Obsługa gniazd sieciowych pod linux'em.

Krzysztof Dynowski

Rodzaje gniazd

- połączeniowe (SOCK_STREAM), tylko pakety unicast, komunikacja pewna
  klasa A 0-127.a.b.c/255.0.0.0
  klasa B 128-191.a.b.c/255.255.0.0
  klasa C 192-223.a.b.c/255.255.255.0
  zakresy prywatne: 10.0.0.0/8(A), 172.16.0.0/12(16xB), 192.168.0.0/16(256xC)
- bezpołączeniowe (SOCK_DGRAM), komunikacja niepewna
  unicast – do jednego odbiorcy
  broadcast – do wszystkich (ip=255.255.255.255, 192.168.0.255 dla 192.168.0.1/24)
  multicast – do grupy odbiorców, adresy 224.0.0.0 - 239.255.255.255
  anycast – podobny do multicast, ale do jednego (najbliższego) z wielu a nie do grupy

Tworzenie gniazda (*man 2 socket)*
int socket(int domain, int type, int protocol);
domain – domena adresowa, określa zakres stosowalności gniazda
PF_LOCAL(PF_UNIX) – lokalnie (poł. w obrębie jednej maszyny)
PF_INET – internet (poł. sieciowe TCP/IP)
PF_IPX – novell
PF_PACKET – komunikacja niskopoziomowa, warstwa fizyczna
PF_xxx (Protocol Family) jest używane zamiennie z AF_xxx (Address Family)
type – sposób komunikacji
SOCK_STREAM – komunikacja strumieniowa, połączeniowa, pewna (TCP)
SOCK_DGRAM – komunikacja datagramowa, bezpołączeniowa, niepewna (UDP)
SOCK_RAW – komunikacja na poziomie ramek warstwy sieciowej (IP)
SOCK_PACKET – komunikacja na poziomie ramek warstwy fizycznej (ethernet)
protocol – protokół którego chcemy używać, zazwyczaj jest tylko jeden i należy podać wartość 0
zwraca deskryptor gniazda lub -1 w przypadku błędu
przykłady użycia:
s=socket(PF_INET,SOCK_STREAM,0);
s=socket(PF_INET, SOCK_DGRAM,0);
s=socket(PF_INET,SOCK_RAW,IPPROTO_ICMP);

Nawiązanie połączenia, klient TCP (*man 2 connect)*
*int connect(int s, struct sockaddr *addr, int addrlen);*
s – deskryptor gniazda, uzyskany z funkcji *socket*
addr – wskaźnik do struktury opisującej adres docelowy
addrlen – rozmiar przekazanej struktury (w bajtach)

Przypisanie adresu i portu nasłuchiwania, serwer TCP/UDP (*man 2 bind)*
*int bind(int s, const struct sockaddr *addr, socklen_t addrlen);*
s – deskryptor gniazda
addr – wskaźnik do struktury opisującej adres lokalny
addrlen – rozmiar przekazanej struktury (w bajtach)

Przypisanie długości kolejki połączeń, serwer TCP (*man 2 listen)*
int listen(int s, int backlog);
s – deskryptor gniazda
backlog – długość kolejki połączeń oczekujących

Przyjęcie(odebranie) połączenia, serwer TCP (*man 2 accept)*
*int accept(int s,const struct sockaddr *addr, socklen_t *addrlen);*
s – deskryptor gniazda (serwera)
addr – wskaźnik do struktury opisującej adres zdalny
addrlen – rozmiar odebranej struktury (w bajtach)

Wysłanie danych/zapytania (*man 2 send)*
*int send(int s, const void *buf, int len, unsigned int flags);*
s – deskryptor gniazda
buf – bufor z danymi do wysłania
len – ilość danych (w bajtach)
flags – flagi (opcje) wysyłania
MSG_OOB wysłanie danych 'out of band' (tylko dla SOCK_STREAM)
MSG_DONTWAIT f.nieblokująca (zwróci błąd jeśli dane nie mogą być wysłane natychmiast)

Odebranie danych/odpowiedzi (*man 2 recv)*
*int recv(int s, void *buf, int len, unsigned int flags);*
s – deskryptor gniazda
buf – bufor z danymi do wysłania
len – ilość danych (w bajtach)
flags – flagi (opcje) odbierania
MSG_OOB odebranie danych 'out of band' (tylko dla SOCK_STREAM)
MSG_PEEK odczyt danych bez usuwania z bufora
MSG_WAITALL odebranie wszystkich żądanych bajtów

Zamknięcie gniazda (*man 2 shutdown)*
int shutdown(int s, int how);
s – deskryptor gniazda
how – określa sposób zamknięcia
0 – odbieranie nie będzie możliwe
1 – wysyłanie nie będzie możliwe
2 – odbieranie/wysyłanie nie będzie możliwe

Zwolnienie deskryptora (*man 2 close)*
int close(int s); automatycznie wywołuje *shutdown(s,2)*
s – deskryptor gniazda

Konwersja reprezentacji liczb
htons, htonl – konwersja z organizacji bajtów hosta na sieciową (BigEndian)
ntohs, ntohl – konwersja z organizacji bajtów sieciowej na hosta

Multicast
Wysyłanie – zwykły pakiet UDP wysłany na adres grupy multicast.

Wysyłanie – należy ustawić odpowiednio TTL (zasięg pakietu):
unsigned char ttl=1; //subnet
setsockopt(s,IPPROTO_IP,IP_MULTICAST_TTL, &ttl,sizeof(ttl);
Odbieranie – należy:
1. przypisać adres i port nasłuchiwania do gniazda

*bind(sfd,(struct sockaddr *)&sock,sizeof(sock));*
2. dołączyć gniazdo do grupy multicast:

hp=gethostbyname(mcgroup);
memcpy(&mcaddr,hp->h_addr_list[0],hp->h_length);
mreq.imr_multiaddr.s_addr=mcaddr.s_addr;
mreq.imr_interface.s_addr=htonl(INADDR_ANY);
r=setsockopt(sfd,IPPROTO_IP,IP_ADD_MEMBERSHIP,&mreq,sizeof(mreq));

Znaczenie flagi TTL (zasięg pakietów)
0 – w obrębie tego samego hosta
1 – w jednej podsieci
<32 – w jednej „organizacji”
<64 – w jednym „regionie”
<128 – w jednym kontynencie
<255 – zasięg globalny

Adresy zarezerwowane
224.0.0.1 Wszystkie systemy w tej podsieci
224.0.0.2 Wszystkie routery w tej podsieci
224.0.0.5 routery DVMRP
224.0.0.5 routery OSPF
224.0.0.13 routery PIM
239.0.0.0-239.255.255.255 przestrzeń prywatna