


© 2002-2009 Volnys Bernal 1

Cliente TCP

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




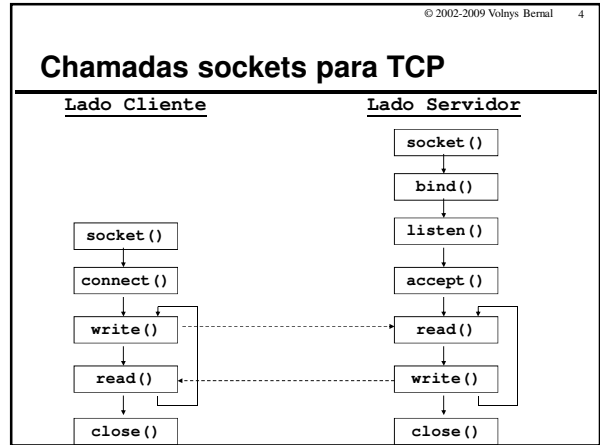
© 2002-2009 Volnys Bernal 2

Agenda

- ❑ Resumo das chamadas sockets para TCP
- ❑ Chamada socket()
- ❑ Chamada connect()
- ❑ Chamada read()
- ❑ Chamada write()
- ❑ Chamada close()


© 2002-2009 Volnys Bernal 3

Resumo das chamadas sockets para TCP

© 2002-2009 Volnys Bernal 5

Chamada socket()



© 2002-2009 Volnys Bernal 6

Chamada socket()

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

```
int socket (int domain, int type, int protocol)
```
- ❑ **Observação:**
 - ❖ Quando um socket é criado ele não possui nenhuma informação armazenada (endereços IPs e portas).
 - ❖ Endereços IPs e portas são informados nas chamadas bind() (lado servidor) e connect() (lado cliente).

© 2002-2009 Volnys Bernal 7

Chamada socket()

□ Sintaxe

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

Socket descriptor

Id. do protocolo

- UDP (17)
- TCP (6)

Família de protocolos

- PF_INET (IPv4)
- PF_INET6 (IPv6)
- PF_X25 (X25)

Tipo da comunicação

- SOCK_STREAM (TCP)
- SOCK_DGRAM (UDP, TCP)
- SOCK_RAW (IP)

© 2002-2009 Volnys Bernal 8

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)

© 2002-2009 Volnys Bernal 9

Chamada socket()

□ Exemplo de criação de socket TCP

```
#include <sys/socket.h>
int sd;
...
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().

© 2002-2009 Volnys Bernal 10

Chamada connect()



© 2002-2009 Volnys Bernal 11

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 procedimento de estabelecimento de conexão TCP

© 2002-2009 Volnys Bernal 12

Chamada connect()

□ Sintaxe

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

Socket descriptor

Socket address (IP + porta) do servidor

Tamanho da estrutura de endereço (sockaddr_in)

© 2002-2009 Volnys Bernal 13

Chamada connect()

```
#include <netdb.h>


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

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

// ativa connect
status = connect( socketdescriptor,
                 (struct sockaddr *)&serveraddr,
                 sizeof(serveraddr) );
if (status != 0)
    perror("Erro na chamada connect");
```

© 2002-2009 Volnys Bernal 14

Chamada write()



© 2002-2009 Volnys Bernal 15

Chamada write()

- ❑ **Objetivo**
 - ❖ Transmissão/escrita de dados em um descritor
 - Descritor: descritor sockets, descritor de arquivo, ...
 - ❖ Pode ser utilizada por cliente ou servidor
- ❑ **Valor retornado pela função**
 - ❖ Positivo: quantidade de bytes escritos
 - ❖ -1: erro

© 2002-2009 Volnys Bernal 16

Chamada write()

- ❑ **Sintaxe**

```
#include <unistd.h>

int write(int sd, void *buf, int count)
```

Socket
Descriptor

Tamanho da
mensagem

Ponteiro para mensagem
(end. do buffer da mensagem)

© 2002-2009 Volnys Bernal 17


Chamada write()

- ❑ **Exemplo:**

```
...
status = write(sd, txbuffer, strlen(txbuffer)+1)
if (status == -1)
    perror("Erro na chamada write");
...
```

© 2002-2009 Volnys Bernal 18

Chamada read()



© 2002-2009 Volnys Bernal 19

Chamada read()

- **Objetivo**
 - ❖ Recepção/leitura de dados de um descritor
 - Descritor: descritor sockets, descritor de arquivo, ...
 - ❖ Pode ser utilizada por cliente ou servidor
- **Valor retornado pela função**
 - ❖ >0: quantidade de bytes lidos
 - ❖ 0: end of file
 - ❖ -1: erro

© 2002-2009 Volnys Bernal 20

Chamada read()

□ **Sintaxe:**

```
#include <unistd.h>

int read(int sd, void *buf, int buffersize)
```

Socket
Descriptor

Tamanho do
buffer

Ponteiro para o buffer
(end. do buffer de recepção)

© 2002-2009 Volnys Bernal 21

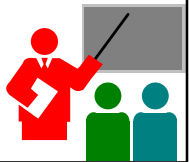
Chamada read()

□ **Exemplo:**

```
...
status = read(sd, bufferp, buffersize)
if (status == -1)
    perror("Erro na chamada read");
...
```

© 2002-2009 Volnys Bernal 22

Chamada close()



© 2002-2009 Volnys Bernal 23

Chamada close()


□ **Exemplo:**

```
int sd; // socket descriptor
...

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

© 2002-2009 Volnys Bernal 24

Exercício



© 2002-2009 Volnys Bernal 25


Exercício

(1) **Implemente um cliente para o serviço “echo” utilizando o protocolo TCP.**

❑ **Obs: O serviço TCP echo responde exatamente com a seqüência ASCII enviada.**

© 2002-2009 Volnys Bernal 26

Referências Bibliográficas



© 2002-2009 Volnys Bernal 27

Referências Bibliográficas

❑ **COMMER, DOUGLAS; STEVENS, DAVID**

- ❖ Internetworking with TCP/IP: volume 3: client-server programming and applications
- ❖ Prentice Hall
- ❖ 1993