

Gniazda BSD

Zbigniew Suski

Gniazda BSD

1

Procesy w środowisku sieciowym

- ❑ Obsługa
- ❑ Protokół
- ❑ Numery portów i ogólnie znane adresy
- ❑ Połączenie (*asocjacja*) i gniazdo (*pólasocjacja*)
- ❑ Model *klient-serwer*
 - Serwer - bierne otwarcie kanału
 - Klient - czynne otwarcie kanału
- ❑ Serwer iteracyjny i współbieżny
- ❑ Obsługa w trybie połączeniowym i bezpołączeniowym

Zbigniew Suski

Gniazda BSD

2

Gniazda – podstawowe funkcje dla serwera

- | | |
|------------------------|---|
| ❑ Utwórz punkt końcowy | <code>socket()</code> |
| ❑ Dowiąż adres | <code>bind()</code> |
| ❑ Określ kolejkę | <code>listen()</code> |
| ❑ Czekaj na połączenie | <code>accept()</code> |
| ❑ Prześlij dane | <code>read()</code> , <code>write()</code>
<code>recv()</code> , <code>send()</code> |
| ❑ Prześlij datagramy | <code>recvfrom()</code> , <code>sendto()</code> |
| ❑ Zakończ | <code>close()</code> , <code>shutdown()</code> |

Zbigniew Suski

Gniazda BSD

3

Gniazda – podstawowe funkcje dla klienta

- | | |
|------------------------|---|
| ❑ Utwórz punkt końcowy | <code>socket()</code> |
| ❑ Dowiąż adres | <code>bind()</code> |
| ❑ Połącz z serwerem | <code>connect()</code> |
| ❑ Prześlij dane | <code>read()</code> , <code>write()</code>
<code>recv()</code> , <code>send()</code> |
| ❑ Prześlij datagramy | <code>recvfrom()</code> , <code>sendto()</code> |
| ❑ Zakończ | <code>close()</code> , <code>shutdown()</code> |

Zbigniew Suski

Gniazda BSD

4

Gniazda – funkcje podstawowe

<sys/types.h>, <sys/socket.h>

```
int socket ( int  rodzina,
             int  typ,
             int  protokół )

int bind (   int  deskryptor,
             struct sockaddr *adres_własny,
             int  rozmiar )

int close ( int  deskryptor )
```

Zbigniew Suski

Gniazda BSD

5

Gniazda – funkcje podstawowe

```
int connect (int  deskryptor,
             struct sockaddr *adres_serwera,
             int  rozmiar )

int listen ( int  deskryptor,
            int  liczba_żądań )

int accept ( int  deskryptor,
            struct sockaddr *adres_partnera,
            int  *rozmiar )
```

Zbigniew Suski

Gniazda BSD

6

Gniazda – funkcje podstawowe

```
int send ( int  deskryptor,
           char  *bufor,
           int  ilość,
           int  flagi )
```

```
int recv ( int  deskryptor,
           char  *bufor,
           int  ilość,
           int  flagi )
```

Zbigniew Suski

Gniazda BSD

7

Gniazda – funkcje podstawowe

```
int sendto ( int  deskryptor,
            char  *bufor,
            int  ilość,
            int  flagi,
            struct sockaddr *odbiorca,
            int  rozmiar )
```

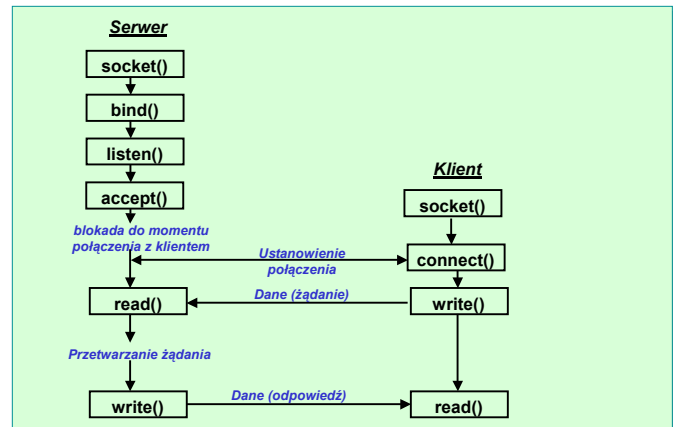
```
int recvfrom (int  deskryptor,
            char  *bufor,
            int  ilość,
            int  flagi,
            struct sockaddr *nadawca,
            int  *rozmiar )
```

Zbigniew Suski

Gniazda BSD

8

Protokół połączeniowy

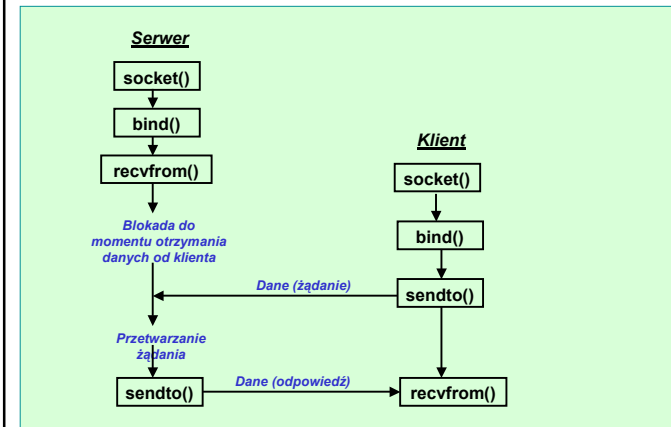


Zbigniew Suski

Gniazda BSD

9

Protokół bezpołączeniowy



Zbigniew Suski

Gniazda BSD

10

Przekształcanie kolejności bajtów

Kolejność *mnijšie niżej*
mnijšie wyżej
sieciowa (w ramach protokołu)

```
#include < sys/types.h >
#include < netinet/in.h >
```

Od stacji do sieci:

```
u_long    htonl ( u_long hostlong )
u_short   htons ( u_short hostshort )
```

Od sieci do stacji:

```
u_long    ntohl ( u_long netlong )
u_short   ntohs ( u_short netshort )
```

Zbigniew Suski

Gniazda BSD

11

Operacje na ciągach bajtów

Kopiowanie
bcopy (char **źródło*,
 char **wynik*,
 int *liczba_bajtów*)

Powielanie bajtów pustych
bzero (char **wynik*,
 int *liczba_bajtów*)

Porównanie ciągów bajtów
bcmp (char **źródło1*,
 char **źródło2*,
 int *liczba_bajtów*)
 Zwraca 0 gdy ciągi są identyczne.

[System V](#)
memset
memcpy
memcmp

Zbigniew Suski

Gniazda BSD

12

Przekształcanie adresów (Internet)

```
#include < sys/socket.h >
#include < netinet/in.h >
#include < arpa/inet.h >

unsigned long inet_addr ( char *łańcuch )
char *inet_ntoa ( struct in_addr *adres )
```

Zbigniew Suski

Gniazda BSD

13

Gniazda BSD - inne funkcje

readv, writev	czytanie i pisanie do/z buforów nieprzyległych
sendmsg, recvmsg	jw. (tylko dla gniazd)
getpeername	pobranie identyfikacji procesu partnera
getsockname	pobranie identyfikacji gniazda
gethostbyname	pobranie identyfikacji hosta
gethostbyaddr	pobranie identyfikacji
getsockopt	pobranie atrybutów (ustawień gniazda)
setsockopt	ustawienie atrybutów gniazda

Zbigniew Suski

Gniazda BSD

14

Nadrzędny serwer Internetu (inetd)

```
#ident "@(#)inetd.conf 1.16 94/03/08 SMI" /* SVr4.0 1.5 */

# Ftp and telnet are standard Internet services.
#
ftp    stream tcp  nowait  root    /usr/sbin/in.ftpd  in.ftpd
telnet stream tcp  nowait  root    /usr/sbin/in.telnetd in.telnetd
#
# Shell, login, exec, comsat and talk are BSD protocols.
#
shell  stream tcp  nowait  root    /usr/sbin/in.rshd  in.rshd
login  stream tcp  nowait  root    /usr/sbin/in.rlogind in.rlogind
exec   stream tcp  nowait  root    /usr/sbin/in.rexecd in.rexecd
talk   dgram  udp    wait    root    /usr/sbin/in.talkd  in.talkd
#
pop3   stream tcp  nowait  root    /opt/SUNWpop/sbin/popper
#
```

Zbigniew Suski

Gniazda BSD

15

Nadrzędny serwer Internetu (inetd)

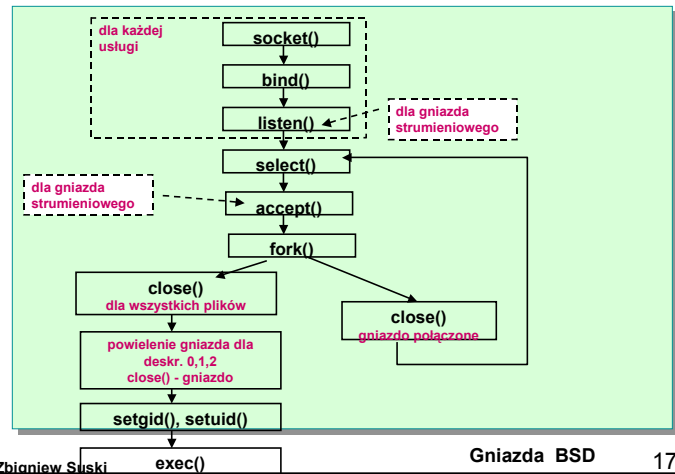
```
# @(#)services;      Network services, Internet style
#
echo      7/tcp
echo      7/udp
netstat   15/tcp
ftp        21/tcp
telnet     23/tcp
smtp      25/tcp      mail
time      37/tcp      timserver
time      37/udp      timserver
domain    53/udp      nameserver
#
# UNIX specific services
exec       512/tcp
login      513/tcp
```

Zbigniew Suski

Gniazda BSD

16

Nadrzędny serwer Internetu (*inetd*)



Zbigniew Suski

Gniazda BSD

17

Przykłady użycia gniazd

Przykład:

plik [Gniazda-przyklady-TCP-UDP](#)

Zbigniew Suski

Gniazda BSD

18

Gniazda surowe

Tworzenie gniazda surowego

```
int gniazdo;
gniazdo = socket( AF_INET, SOCK_RAW, protokół);
```

- ☐ Stałe określające protokół zdefiniowane są w pliku *netinet/in.h*. Np. `IPPROTO_ICMP`.
- ☐ Nagłówek IP jest tworzony przez jądro.
- ☐ Gniazdo surowe może tworzyć tylko nadzorca systemu.

Zbigniew Suski

Gniazda BSD

19

Gniazda surowe

Wysyłanie danych

sendto lub *sendmsg*.

Odbieranie danych

recvfrom

- ☐ Nie są przekazywane odebrane pakiety TCP i UDP.
- ☐ Nie są przekazywane niektóre odebrane pakiety ICMP.
- ☐ Są przekazywane wszystkie odebrane pakiety IGMP.
- ☐ Są przekazywane wszystkie odebrane, nierozpoznane pakiety IP.

Zbigniew Suski

Gniazda BSD

20

Gniazda surowe

Opcja IP_HDRINCL

```
const int w = 1;  
setsockopt( gniazdo, IPPROTO_IP, IP_HDRINCL,  
           &w, sizeof(w))
```

- ❑ Adres początkowy danych odsyłanych do jądra odpowiada pierwszemu bajtowi nagłówka IP.
- ❑ Samodzielnie należy zbudować nagłówek IP z wyjątkiem:
 - pola sumy kontrolnej nagłówka IP,
 - pola identyfikacji.
- ❑ Rozmiar odsyłanych danych powinien uwzględniać rozmiar nagłówka IP.

Gniazda surowe

Tworzenie gniazda pakietowego

```
int gniazdo;  
gniazdo = socket(AF_PACKET, SOCK_RAW, protokół);
```

- ❑ Stałe określające protokół zdefiniowane są w pliku *if_ether.h*. Np. *ETH_P_ALL* lub *ETH_P_ARP*.
- ❑ Należy utworzyć całą ramkę (łącznie z nagłówkiem)
- ❑ Gniazdo pakietowe może tworzyć tylko nadzorca systemu

Przykłady: PING

Przykład:

plik [Gniazda-przyklady-PING](#)

Przykłady: protokół ARP

Przykład:

plik [Gniazda-przyklady-ARP](#)