

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

OLINUX

Powered By:
DEBIAN
GNU/LINUX

English Version

Linux Solutions

Shopping
OLinux



Programação

Tutorial de sockets - Parte VI

Por: [Frederico Perim](#)

Finalmente, o que é a estranha struct timeval? Bem, algumas vezes você não quer esperar para sempre alguém enviar dados. Talvez a cada 96 segundos você deseja mostrar "Ainda esperando..." no terminal embora nada tenha acontecido. Esta estrutura lhe permite especificar um período de tempo. Se esse período se excedeu e select() ainda não encontrou nenhum descritor de arquivo, select() retornará para que você continue processando.

struct timeval tem os seguintes campos:

```
struct timeval {
    int tv_sec; // segundos
    int tv_usec; //microsegundos
```

Apenas atribua a tv_sec o número de segundos para esperar, e a tv_usec o número de microsegundos para esperar. Sim, isto é microsegundos, não milissegundos. Existem 1000 microsegundos em um milissegundo, e 1000 milissegundos em um segundo. Assim, existem 1000000 de microsegundos em um segundo. Porque "usec"? O u deve parecer com a letra grega μ que nós usamos para "micro". Ainda, quando a função retorna, timeout deve ser atualizado para mostrar o tempo restante. Isto depende da sua plataforma Unix.

Uau! Nós temos um timer em microsegundos! Bem, não conte com isso. O padrão timeslice do Unix é por volta de 100 milissegundos, por isso você deve esperar esse tempo independentemente do valor que você atribuir a struct timeval.

Outras coisas de interesse. Se você setar os campos em sua struct timeval para 0, select() irá encerrar imediatamente, consultando todos os descritores em seu conjunto. Se você atribuir NULL a timeout, então select() nunca irá encerrar e vai esperar até que o primeiro descritor esteja pronto. Finalmente, se você não se importar em esperar por algum conjunto, pode atribuir NULL na chamada select().

O código a seguir espera 2.5 segundos para alguma coisa aparecer na entrada padrão:

```
// descritor de arquivo para entrada
padrão

int main(void)
{
    struct timeval tv;
    fd_set readfds;
    tv.tv_sec = 2;
    tv.tv_usec = 500000;
    FD_ZERO(&readfds);
    FD_SET(0, &readfds);
    //não se importa com writefds e exceptfds:
    select(0 + 1, &readfds, NULL, NULL, &tv);
    if (FD_ISSET(0, &readfds))
        printf("uma tecla foi pressionada!\n");
```

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](#)

```
else
    printf("Tempo esgotado.\n");
    return 0;
}
```

Alguns de vocês devem estar pensando que esta é uma ótima forma de esperar por dados em um socket UDP (SOCK_DGRAM) - e você está certo, pode ser. Alguns versões Unix podem usar select() dessa forma, outras não. Você deve consultar suas páginas man quando tentar fazer isso.

Algumas versões Unix atualizam o tempo em sua struct timeval para refletir a quantidade de tempo que ainda falta antes de se esgotar. Mas outras não. Não confie nisso se quiser que seu programa seja portátil. (Use a função gettimeofday() se você precisar saber o tempo decorrido. É um saco, eu sei, mas tem que ser desse jeito.

Mais uma nota a respeito de select() : se você tiver um socket escutando (listen()), você pode checar se há uma nova conexão colocando o descritor socket desejado em um conjunto readfds.

E esta é, meus caros amigos, uma rápida análise da poderosa função select().

Mas, devido a pedidos, aqui esta um exemplo mais aprofundado. Infelizmente, a diferença entre o exemplo mais simples acima e este, é bastante significante. Mas dê uma olhada , e leia a descrição que a vem a seguir.

[<<Anterior](#)

[Próximo>>](#)

► **Técnicas Avançadas**

► **Blocking**

► **select()**



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.