

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

COMUNIDADE

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

SERVIÇOS

Powered By:
DEBIAN
GNU/LINUX

English Version

Linux Solutions

Shopping
OLinux



Programação

Tutorial de Sockets - Parte III

Por: [Frederico Perim](#)

▶ listen() - Tem alguém me chamando?

Hora de mudar um pouco o rumo da prosa. E se você não quiser se conectar a um host. Digamos que você queira esperar por conexões remotas e tratá-las de alguma forma. É um processo de dois passos: primeiro você escuta(listen()), e então aceita (accept()) a conexão.

A chamada listen() é muito simples, mas requer um pouco de explicação:

```
int listen( int sockfd, int backlog);
```

sockfd é o já conhecido descritor de [socket](#) da chamada socket(). backlog é o número de conexões permitidas. O que isso significa? Bem, as conexões ficarão na fila (queue) até que você as aceite(accept()) e isto é o limite de quantas conexões poderão ficar na fila. A maioria dos sistemas limita este número para cerca de 20, mas você pode tranquilamente usar 5 ou 10.

Novamente, como sempre, listen(0 retorna -1 em caso de erro e seta errno.

Bem, como você deve ter imaginado, precisamos chamar bind() antes de listen() ou o kernel irá escutar em uma porta remota. Então se você for escutar por conexões remotas, a sequência de chamadas de [sistema](#) que você vai executar é:

```
socket();  
bind();  
listen();  
/* accept() vem aqui */
```

Eu vou deixar assim ao invés de código fonte, já que é bastante elucidativo.(O código na seção accept() abaixo. é mais completo.)A parte mais importante da coisa é a chamada accept().

▶ 4.5. accept() - "Obrigado por chamar a porta 3490."

Prepare-se - a chamada accept() é um tanto estranha! O que vai acontecer é o seguinte: alguém em algum lugar bem distante vai tentar se conectar(connect()) a sua máquina na porta em que você estiver ouvindo(listen()). Está conexão vai ficar na fila (queue) esperando ser aceita(accept()). Você chama accept() e diz para retornar a conexão pendente. A seguir você vai ter um descritor socket novinho para usar nessa conexão. Você, de uma hora pra outra, tem duas conexões pelo preço de uma! A original ainda está ouvindo na porta especificada e a outra está pronta para enviar(send()) e receber(recv()).

A chamada é da seguinte forma:

```
int accept (int sockfd, void *addr, int *addrlen);
```

sockfd, bem não vou ser repetitivo. addr será geralmente um ponteiro para um struct sockaddr_in. Aqui é onde a [informação](#) sobre a conexão vai ficar(e com isso você pode determinar qual host está chamando de qual porta.) addrlen é um variável inteira local que deve ser setada para sizeof(struct sockaddr_in) antes que seu endereço seja passado para accept(). accept() não colocará mais bytes que isto em addr. Se colocar

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

Jaguar

Vários modelos.
Entre e confira.

**Auto DVD
Player**

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

Filmadora

Multilaser CR-518
Digital.
Compare!

Nintendo DS

Mais diversão pelo
menor preço.

[COMPARE PREÇOS](#)

menos, irá mudar o valor de addrlen para refletir essa mudança.

Adivinha? accept() retorna -1 e o resto você já deve saber.

[Próximo»](#)

▶ **listen() - Tem alguém me chamando?**
▶ **4.5. accept() - "Obrigado por chamar a porta 3490."**
▶ **send() e recv() - Fale comigo!**
▶ **sendto() e recvfrom() -- Fale comigo no estilo DGRAM**



Enviar para um
amigo



Imprimir



Índice de
artigos

[Publicidade](#) / [Sobre OLinux](#) / [Entre em Contato](#) / [Privacidade](#)
Copyright (c) 2000-2007, OLinux - O Portal de Linux do Brasil.
Desenvolvido por: [Linux Solutions](#)
Todos os Direitos Reservados.