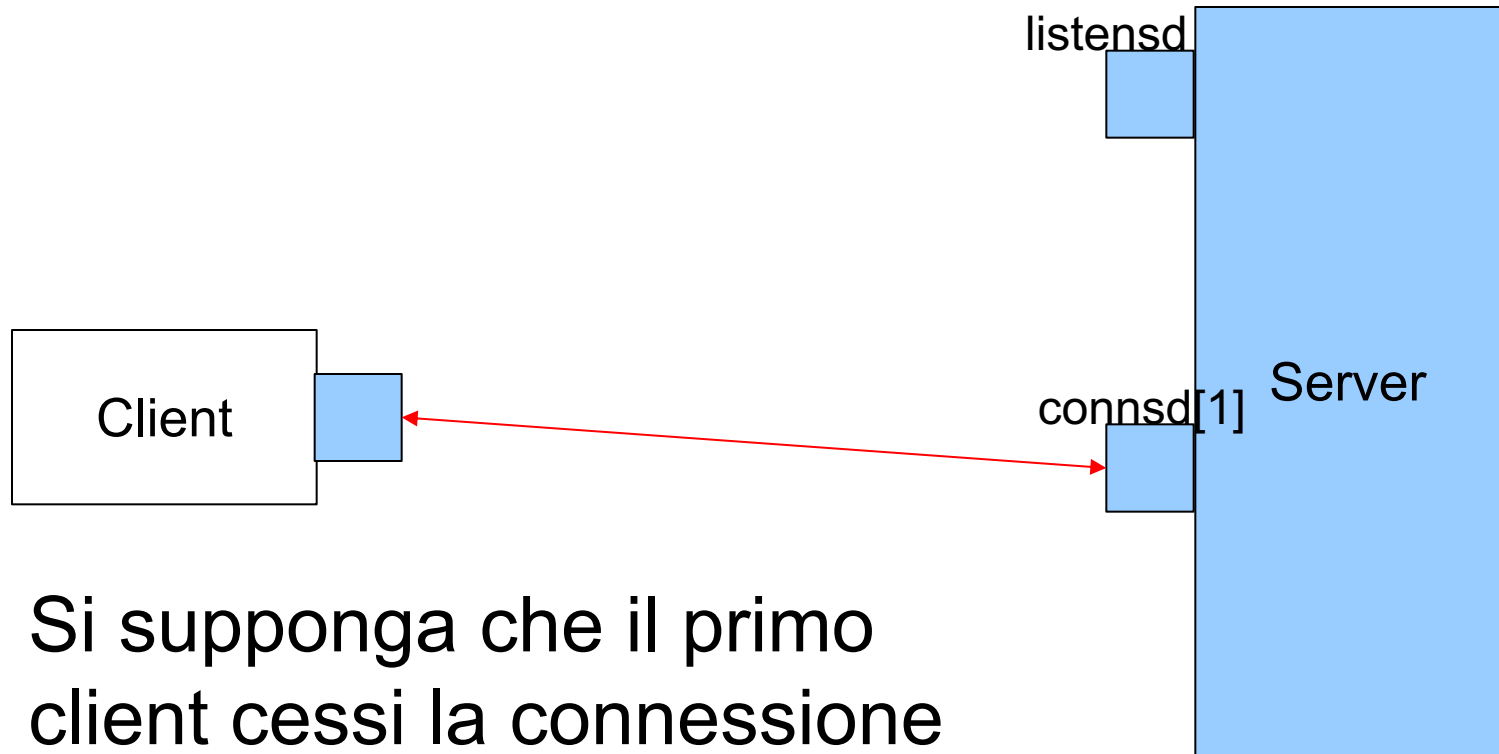
Array di descrittori dei  
socket connessi con i client  
connsd[ ]



fd\_set dei descrittori  
da monitorare in lettura



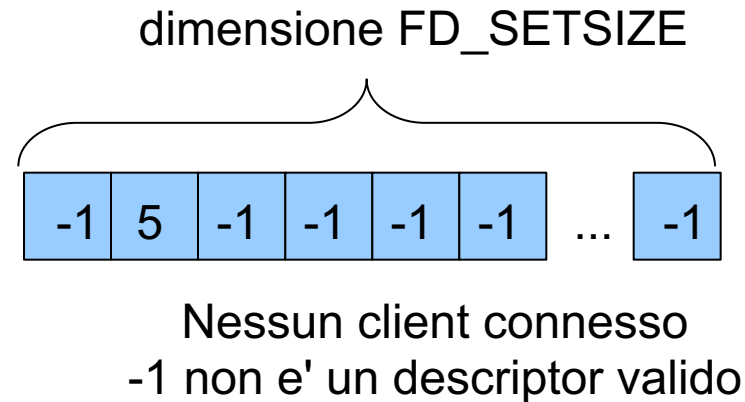
# Server basato su I/O Multiplex



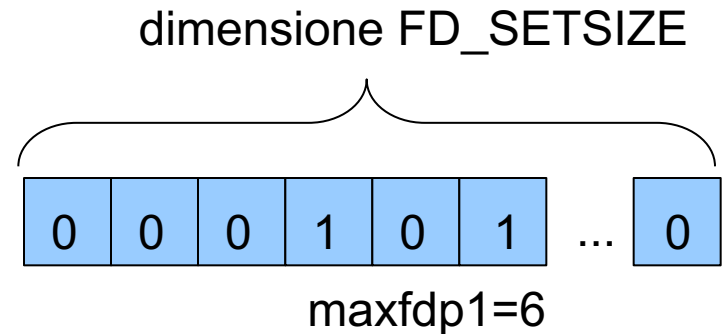
Si supponga che il primo client cessi la connessione

# Strutture dati Server

Array di descrittori dei socket connessi con i client  
`connsd[ ]`



`fd_set` dei descrittori da monitorare in lettura



# Server I/O multiplex

```
max_fd = list_fd
fd_open[max_fd] = 1
while(1)
    for (i = list_fd; i <= max_fd; i++)
        if (fd_open[i] != 0)
            FD_SET(i, &fset)
    n=select(mafd+1, &fset, ...)
    if (FD_ISSET(list_fd, &fset)
        n--
        fd=accept(list_fd, ...)
        fd_open[fd] = 1
        if (max_fd<fd)
            max_fd=fd
    i=list_fd
```

# Server I/O multiplex

```
while(n!=0)
    i++
    if( fd_open[i]==0)
        continue
    if(FD_ISSET(i,&fset))
        n--
        nread=FullRead(i,...)
        if(nread<0)
            error
        if(nread==0)
            fd_open[i]=0
            if (max_fd == i)
                while (fd_open[--i] ==0);
                max_fd = i
                break;
        FullWrite(i, ...)
```

# Esercizi

- Si riscriva il server che conta i caratteri in modo che gestisca le connessioni mediante I/O mutliplex