

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

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

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

Powered By:
DEBIAN
GNU/LINUX

English Version



Programação

Tutorial de Sockets - Parte I

Por: [Frederico Perim](#)

► Convertendo Valores

Houve muito falatório sobre conversão para Network Byte Order. Agora é hora de ação!

Tudo bem. Existem dois tipos que você pode converter: um short de Host Byte Order para Network Byte Order. Comece com "h" para "host", seguindo com "to", e então "n" para "network", e "s" para "short": h-to-n-s ou htons() (leia-se: "Host to Network Short").

Nem é tão difícil assim...

Você pode usar combinações de "n", "h", "s" e "l" que quiser, sem contar as estúpidas. Por exemplo, não existe uma função stolh() ("Short to Long Host") - pelo menos não aqui. Mas existem:

- htons() -- "Host to Network Short"
- htonl() -- "Host to Network Long"
- ntohs() -- "Network to Host Short"
- ntohl() -- "Network to Host Long"

Agora, você deve estar começando a entender. Você pode pensar, "O que eu faço para mudar a ordem de byte em char?" Então você pensa, "Deixa pra lá". Você também deve imaginar que sua máquina 68000 já usa "Network Byte Order", e não precisa chamar htonl() no seus endereços IP. Você estaria certo, MAS se você tentar portar para uma máquina que tem ordem de byte de rede reverso seu programa não vai rodar. Tente escrever programas portáveis a outras arquiteturas! Este é um mundo Unix ([Linux](#)). Lembre-se: pode seus bytes em Network Byte Order antes de botá-los na rede.

Uma última esclarecimento: porque sin_addr e sin_port precisam estar em Network Byte Order em um struct sockaddr_in, mas sin_family não? A resposta: sin_addr e sin_port são encapsulados no pacote nas layers IP e UDP, respectivamente. Assim, eles devem estar em Network Byte Order. No entanto, o campo sin_family não é enviado pela rede, ele pode estar em Host Byte Order.

► Endereços IP e como Lidar com eles

Felizmente para você, existem várias funções que te permitem manipular endereços IP. Não é necessário tratá-los manualmente e colocá-los em um long com um operador <<.

Primeiro, digamos que você tem um endereço IP "10.12.110.57" e deseja armazená-lo em um struct sockaddr_in ina. A função que você deseja usar, inet_addr(), converte um endereço IP na notação de números e pontos em um unsigned long. A forma é a seguinte:

```
ina.sin_addr.s_addr = inet_addr("10.12.110.57");
```

Note que inet_addr() retorna o endereço em Network Byte Order, você não precisa chamar htonl(). Beleza!!!

Agora, o código acima não é muito robusto porque não existe checagem de erro. inet_addr() retorna -1 em caso de erro. Lebra de números binários? (unsigned) -1 corresponde ao endereço IP 255.255.255.255! Este é o endereço de broadcast!. Lembre-se de checar erros apropriadamente.

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.

Vinhos

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

Esteira

Entre em forma
antes do verão.

[COMPARE PREÇOS](#)

Na verdade , há uma forma interface melhor que `inet_addr()`: é chamada `inet_aton()` ("aton" significa "ascii to network");

```
int inet_aton(const char *cp , struct in_addr *inp);
```

E aqui está um exemplo prático: este exemplo fará mais sentido quando abordamos `bind()` e `connect()`!!

```
struct sockaddr_in meu_end;
```

```
meu_end.sin_family = AF_INET; //host byte order
```

```
meu_end.sin_port = htons(MYPORT); //short, network byte order
```

```
inet_aton("10.12.110.57", &(meu_end.sin_addr));
```

```
memset(&(meu_end.sin_zero), ' \0' , 8); // zera o resto da estrutura
```

`inet_aton`, diferente de praticamente todas as outras funções relacionadas a sockets, retorna um número diferente de zero em caso de sucesso, e zero caso contrário (Se alguém souber porque , me diga) E o endereço é passado de volta em `inp`.

Infelizmente , nem todas plataformas implementam `inet_aton()` assim , embora seu uso seja recomendado, a função `inet_addr()` será usada neste artigo.

Tudo bem, agora você pode converter um string de endereço IP em suas representações binárias. E a forma inversa? E se você tem um struct `in_addr` e quer mostrar na notação de pontos e números? Neste caso , você terá que usar a função `inet_ntoa()` ("ntoa" significa "network to ascii") da seguinte forma:

```
printf ("%s", inet_ntoa(ina.sin_addr));
```

Isto irá mostrar o endereço IP. Note que `inet_ntoa()` tem um struct `in_addr` como argumento, não um long. Note também que ela retorna um ponteiro para um char. Isto aponta para um array estaticamente armazenado em `inet_ntoa()` assim toda vez que você chamar `inet_ntoa()` ela irá sobrescrever o último endereço IP que você chamou. Por exemplo:

```
char *a1, *a2;
//
//
a1 = inet_ntoa(ina1.sin_addr); isto é 192.168.4.14
a2 = inet_ntoa(ina2.sin_addr); isto é 10.14.110.60
printf ("endereço 1: %s\n", a1);
printf("endereço 2: %s\n", a2);
a saída é:
endereço 1: 10.14.110.60
endereço 2: 10.14.110.60
```

Se você precisar salvar o endereço, use `strcpy()` em seu próprio array de caracteres.

➤ Conclusão

No próximo artigo desta série nos aprofundaremos nas chamadas de sistemas e portas desconhecidas, como o caso do Bind. Até lá!

Arquivo original em: <http://www.ecst.suchico.edu/~beej/guide/net/>

[<< Anterior](#)

➤ 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



Copyright (c) 2000-2007, OLinux - O Portal de Linux do Brasil.

Desenvolvido por: [Linux Solutions](#)

Todos os Direitos Reservados.