Programowanie z wykorzystaniem usług TCP/IP

- Informacje wykorzystywane przez programistę: adresy IP, porty
- Klasyfikacja portów: BSD a ustalenia IANA
  - BSD: reserved, ephemeral(1024-5000), non-privileged
  - IANA: well-known, registered(1024-49151), dynamic
- Identyfikacja połączenia TCP (socket-pair)
- Identyfikatory połączenia dla serwerów
- Gniazdo (socket)
- Typy gniazd w zależności od protokołu
- Blokowanie operacji wykonywanych na gniazdach
- Ograniczenia wielkości przesyłanych danych: MTU a MSS
  - Bufory nadawcze: scenariusze dla TCP i UDP

Modele komunikacji z wykorzystaniem gniazd

Bezpołączeniowy

- Utwórz gniazdo
- Powiąż gniazdo z konkretnym SAP
- Wysyłaj/odbieraj dane
- Zamknij gniazdo

Obydwie aplikacje

Połączeniowy

Aplikacja "klient"	*Aplikacja "serwer"*
• Utwórz gniazdo • Powiąż gniazdo z konkretnym SAP	
• Wyślij żądanie nawiązania poł. • Jeśli połączenie nawiązane wysyłaj/odbieraj dane	• Ustaw gniazdo w tryb nasłuchu i czekaj • Jeśli nadeszło żądanie, utwórz nowe gniazdo i powiąż je z nowym klientem • Wysyłaj/odbieraj dane
• Zamknij gniazdo (rozwiąż połączenie)	

Struktury i typy danych

Adresy:

```
struct in_addr {
    in_addr_t  s_addr; }
```

```
struct sockaddr {
   sa_family_t  sa_family;
   char         sa_data[14];
 }
```

```
struct sockaddr_in {
      sa_family_t   sin_family;
      in_port_t     sin_port;
      struct in_addr    sin_addr;
      char          sin_zero[8];
}
```

```
struct sockaddr_un {
      sa_family_t   sun_family;
      char          sun_path[ ];
 }
```

sin_len ?

Struktury i typy danych

Konwersje formatu danych:

```
uint_16  htons ( uint_16 value )
uint_16  ntohs ( uint_16 value )
uint_32  htonl ( uint_32 value )
uint_32  ntohl ( uint_32 value )
```

Konwersje adresu IP:

```
in_addr_t  inet_addr ( const char *address )
int  inet_aton ( const char *address, struct in_addr *adstr )
char  *inet_ntoa ( struct in_addr adstr )
```

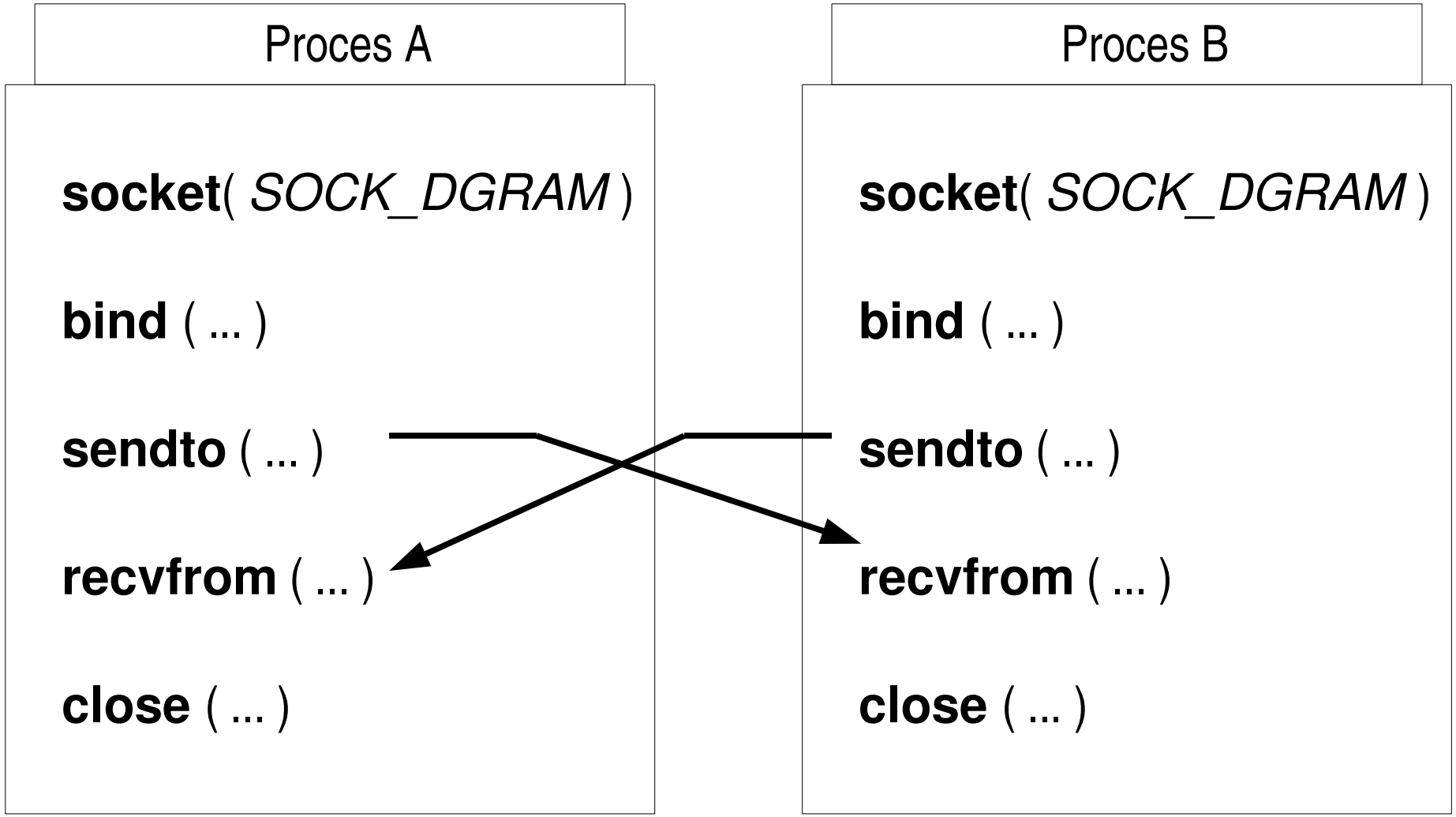
Ważniejsze stałe:

- Rodziny adresów/protokołów - AF_INET,AF_UNIX
- Rodzaje gniazd: SOCK_STREAM,SOCK_DGRAM

Podstawowe funkcje

- int **socket**(int *domain*, int *type*, int *protocol*)
- int **bind**(int *sockfd*, struct sockaddr **addr*, int *addrlen*)
- int **sendto**(int *sockfd*, const void **msg*, int *len*, unsigned int *flags*, struct sockaddr **addr*, int *addrlen*)
- int **recvfrom**(int *sockfd*, const void **msg*, int *len*, unsigned int *flags*, struct sockaddr **addr*, int **addrlen*)
- int **connect**(int *sockfd*,struct sockaddr **addr*, int *addrlen*)
- int **listen**(int *sockfd*,int *backlog*)
- int **accept**(int *sockfd*, void **addr*,int **addrlen*)
- int **write**(int *sockfd*,void **data*,int *len*) [**send**]
- int **read**(int *sockfd*,void **data*,int *len*) [**recv**]

Schemat trybu bezpołączeniowego



Schemat trybu połączeniowego