

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 IV

Por: [Frederico Perim](#)

▶ gethostname() - Quem sou eu?

Ainda mais fácil que getpeername() é a função gethostname(). Ela retorna o nome do computador onde o programa está rodando. O nome pode então ser usado por gethostbyname(), abaixo, para determinar o endereço IP da sua máquina local.

O que poderia ser mais divertido? Aqui está o protótipo:

```
int gethostname (char *hostname, size_t size);
```

Os argumentos são simples: hostname é um ponteiro para um array de caracteres que contém o hostname logo após o retorno da função, e size é o tamanho em bytes do array hostname.

A função retorna 0 em caso de sucesso, e -1 caso contrário como sempre.

▶ DNS - Você diz "receita.fazenda.gov.br", eu digo "161.148.231.100"

Caso você não saiba o que significa DNS, simples: "Domain Name Service", ou Serviço de Nome de Domínio. Rapidamente, você diz o endereço no formato que seja legível ao humanos, e o DNS retorna o endereço IP (assim você pode usar bind(), connect(), sendto(), ou o que precisar.) Dessa forma, quando alguém digita:

```
% telnet receita.fazenda.gov.br
```

telnet pode descobrir que precisa se conectar a "161.148.231.100".

Mas como funciona? Você vai usar a função gethostbyname():

```
struct hostent *gethostbyname(const char *name);
```

Como você pode ver, a função retorna um ponteiro para um struct hostent, que tem o seguinte layout:

```
struct hostent {  
    char *h_name;  
    char **h_aliases;  
    int h_addrtype;  
    int h_length;  
    char **h_addr_list;  
};
```

E aqui está a descrição dos campos na struct hostent:

- h_name -- Nome oficial do host

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.

Brinquedos

das Meninas Super
Poderosas. Clique!

Vinhos

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

Esteira

Entre em forma
antes do verão.

COMPARE PREÇOS

Buscar

- `h_aliases` -- Um array terminado em nulo de nomes alternativos do host
- `h_addrtype` -- O tipo de endereço retornado, geralmente `AF_INET`
- `h_length` -- O tamanho do endereço em bytes
- `h_addr_list` -- Um array terminado em zero de endereços de [rede](#) de um host. Endereços de rede estão em "Network Byte Order"
- `*h_addr_list[0]` -- O primeiro endereço em `h_addr_list`.

`gethostbyname()` retorna um ponteiro para um struct `hostent` preenchido, ou nulo em caso de erro. (Mas `errno` não é setada, e sim `h_errno`. Veja `h_errno()`, abaixo).

Mas como é usada? Algumas vezes (ao ler manuais), apenas "jogar" a [informação](#) no leitor não é suficiente. A função é mais fácil de usar do que parece.

```
/*
** getip -- um demo que procura nomes de host
*/

int main(int argc, char *argv[])
{
    struct hostent *h;

    if (argc != 2) { //checa a linha de comando
        fprintf(stderr, "Uso: getip endereço\n");
        exit(1);
    }

    if ((h=gethostbyname(argv[1])) == NULL) { // pega
informação do host

        perror("gethostbyname");
        exit(1);
    }

    printf ("Nome do Host : %s\n", h->h_name);
    printf ("Endereço IP   : %s\n", inet_ntoa(*(struct
in_addr *)h->h_addr_list[0] ));

    return 0;
}
```

Com `gethostbyname()`, você não pode usar `perror()` para imprimir a mensagem de erro. Ao invés, chame `h_errno()`.

É muito simples. Você passa o string que contém o nome da máquina ("receita.fazenda.gov.br") para `gethostbyname()`, e então pega a informação de um struct `hostent` retornado.

A única coisa estranha deve ser na hora de imprimir o endereço IP, acima. `h->h_addr_list[0]` é um `char*`, mas `inet_ntoa()` precisa receber um struct `in_addr`. Então eu converto `h->h_addr_list[0]` para um struct `in_addr*`, e depois retiro a referência para chegar aos [dados](#).

[< Anterior](#)

[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

[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.