

Publicidade



ASSINE 0800 703 3000

BATE-PAPO

E-MAIL

SAC

Voip

E-Mail Grátis

Shopping

ÍNDICE PRINCIPAL

PROCURAR:

no site

OK

Terça, 24/02/2009

- » Introdução
- » Programação
- » Administração
- » Hardware
- » Aplicativos
- » Jogos
- » Segurança
- » Editorial
- » Entrevistas

ARTIGOS

- » Fórum
- » Links
- » Notícias
- » Pegue o Linux
- » Documentação

OLINUX SERVIÇOS COMUNIDADE

- » Programas
- » Dúvidas
- » Oportunidades
- » Sobre
- » Contato
- » Publicidade

Powered By:
DEBIAN
GNU/LINUX

English Version

Linux Solutions
Shopping
OLinux



Programação

Tutorial de sockets - Parte IV

Por: [Frederico Perim](#)

► close() e shutdown() - Saia daqui!

Nossa! Você enviou e recebeu [dados](#) o dia inteiro, e já está não aguenta mais. Você está pronto para fechar a conexão em um descritor socket. Isso é fácil. Você pode usar a função close() do Unix:

```
close(sockfd);
```

Isso irá evitar escrever ou ler através deste socket. Se alguém tentar se conectar, receberá uma mensagem de erro.

Caso você queira ter mais controle sobre como o socket fecha, você pode usar a função shutdown(). Ela te permite encerrar a conexão em uma determinada direção, ou em ambas (como close()):

```
int shutdown(int sockfd, int how);
```

sockfd é o socket que você deseja encerrar, e how pode ser o seguinte:

- 0 - encerra o recebimento;
- 1 - encerra o envio;
- 2 - encerra o recebimento e envio (como close()).

shutdown() retorna 0 em caso de sucesso, e -1 caso contrário.

Se você insistir em usar shutdown() em sockets de datagrama, ele vai fazer com que socket fique indisponível para futuras chamadas recv() e send() (lembre-se de que você pode usá-las caso use connect() em seus "SOCK_DGRAM").

É importante ressaltar que shutdown() não fecha o descritor de [arquivo](#), apenas muda sua usabilidade. Para liberar um descritor socket, você precisa usar close().

► getpeername() - Quem é você?

Essa função é fácil.

É tão fácil, que quase não dedico uma seção. Mas enfim, aqui está.

A função getpeername() irá dizer quem está conectado na outra extremidade de um stream socket. A sinopse:

```
int getpeername(int sockfd, struct sockaddr *addr, int *addrlen);
```

sockfd é o descritor do socket conectado, addr é um ponteiro para um struct sockaddr (ou uma struct sockaddr_in) que vai ter informações sobre o outro lado da conexão, e addrlen é um ponteiro para um int, que deve ser inicializado como sizeof(struct sockaddr).

Uma vez que você tenha o endereço, você pode usar inet_ntoa() ou gethostbyaddr() para imprimir ou obter mais [informação](#). Não, você não pode ler o nome de login. (Tá

ENQUETE

Com qual frequência você acessa o site Olinux?

- ☐ Todos os dias
- ☐ Uma vez por semana
- ☐ Cinco vezes aos mês
- ☐ Poucas vezes ao mês
- ☐ Outra

VOTAR

NEWSLETTER

Inscreva-se e receba as últimas notícias, programas, artigos, novidades e tudo do mundo Linux que aconteceu na semana.

Digite seu email:

OK

Relógio

de Pulso em até
12x.

**Auto DVD
Player**

Diversas marcas e
modelos. Encontre
em até 12x sem
juros.

Vinhos

Diversas marcas a
partir de R\$ 9,90!
Aproveite!

Nintendo DS

Mais diversão pelo
menor preço.

[COMPARE PREÇOS](#)

bom, Se outro computador estiver rodando um ident daemon, isso é possível.Mas isso está além do escopo deste documento).

[Próximo»](#)

- ▶ [close\(\) e shutdown\(\) - Saia daqui!](#)
- ▶ [getpeername\(\) - Quem é você?](#)
- ▶ [gethostname\(\) - Quem sou eu?](#)
- ▶ [DNS - Você diz "receita.fazenda.gov.br", eu digo "161.148.231.100"](#)



Enviar para um
amigo



Imprimir



Índice de
artigos