

© 2002-2010 Volnys Bernal 1

Cliente UDP

Volnys Borges Bernal
volnys@lsi.usp.br
<http://www.lsi.usp.br/~volnys>



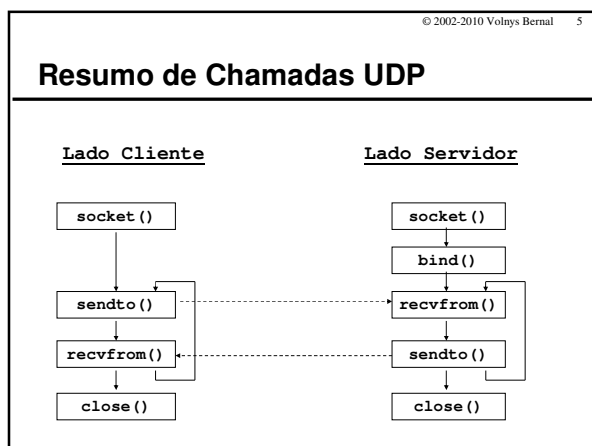
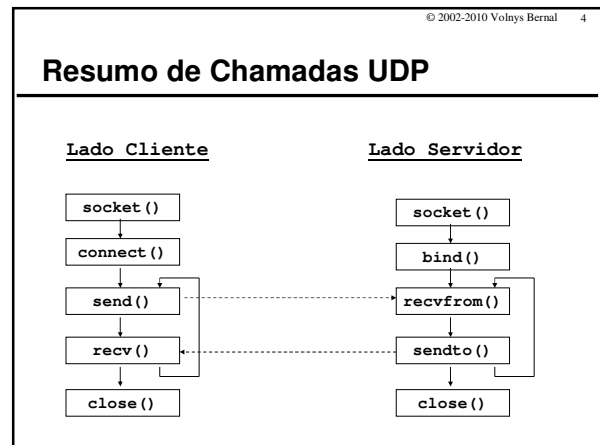
© 2002-2010 Volnys Bernal 2

Agenda

- ❑ Resumo de Chamadas UDP
- ❑ Chamada socket()
- ❑ Chamada connect()
- ❑ Chamada send()
- ❑ Chamada recv()
- ❑ Chamada sendto()
- ❑ Chamada recvfrom()
- ❑ Chamada close()

© 2002-2010 Volnys Bernal 3

Chamadas UDP

© 2002-2010 Volnys Bernal 6

Chamada socket()



© 2002-2010 Volnys Bernal 7

Chamada socket()

- ❑ **Objetivo**
 - ❖ Criar um novo socket (plug de comunicação)
- ❑ **Resultado**
 - ❖ Retorna um descritor de arquivo.
- ❑ **Sintaxe**

```
sd = socket (int domain, int type, int protocol)
```
- ❑ **Observação:**
 - ❖ Quando um socket é criado ele não possui nenhuma informação armazenada (IPs e portas).
 - ❖ Estas devem ser informadas nas chamadas bind() (lado servidor) e connect() (lado cliente).

© 2002-2010 Volnys Bernal 8

Chamada socket()

- ❑ **Sintaxe**

```
#include <sys/socket.h>
int socket(int domain, int type, int protocol)
```

Socket descriptor

Pilha de protocolos
 - PF_INET
 - PF_INET6
 - PF_X25

Id. do protocolo
 - UDP (17)
 - TCP (6)

Tipo da comunicação
 - SOCK_STREAM (TCP)
 - SOCK_DGRAM (UDP)
 - SOCK_RAW (IP)

© 2002-2010 Volnys Bernal 9

Chamada socket()

- ❑ **Tipo de serviço**
 - ❖ SOCK_STREAM
 - Fluxo de bytes sem delimitação
 - Full duplex
 - Orientada a conexão
 - Comunicação confiável:
 - sem perda de dados, sem duplicação, entrega na ordem
 - Deve ser utilizado com o protocolo TCP
 - Chamadas para transmissão e recepção de dados
 - read(), write() ou send(), recv()
 - ❖ SOCK_DGRAM
 - Datagrama (mensagens)
 - Deve ser utilizada com o protocolo UDP
 - Chamadas para transmissão e recepção de dados
 - send(), recv()
 - ❖ SOCK_RAW
 - Permite acesso a protocolos de mais baixo nível
 - Datagrama (mensagens)
 - Chamadas para transmissão e recepção de dados
 - send(), recv()

© 2002-2010 Volnys Bernal 10

Chamada socket()

- ❑ **Exemplo de criação de socket TCP**

```
#include <sys/socket.h>

int sd; // socket descriptor

...
sd = socket(PF_INET, SOCK_STREAM, 6)
if (sd == -1)
    perror("Erro na chamada socket");
...
```
- ❑ **Obs:**
 - ❖ O valor 6 representa o protocolo TCP, foi obtido consultando /etc/protocols. Pode também ser obtido através de resolução de nomes via getprotobyname().

© 2002-2010 Volnys Bernal 11

Chamada socket()

- ❑ **Exemplo de criação de socket UDP**

```
int sd; // socket descriptor

...
sd = socket(PF_INET, SOCK_DGRAM, 17);
if (sd == -1)
    perror("Erro na chamada socket");
...
```
- ❑ **Obs:**
 - ❖ O valor 17 representa o protocolo UDP, foi obtido consultando /etc/protocols. Pode também ser obtido através de resolução de nomes via getprotobyname().

© 2002-2010 Volnys Bernal 12

Chamada Connect()



© 2002-2010 Volnys Bernal 13

Chamada connect()

- ❑ **Objetivo**
 - ❖ Define uma sessão de comunicação TCP, UDP ou IP
 - ❖ Informa à pilha da comunicação o endereço (IP+porta) do parceiro de comunicação
- ❑ **Funcionamento**
 - ❖ Deve ser executado no lado cliente
 - ❖ UDP:
 - Não são enviados datagramas
 - ❖ TCP:
 - Ativa o processo de estabelecimento de conexão TCP

© 2002-2010 Volnys Bernal 14

Chamada connect()

- ❑ **Sintaxe**

```
#include <netdb.h>
int connect(int sd,
            struct sockaddr *serversockaddr,
            int size)
```

© 2002-2010 Volnys Bernal 15

Chamada connect()

```
#include <netdb.h>

int          status;           //estado da chamada
struct sockaddr_in serversockaddr; //endereço do servidor
...

// define endereço destino
serversockaddr.sin_family = AF_INET;
serversockaddr.sin_port   = htons(serverport);
status = inet_pton(AF_INET, stringIP, &serversockaddr.sin_addr);
if (status <= 0)
    perror("Erro na conversão do endereço IP");

// ativa connect
status = connect( socketdescriptor,
                 (struct sockaddr *)&serversockaddr,
                 sizeof(serversockaddr) );

if (status != 0)
    perror("Erro na chamada connect");
```

© 2002-2010 Volnys Bernal 16

Chamada connect

- ❑ **Exemplo de iniciação da estrutura sockaddr_in**

© 2002-2010 Volnys Bernal 17

Chamada send()

© 2002-2010 Volnys Bernal 18

Chamada send()

- ❑ **Função para transmissão de dados**
- ❑ **Pode ser utilizada por clientes e servidores**

```
int send(int sd, void * sendbuff, int msgsize, int flags)
```

Chamada send()

Exemplo:

```

. . .
status = send(sd, sendbufp, msgsize, 0)
if (status <= 0)
    perror("Erro na chamada send");
. . .

```

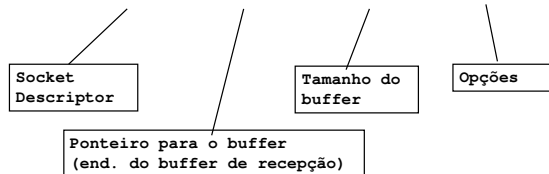
Chamada recv()



Chamada recv()

- ❑ Recebimento de datagramas
- ❑ Pode ser utilizada por clientes e servidores

```
int recv(int sd, void * rcvbufp, int rcvbufsize, int flags)
```



Chamada recv()

Exemplo:

```

. . .
status = recv(sd, rcvbufp, buffersize, 0)
if (status <= 0)
    perror("Erro na chamada recv");
. . .

```

Chamada recv()

Bloqueante

- ❖ Se não existirem mensagens na fila de recepção o processo fica aguardando sua chegada
- ❖ Exceção: quando o socket for criado como não bloqueante (ver `fcntl(2)`).

Retorno

- ❖ Se a chamada tiver sucesso, o valor retornado é o tamanho do datagrama

Chamada sendto()



© 2002-2010 Volnys Bernal 25

Chamada sendto()

- ❑ Função para transmissão de dados (datagramas)
- ❑ Para um destino especificado
- ❑ Não deve ser precedido por connect()
- ❑ Pode ser utilizado pelo cliente ou pelo servidor

Socket
Descriptor

Ponteiro para
buffer da mensagem

Tamanho da
mensagem

```
int sendto(int sd, void * msgp, int msgsize, int flags,
           struct sockaddr *to, int sockaddrsz)
```

Endereço
do destino

Tamanho da
estrutura
sockaddr

Opções

© 2002-2010 Volnys Bernal 26

Chamada sendto()

- ❑ Exemplo:

```
...
status = sendto(sd, msgp, msgsize, 0,
                (struct sockaddr *) &toaddr,
                sizeof(struct sockaddr))

if (status <= 0)
    perror("Erro na chamada sendto");
...
```

© 2002-2010 Volnys Bernal 27

Chamada recvfrom()



© 2002-2010 Volnys Bernal 28

Chamada recvfrom()

- ❑ Recebimento de datagramas
- ❑ Permite obter a identificação do parceiro
- ❑ Pode ser utilizado pelo cliente ou servidor

Socket
Descriptor

Ponteiro para
buffer da mensagem

Tamanho do
buffer

```
int recvfrom(int sd, void * buffer, int buffersize, int flags,
             struct sockaddr *from, int *sockaddrsz)
```

Endereço
da origem

Tamanho da
estrutura
sockaddr

Opções

© 2002-2010 Volnys Bernal 29

Chamada recvfrom()

- ❑ Exemplo:

```
char          bufferp[80];
struct sockaddr_in fromaddr; // sockaddr do cliente
int          size;

...
size = sizeof(struct sockaddr_in);
status = recvfrom(sd, bufferp, 80, 0,
                  (struct sockaddr *) &fromaddr,
                  &size);

if (status <= 0)
    perror("Erro na chamada recvfrom");
...
...
```

© 2002-2010 Volnys Bernal 30

Chamada recvfrom()

- ❑ Bloqueante
 - ❖ Se não existirem mensagens na fila de recepção o processo fica aguardando sua chegada
 - ❖ Exceção: quando o socket for criado como não bloqueante (ver fcntl(2)).
- ❑ Retorno
 - ❖ Se a chamada tiver sucesso, o valor retornado é o tamanho do datagrama

Chamada close()



Chamada close()

Exemplo:

```
int sd; // socketdescriptor
...

status = close(sd);
if (status == -1)
    perror("Erro na chamada close");
...
```

Exercício



Exercício

(1) Implemente um cliente para o serviço UDP echo.

- ❖ O serviço UDP echo responde exatamente com a sequência ASCII enviada.

Exercício

(2) Implemente um cliente para o serviço daytime UDP.

- ❖ O serviço daytime UDP responde com a data e hora do servidor no instante de recebimento do datagrama UDP.

Referências Bibliográficas



Referências Bibliográficas

- ❑ **COMMER, DOUGLAS; STEVENS, DAVID**
 - ❖ Internetworking with TCP/IP: volume 3: client-server programming and applications
 - ❖ Prentice Hall
 - ❖ 1993