

Publicidade

6x de
R\$38,16
no cartão **FAST****6x de**
R\$38,16
no cartão **FAST**

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 I

Por: [Frederico Perim](#)

► Estruturas e Manipulação de Dados

Bem finalmente é hora de falar sobre programação. Nesta parte, vou cobrir alguns tipos de dados usados pelas interfaces de sockets, já que algumas são um pouco complicadas.

Primeiro a mais fácil: um descritor de socket. O descritor de socket é do seguinte tipo:

int

Apenas um simples int.

As coisas começam a se complicar daqui em diante, então vamos lá. Saiba isto: existem dois tipos de ordenamentos de byte: primeiro byte mais significativo (também chamados de "octet"), e primeiro byte menos significativo, chamado de "Network Byte Order". Algumas máquinas armazenam números internamente em Network Byte Order, algumas não. Quando digo que algo deve estar em "Network Byte Order", você tem que chamar uma função (`htons()`) para mudar de "Host Byte Order". Se não for dito "Network Byte Order", então você pode deixar como "Host Byte Order".

A primeira estrutura contém informações sobre tipos de sockets:

```
struct sockaddr {  
    unsigned short sa_family; // tipo de endereço, AF_XXX  
  
    char sa_data[14]; // 14 bytes do endereço de protocolo  
};
```

`sa_family` pode ser uma variedade de coisas, mas será `AF_INET` para tudo que fizermos neste documento. `sa_data` contém um endereço de destino e número de porta para o socket. Isto é um tanto confuso, já que não queremos empacotar o endereço `sa_data` manualmente.

Para tratar de `struct sockaddr`, programadores criaram uma estrutura paralela: `struct sockaddr_in` para "[internet](#)".

```
struct sockaddr_in {  
  
    short int sin_family; // tipo de endereço  
  
    unsigned short int sin_port; // Número da porta  
  
    struct in_addr sin_addr; // Endereço da Internet  
  
    unsigned char sin_zero[8]; // Mesmo tamanho de struct sockaddr  
};
```

Esta estrutura torna mais fácil referenciar elementos de um endereço de socket. Note que `sin_zero` (que é incluído para acomodar a estrutura ao tamanho de `struct sockaddr`) deve ser setada para zeros através da função `memset()`. Ainda, e isto é muito importante, um ponteiro para um `struct sockaddr_in` pode ser convertido para um `struct`

ENQUETE

Com qual frequência você
acessa o site Olinux?

- ☐ Todos os dias
- ☐ Uma vez por semana
- ☐ Cinco vezes ao 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.

Filmadora

Multilaser CR-518 Digital. Compare!

Esteira

Entre em forma antes do verão.

[COMPARE PREÇOS](#)

sockaddr e vice-versa. Assim mesmo que socket() queira um struct sockaddr*, você pode ainda usar um struct sockaddr_in e converter no momento apropriado. Note também que sin_family corresponde a sa_family em structt sockaddr e deve ser setado para "AF_INET". Finalmente ,sin_port e sin_addr devem estar em "Network Byte Order".

"Mas um momento! Como pode toda estrutura, struct in_addr sin_addr, estar em Network Byte Order?" Esta questão requer uma análise cuidadosa da estrutura struct in_addr :

```
// Endereço de Internet
```

```
struct in_addr {  
    unsigned long s_addr;  
};
```

Então se você declarou ina para ser do tipo struct sockaddr_in, então ina.sin_addr.s_addr referencia um endereço IP de 4 bytes (em "Network Byte Order").

[<< Anterior](#)

[Próximo >>](#)

- [O que é um Socket?](#)
- [Dois Tipos de Sockets](#)
- [Alguma Teoria de Rede](#)
- [Estruturas e Manipulação de Dados](#)
- [Convertendo Valores](#)
- [Endereços IP e como Lidar com eles](#)
- [Conclusão](#)



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.