

O slackware é a distribuição linux mais antiga ainda em atividade. Tendo sido criada por Patrick Volkerding em 1993, a partir da SLS.

Em todos esses anos, a distro conquistou ardorosos utilizadores, principalmente graças à sua filosofia de simplicidade e estabilidade.

Um produto de extrema qualidade para usuários com esta mesma característica. E este zine é de slacker para slacker.



# slackware zine

Slackware is a **registered trademark** of Slackware Linux, Inc.

15 de Junho de 2005 - Edição #9

## Editorial

Um atraso monstruoso. Conseguimos uma proeza, lançamos a edição #9.5 antes da #9! Agora, porquê a #9 atrasou?

O primeiro motivo foi um pouco de desorganização da nossa parte. Nós, editores da zine, nos responsabilizamos por isso. Falta de planejamento e uma má distribuição das tarefas foram os principais problemas. E, esperamos que tenham sido resolvidos.

O segundo motivo (relacionado com o primeiro) foi o FISL. Estivemos lá presentes e foi muito gratificante ver o retorno dos leitores e, claro, tivemos que fazer uma edição especial para o evento. O que foi "em cima" dessa edição.

Agora, quem não teve culpa pelo que aconteceu foram os nossos colaboradores! Todos estiveram de parabéns e nos enviaram ótimos artigos, tanto para a edição especial, como para essa. Esperamos que as colaborações e o retorno da comunidade continue melhorando cada vez e que nós possamos atender a essas expectativas.

Essa edição deve agradar todos os tipos de usuário, tem artigos de programação, artigos para desktop e vários artigos para administradores de sistemas (que estavam reclamando do teor da revista e, pelo jeito, responderam do jeito certo, mandando artigos). No mais, divirtam-se com essa edição e aguardem a #10, para julho.

Piter PUNK

## Índice

### Iluminação Virtual em OpenGL

Diego Fiori de Carvalho

2

### Instalando o Skype 1.1.03 no slackware

Clayton dos Santos

5

### Controlando Banda com o HTB

Wanderson Santiago dos Reis

6

### Integrando Linux e Netware

Flavio do Carmo Jr.

8

### Mais segurança com o AIDE

Leandro Toledo

10

### Autenticação do Squid via Samba

Frederico R. C. Costa

13

### Backups Usando Fitas DAT

Clayton dos Santos

16

# slackwarezine – antes tarde do que nunca.

Reprodução do material contido nesta revista é permitida desde que se incluam os créditos aos autores e a frase:

**"Reproduzida da Slackware Zine #9 -  
www.slackwarezine.com.br"**

com fonte igual ou maior à do corpo do texto e em local visível



**slack  
users**

# Iluminação Virtual em OpenGL

## Introdução

Para este artigo ficar mais didático, procurei dividi-lo em 2 partes principais: Divagando sobre luz e Manipulação de luzes.

A primeira parte ilustra as propriedades físicas da luz, e a segunda comenta sobre a criação, seleção e posicionamento de luzes. No tópico manipulação de luzes, encontra-se a explicação das três principais características de iluminação: cor, atenuação e posicionamento. Estas propriedades são ilustradas com exemplos de código e figuras.

Mesmo tendo um cuidado especial com o didatismo, é importante que o leitor possua conhecimentos básicos de OpenGL, inclusive para compreender os códigos apresentados.

## Divagando sobre Luzes

Fisicamente, a luz é composta de fótons, partículas minúsculas de luz que são determinados por sua posição, sentido e frequência.

A luz é uma onda eletromagnética que possui uma grande faixa de comprimentos de onda. O olho humano percebe apenas uma faixa estreita desse espectro. A luz visível pelo ser humano, pode variar de 390 nanômetros (nm), para a cor violeta, até 720 nm para a cor vermelha. Na realidade o que seu olho vê é uma mistura dos fótons de ondas diferentes que geram as cores do mundo real.

A luz branca ideal consiste em uma quantidade exatamente igual de luz de todas as faixas. Nosso olho percebe a cor quando as nossas células especiais, chamadas cones, presentes na retina excitam-se após serem golpeados pelos diversos fótons.

Existem 3 tipos de cones, os que são mais sensíveis a cor vermelha, outros a cor verde e a azul. A partir deste princípio de sensibilização dos cones, que foi inspirado a construção dos monitores gráficos, os quais emulam as cores visíveis com a iluminação dos pixels nos vários níveis de intensidade das três componentes de cor(RGB).

Do mesmo modo o OpenGL aproxima a luz, caracterizando-a pela quantidade de luz vermelha, verde e azul que emite, e o material das superfícies é determinada pela porcentagem de componentes vermelhos, verdes e azuis que incidem sobre o objeto e são refletidos.

Mas não se preocupe com todos estes conceitos físicos, pois a SGI procurou facilitar a vida do programador criando comandos que implementam as funções matemáticas necessárias para a iluminação de cenas 3D em superfícies 2D(monitor de vídeo). O objetivo maior de uma aplicação 3D é alcançar a perfeição de confundir-se com uma cena real, é desse modo que os criadores de animações 3D pensam.

## Criação, seleção e posicionamento de luzes

Basicamente, para interagirmos com iluminação 3D, necessitamos interagir com a função `glLightfv()`, a qual é utilizada para definir os parâmetros de uma luz. Em OpenGL, determinamos uma fonte de luz após setarmos vários parâmetros que refletem as grandezas físicas do mundo real.

Os parâmetros fundamentais para criação de uma luz em OpenGL são: a propriedade de cor, propriedades de posicionamento e propriedades de atenuação.

## Propriedade de Cor: Cores e Reflexão dos Materiais

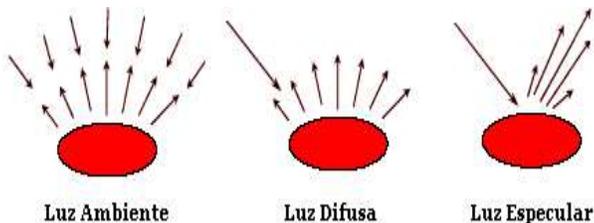
A manipulação das propriedades de cor permitem interações com as diferentes propriedades físicas dos materiais. A aproximação da cor de um material é realizada dependendo da quantidade percentual de luz de cada nível de cor RGB, ou seja, se tivermos um objeto inteiramente azul, este refletirá toda a luz azul que incide sobre ele e absorverá toda a luz vermelha e verde.

Dessa maneira, são absorvidos alguns fótons e outros são refletidos pela superfície, consequentemente as propriedades da superfície podem brilhar de diferentes maneiras. E esse efeito físico é simulado pelo OpenGL.

Existem 3 tipos de luzes(efeito "brilho") que podemos determinar seus valores para a cena criada:

- Luz Ambiente(GL\_ AMBIENT): Luz proveniente de todas as direções, devido a múltiplas reflexões da cena e refletida pela superfície do material em todas as direções.
- Luz Difusa(GL\_ DIFFUSE): Luz que incide no objeto de uma direção bem definida e reflete igualmente em todas as direções, a intensidade de luz refletida depende diretamente do ângulo de incidência da luz na superfície do material.
- Luz Especular(GL\_ SPECULAR): Luz que incide no objeto de uma direção bem definida e que tende a refletir-se pela normal, ou seja 90°. O melhor exemplo deste tipo de efeito é a reflexão de luz em superfícies metálicas (ou em um espelho).

Podemos visualizar esses tipos de luzes na figura abaixo:



### Propriedades de Posicionamento (GL\_POSITION):

Controlam a localização e o tipo da luz. OpenGL respeita o sistema de coordenadas homogêneas (x,y,z)[1], para definirmos a posição de uma luz, basta preenchermos este padrão de coordenadas e ainda definirmos um valor para w, a componente de posição de luz. A componente w pode receber dois valores distintos:

- 0 : para uma luz direcional que ilumina a partir do infinito.
- 1 : para uma luz localizada na cena.

Para entendermos melhor podemos observar o valor padrão do GL\_POSITION (0.0, 0.0, 1.0, 0.0) ilumina a partir da direção do eixo z no infinito (considerando que o eixo z avança do fundo do monitor para fora da tela). Podemos criar diversos efeitos de luz movimentando a câmera ou a própria luz pela cena.

Para criar uma fonte de luz que se move com o observador, indique a posição da luz antes de especificar a posição da câmera.

```
glPushMatrix();
glLightfv(GL_LIGHT0, GL_POSITION, \
          lightposition);
/* Colocar aqui os comandos de */
/* transformação de visão.      */

/* Colocar aqui os comandos para */
/* desenhar os objetos          */
glPopMatrix();
```

GL\_LIGHT0 define a fonte de luz que você está trabalhando. GL\_POSITION define que o parâmetro a ser modificado é o de posição lightposition é a posição no eixo x,y,z da fonte de luz. Você pode mover a câmera em x,y e z e a luz circula o objeto.

Para criar uma fonte de luz fixa, basta indicar a posição da luz e depois especificar a câmera.

```
glPushMatrix();
/* Colocar aqui os comandos de */
/* transformação de visão.      */

glLightfv(GL_LIGHT0, GL_POSITION, \
          lightposition);

/* Colocar aqui os comandos para */
/* desenhar os objetos          */
glPopMatrix();
```

No exemplo de luz móvel implementamos primeiramente a posição da câmera e depois as transformações de visão e finalmente o desenho da cena, nesse caso, podemos setar as condições da visão entre uma renderização e outra do desenho da cena, isto nos permite criar a sensação de movimento do foco de luz.

Já no segundo exemplo perdemos a sensação de movimentação, pois determinamos as transformações de visão antes de fixarmos o foco de luz.

### Propriedades de Atenuação

Esta propriedade define o nível de atenuação de uma fonte de luz de acordo com a distância da mesma em relação aos objetos da cena. Existem 3 tipos de atenuação de fontes de luz em OpenGL[2]:

- Constante(GL\_CONSTANT\_ATTENUATION): Não realiza atenuação, deixa a luz do objeto como estava anteriormente, ou seja o afastamento do objeto não influencia na iluminação.
- Linear(GL\_LINEAR\_ATTENUATION): A intensidade diminui linearmente com a distância, ou seja o afastamento do objeto provoca alterações na iluminação, quanto mais longe o objeto estiver menos iluminação terá.

- Quadrática(GL\_QUADRATIC\_ATTENUATION): A intensidade diminui quadraticamente com a distância, ou seja, o afastamento do objeto trará a sensação de distância percorrida, pois este desaparecerá aos poucos, quando estiver sendo posicionado em direção ao infinito.

Vamos agora observar um exemplo de aplicação com iluminação, para verificarmos os métodos apresentados neste artigo.

```
//Definição dos valores dos vetores para
//as componentes de luz.
GLfloat luz_ambiente[] = \
    { 0.0, 0.0, 0.0, 1.0 };
GLfloat luz_difusa[] = \
    { 1.0, 1.0, 1.0, 1.0 };
GLfloat luz_especular[] = \
    { 1.0, 1.0, 1.0, 1.0 };
GLfloat luz_posicao[] = \
    { 1.0, 1.0, 1.0, 0.0 };

//Essas Linhas definem as propriedades
//de cor para a cena, com a determinação
//para da fonte de luz GL_LIGHT0.
glLightfv(GL_LIGHT0, GL_AMBIENT,\
    luz_ambiente);
glLightfv(GL_LIGHT0, GL_DIFFUSE,\
    luz_difusa);
glLightfv(GL_LIGHT0, GL_SPECULAR,\
    luz_especular);

//Define a propriedade de posicionamento
//da fonte de luz GL_LIGHT0.
glLightfv(GL_LIGHT0, GL_POSITION,\
    luz_posicao);

//Define a propriedade de atenuação da
//fonte de luz GL_LIGHT0.
glLightfv(GL_LIGHT0,\
    GL_LINEAR_ATTENUATION, 0.1);

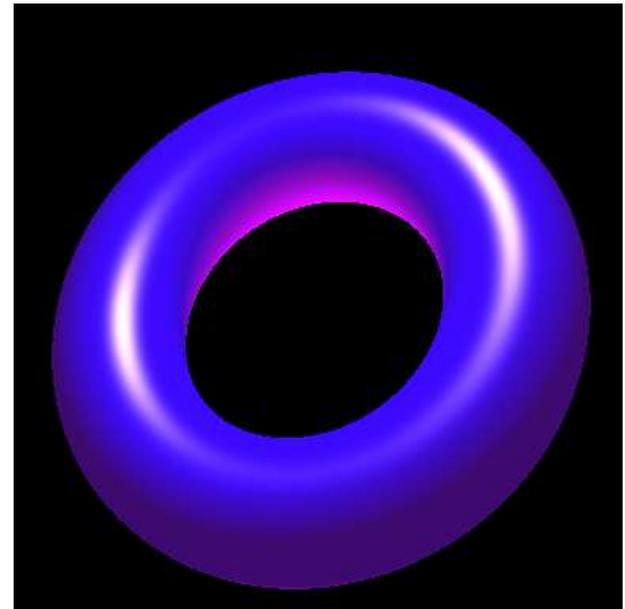
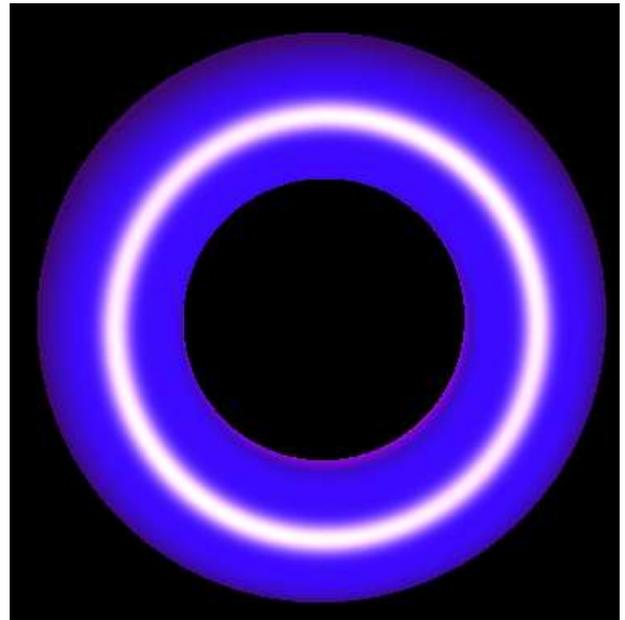
//Habilita o uso de iluminação
glEnable(GL_LIGHTING);

//Habilita cálculo de remoção de
//superfícies ocultas
glEnable(GL_DEPTH_TEST);

//Habilita a fonte de luz de número 0. Em
//Opengl podemos definir até 8 focos
//(GL_LIGHT0 até GL_LIGHT7).
glEnable(GL_LIGHT0);
```

#### Referências:

- [1] "Programação 3D com OpenGL" na edição nº 8 da slackwarezine.
- [2] Versão eletrônica do RedBook Opengl.  
[http://www.opengl.org/documentation/red\\_book\\_1.0/](http://www.opengl.org/documentation/red_book_1.0/)



# Instalando o Skype 1.1.03 no slackware

Recentemente, realizei o setup de um novo desktop aqui onde trabalho e coloquei o bom e velho **slackware** 10.0 para rodar nele. Depois de instalado e devidamente "patchado", comecei a instalar os aplicativos que mais utilizo no meu cotidiano e um deles é o skype.

Pra quem não conhece, o skype é uma "suíte de comunicação completa", com funções que vão desde a troca de mensagem instantâneas, até a utilização da tecnologia de Voip (voz sobre ip), possibilitando conversas entre usuários de todo o mundo a custo zero e com excelente qualidade de som, suportando inclusive conferências entre múltiplos usuários simultaneamente.

Bem, como estava dizendo, fui até o site do skype e baixei a última (e única) versão disponível para download, a versão 1.1.03 (linux), que pode ser encontrada em:

```
http://www.skype.com/go/\
  getskype-linux-dynamic
http://www.skype.com/go/\
  getskype-linux-static
```

O processo de instalação do skype é trivial:

```
su -
cd /usr/local
cp ~clayton/\
  skype_staticQT-1.1.0.3.tar.bz2 .
tar -xvzf skype_staticQT-1.1.0.3.tar.bz2
mv skype_staticQT-1.1.0.3.tar.bz2 skype
cd skype
exit
```

No momento da execução é que ocorre o grande problema... :(

```
$ ./skype
./skype: error while loading shared
libraries: libdbus-1.so.0: \
  cannot open shared object file: \
  No such file or directory

$ ldd skype
libdbus-1.so.0 => not found
...
<um monte de bibliotecas>
```

De fato, a libdbus é a única lib requerida pelo binário não existe no sistema... ou seja, devemos instalá-la se quisermos usar o skype.

Vale lembrar que versões anteriores do skype funcionam normalmente no **slackware**, no entanto, como o site não mantém as versões anteriores para download, acredito que na maior parte dos casos, seja necessário fazer o release mais atual funcionar.

Bem, quem disponibiliza a biblioteca é o projeto FreeDesktop, através do pacote dbus, infelizmente não disponível para o nosso querido **slackware** via builds oficiais. A única exigência do dbus é um parser xml como a libxml ou o expat, ambos disponíveis no slackware via CD's de instalação. Desse modo, não precisamos nos preocupar com isso. Vamos ao trabalho:

```
$ wget http://dbus.freedesktop.org/\
  releases/dbus-0.33.tar.gz
$ tar -xvzf dbus-0.33.tar.gz
$ cd dbus-0.33
$ ./configure --disable-glib
$ make
```

Por default, são gerados os builds para Qt e Glib, vamos gerar um build somente para Qt, afinal é esse toolkit que o skype utiliza.

```
$ su -
# cd ~clayton/dbus-0.33
# checkinstall
# ldconfig
# cd /usr/local/skype
# ldd skype | grep libdbus
libdbus-1.so.0 => not found
```

Ué... ???

```
# locate libdbus
/usr/local/lib/libdbus-qt-1.so.1.0.0
```

Hummmm, o velho truque para enganar o ld... (ainda como root)

```
# ln -s /usr/local/lib/\
  libdbus-qt-1.so.1.0.0 \
  /usr/local/lib/libdbus-1.so.1.0.0
# ldconfig
# exit
$ cd /usr/local/skype
$ ./skype
```

Prontinho, tudo funcionando!!! Espero que esse artigo seja útil a toda a comunidade...

# Controlando Banda com o HTB

## 1 - Introdução

O HTB (Hierarchical Token Bucket) é uma boa alternativa em substituição ao CBQ (Class Based Queueing) pois este é mais preciso e fácil de utilizar. A diferença para o CBQ é que ele aloca banda para uma ou mais classes e toma emprestada temporariamente a banda de outras classes que não estejam sendo utilizadas completamente.

Para utilizar o HTB você precisa de um kernel maior ou igual 2.4.20 e da ferramenta `tc` (Traffic Control) incluída no pacote `iproute2`. Eu utilizei apenas o **slackware 10.1** (kernel 2.4.29).

Para configurarmos o HTB temos basicamente três alternativas: Criar um script com todos os comandos (se você souber quais é claro); Utilizar o `htb.init` semelhante ao `cbq.init` e que demanda uma série de configurações, bem familiar para quem já utiliza o CBQ ou utilizar a ferramenta HTB Tools. Como eu quero simplificar e não tenho experiência com o CBQ optei pelo HTB Tools criada dentro da filosofia do **slackware**.

## 2 - Instalação e Configuração

Faça o download em:

```
http://htb-tools.arny.ro/\
  htb_tools-0.2.5.tar.gz
```

Descompacte e instale da maneira tradicional (`./configure && make && checkinstall`)

Antes que eu me esqueça, utilizando o HTB Tools não é necessário instalar o pacote `iproute2` (caso não queira), pois o HTB Tools já inclui uma versão adequada da ferramenta `tc` para o HTB, apenas dê um: `cp tc /sbin`. Instalado o HTB Tools seu **slackware** terá os binários :

- `q_parser`: lê o arquivo de configuração onde os clientes, as classes, e a banda alocada é definida e gera um script conforme as configurações estabelecidas;
- `q_show`: exhibe em tempo real a banda usada/alocada para cada classe/cliente de acordo com a configuração;
- `q_checkcfg`: verifica a sintaxe do arquivo de configuração.

A instalação cria arquivos de configuração padrão em `/etc/htb`. Utilizando o HTB Tools conseguimos simplificar bastante a configuração e monitoramento de alocação de banda tanto para upload como para download. A grande sacada do criador do HTB Tools (<http://www.arny.ro>) foi definir uma configuração semelhante a do arquivo `named.conf` (quem nunca deu uma espiada?).

Vamos ao exemplo: você possui um link de 512kpbs compartilhado entre dois clientes, teoricamente cada um deveria ter 256kpbs garantidos (QOS), contudo você deixou a coisa frouxa e um dos clientes começa a reclamar que o link está muito lento e que não consegue realizar transações importantes. Não precisa dizer mais nada, o outro cliente está abusando do link. A culpa não é dele, pois você deixou, não é mesmo? Para resolver este problema vamos de fato distribuir o link da seguinte forma: cada cliente terá 192kpbs garantidos e no máximo 256kpbs para upload/download.

Supondo que a sua interface WAN seja a `eth0` vamos controlar o upload criando o arquivo `/etc/htb/eth0-qos.cfg` com o seguinte conteúdo:

```
class upload {
    bandwidth 480;
    limit 512;
    burst 2;
    priority 1;
    client cliente_1 {
        bandwidth 192;
        limit 256;
        burst 2;
        priority 1;
        src {
            192.168.1.2;
        };
    };
    client cliente_2 {
        bandwidth 192;
        limit 256;
        burst 2;
        priority 1;
        src {
            192.168.2.2;
        };
    };
};
class default { bandwidth 8; };
```

Como podemos observar a configuração é auto explicativa. Mas para não deixar dúvidas podemos observar que o `src`, como devemos suspeitar, é o source ou seja a origem do tráfego, por tando estamos limitando a saída (upload). A estrutura básica pode ser resumida em uma classe principal que é subdividida dentro de outras classes secundárias. Quando existe mais de uma classe principal estas não compartilham banda entre elas. As classes secundárias (clientes) podem compartilhar banda entre elas de acordo com a configuração (limit maior). Cada classe principal possui uma ou mais classes secundárias (clientes). A classe especial `default` especifica uma banda para os outros clientes/tráfegos que não estejam contemplados na configuração. A taxa de transferência é dada em `kbit` por segundo (`kpbs`).

Para controlar o download basta criar um arquivo semelhante ao `/etc/htb/eth0-qos.cfg` em `/etc/htb/eth1-qos.cfg` supondo que a sua interface `eth1` seja da sua LAN. Em `/etc/htb/eth1-qos.cfg` substitua o nome da classe principal por `download` e o `src` por `dst`, ou seja estaremos controlando o destino (a entrada de dados para os clientes). Em configurações mais complexas você pode especificar diversos IP's ou redes (rede/máscara) dentro de uma mesma classe secundária entre as chaves do `src` ou `dst`, sempre um(a) por linha e finalizado por um ponto-e-vírgula. Agora caso você queira limitar a banda para um serviço específico por exemplo `ftp` ou `http` dê um espaço e coloque a porta do serviço ( em `src` ou `dst` ), assim :

```
dst {
    192.168.3.0/24 21;
    192.168.4.0/24 80;
};
```

Antes de ativar o controle de banda é recomendável verificar a sintaxe da configuração:

```
# q_checkcfg /etc/htb/eth0-qos.cfg
# q_checkcfg /etc/htb/eth1-qos.cfg
```

**slackware**  
linux

simplicity is...

### 3 - Ativando o HTB

Estando tudo correto vamos cuidar para que o HTB seja ativado a cada boot. Para facilitar as coisas copiamos o script `htb`, instalado junto com o HTB Tools:

```
# cp /sbin/htb /etc/rc.d/rc.htb
```

Com este script não precisamos executar diretamente os binários do HTB Tools. Para ativarmos o `htb` para a `eth0` executarmos dentro de `/etc/rc.d`:

```
# ./rc.htb eth0 start
```

Faça o mesmo para `eth1` obviamente fazendo a substituição necessária de `eth0` por `eth1`. Para garantir a ativação do HTB ao reiniciar acrescente os comandos acima no `rc.local` ou em outro script de inicialização de sua preferência.

### 4 - Monitorando o Controle de Banda

Iniciado o HTB, você pode monitorar o uso do link em tempo real, para monitorar individualmente cada cliente fazendo upload ou download, respectivamente, execute:

```
# ./rc.htb eth0 stats
# ./rc.htb eth1 stats
```

Dê uma olha no pacote HTB Tools e você ainda terá à mão um utilitário para gerar o arquivo de configuração via assistente (`htbgen`) e uma forma de monitorar a utilização da banda pela web (`q_show.php`). É mole ou quer mais?

Espero que consigam descomplicar o controle de banda com HTB Tools assim como eu consegui.

Wanderson Santiago dos Reis  
<wandersonsreis@gmail.com>

...divine!

# Integrando Linux e Netware

## 1. Início

Este texto pretende de forma simples tanto explicar como mapear compartilhamentos criados em um servidor Netware (Novell) em seu micro com Linux.

## 2. Verificando/Preparando o Kernel

Verificando o sistema atual:

```
# modprobe -l | egrep 'ncp|ipx'
```

Se retornar os dois módulos `ncpfs` e `ipx` você não precisa recompilar seu kernel. Não retornar nada pode indicar que estes módulos estão compilados como built-in na imagem do kernel, portanto antes de recompilar o kernel eu aconselho que testem criar o mapeamento antes (passo 5).

Para que seu sistema reconheça o sistema de arquivos `ncpfs`, será necessário que seu kernel tenha suporte ao mesmo e também suporte ao IPX. Acesse o diretório `/usr/src/linux` e execute:

```
# make menuconfig
```

Vá até `Networking Options` e ative:

```
<M> The IPX Protocol
    [*] IPX: Full Internal IPX Network
```

Agora volte a tela inicial e acesse:

```
File Systems
  Network File Systems
    <M> NCP File System Support
      [*] Em todos abaixo referentes ao NCP
```

Saia do `menuconfig` salvando suas alterações e faça os seguintes passos:

```
# make dep && make clean && make bzImage
# make modules && make modules_install
# cp arch/i386/boot/bzImage /boot/lin-ncp
# echo -e "
image = /boot/lin-ncp
root = `mount | head -1 | cut -c1-9`
label = NCP
read-only" >>/etc/lilo.conf
# lilo && reboot
```

Eu aconselho 100% que você leia um documento sobre recompilação do kernel, já que não descrevemos aqui todo o processo. A máquina irá reiniciar e rebootar com o kernel novo.

## 3. Obtendo o pacote `ncpfs-2.2.4.tar.gz`

Você pode baixar a versão mais atualizada do pacote `ncpfs` no site:

```
ftp://platan.vc.cvut.cz/pub/linux/ncpfs
```

A versão utilizada neste documento será a 2.2.4, para baixá-la usaremos:

```
$ wget ftp://platan.vc.cvut.cz/pub/\
linux/ncpfs/ncpfs-2.2.4.tar.gz
```

## 4. Instalando o `ncpfs-2.2.4`

Vá até o diretório onde você baixou o pacote e execute:

```
$ tar -zxvf ncpfs-2.2.4.tar.gz
$ cd ncpfs-2.2.4
$ ./configure
$ make
$ su
# make install
```

## 5. Montando o serviço Novell no Linux

Primeiro vamos conhecer um pouco do `ncpmount` e suas opções (entre parênteses os equivalentes no `/etc/fstab`):

- h .exibe o help
- n .tenta o mapeamento sem senha (`nopasswd`)
- S server .nome do servidor Netware (`server=`)
- P password .faz o mapeamento com essa senha (`passwd=`)
- U username .nome do usuário para o mapeamento (`user=`)
- V volume .nome do serviço a ser montado (`volume=`)
- m .permite vários mapeamentos com o mesmo usuário (`multiple`)

OK, com estas opções já somos capazes de montar nosso serviço Novell:

```
# ncpmount -S SERVERNAME \  
-U USERNAME \  
-V DADOS \  
-P 123456 /mnt/novell
```

Com isso eu montei o volume DADOS do servidor SERVERNAME pelo usuário USERNAME e senha 123456 no diretório do Linux /mnt/novell. Vamos adicionar uma linha ao arquivo /etc/fstab para facilitar nossa vida, com a seguinte sintaxe:

```
SERVERNAME/USERNAME    \  
    /mnt/novell    ncpfs \  
    noauto,defaults,\  
    passwd=123456,multiple
```

noauto significa que não será montado durante o boot da máquina, para que seja montado sempre que a máquina for iniciada utilize auto. Agora para montar nosso serviço Novell no Linux digitamos somente:

```
# mount /mnt/novell
```

Caso seu usuário na Novell não possua senha, utilize no método 1 a opção -n e no método 2 (fstab) a opção nopasswd invés de passwd=xxxxxx

## 6. Conclusão

Com esse pacote ncpfs e algum "trabalhinho" por parte nossa, nós temos a capacidade de integrar o Linux com serviços Netware. Cada vez mais o Linux mostrando sua compatibilidade.

Flavio do Carmo Jr a.k.a. drkn  
<billpp@gmail.com>



## Autores

**Clayton Eduardo dos Santos**, Trabalha com Linux desde 2003 e com Slackware desde 2004. Atualmente desenvolve seu projeto de pesquisa de Doutorado no Departamento de Engenharia Elétrica na USP de São Carlos e é um dos administradores do slackpoint, portal 100% voltado à comunidade slackware brasileira.

**Diego Fiori Carvalho**, É aluno do Bacharelado em Informática do ICMC-USP, São Carlos S.P., utiliza Linux desde 2002, desenvolveu um sistema de multimídia interativa para Treinamento em Linux, tem grande interesse por desenvolvimento de aplicações gráficas e atualmente é desenvolvedor da empresa 3WT de São Carlos.

**Toledo**, Iniciou com computadores em 1993 e Linux em 1998, usando Slackware 3.4 kernel 2.0.30. Desde então, vem acompanhando a evolução desse maravilhoso sistema juntamente com toda a comunidade de software livre. Atualmente editor e mantenedor do Slackware Zine.

**Flavio do Carmo Junior** a.k.a. drkn, 22 anos, cursando Sistemas de Informações. Teve seu primeiro computador, um 386, aos 10 anos que o satisfez até enjoar de "Prince of Persia" depois de uns 2 anos, voltou a mexer em computadores em 95 quando ganhou um Celeron 400Mhz, quakemaniaco de carteirinha se viu em desvantagem quando surgiu o ADSL e então iniciou-se no mundo Linux em 99 pulando por várias distribuições até conhecer o slackware. Obrigatoriamente trabalhou com debian e suse por 1 ano e meio, mas nunca se afastou do seu desktop slackware. Atualmente trabalha na área de administração de servidores Linux na Bebidas Asteca & Molhos Hinomoto.

**Frederico R. C. Costa a.k.a Cyberfred**, estudante de Sistemas de Informação. Usuário Slackware desde 1998. Trabalha como administrador de redes no jornal Diário da Manhã e no ponto de presença (PoP-GO) da RNP (Rede Nacional de Ensino e Pesquisa) em Goiás.



# Mais segurança com o AIDE

## 1. Introdução

O AIDE (Advanced Intrusion Detection Environment) para quem não sabe, é um sistema que checa a integridade de arquivos e diretórios definidos por você mesmo no arquivo de política. Ele é uma alternativa ao tão famoso (que não funciona) tripwire. Com ele é possível detectar quais arquivos foram adicionados, mudados e apagados; entre outras funções.

## 2. Como funciona?

Seu funcionamento é simples, consiste em uma criação de um banco de dados contendo informações sobre os arquivos e diretórios para mais tarde fazer a verificação entre esse banco de dados e os arquivos atuais utilizando criptografia. Após essa verificação é gerado um log de tudo que aconteceu no seu sistema.

## 3. Instalação

Requerimentos:

AIDE-0.10: <http://aide.sourceforge.net>  
Mhash-0.9.1: <http://mhash.sourceforge.net>

### 3.1 Instalando a lib Mhash

Descompacte o source e compile como outro programa qualquer, tirando a opção de compilar estaticamente a lib (a compilação dinâmica gera erros ao AIDE depois):

```
$ tar -zxvf mhash-0.9.1.tar.gz
$ cd mhash-0.9.1.tar.gz
$ ./configure --enable-static
$ make
(como root)
# make install
```

**NOTA:** O `slackware 10` vem com a lib mhash compilada dinamicamente mas precisamos dela compilada estaticamente para que o AIDE seja compilado perfeitamente.

## 3.2 Instalando o AIDE

Como todo programa source, descompactamos e compilamos:

```
$ tar -zxvf aide-0.10.tar.gz
$ cd aide-0.10
$ ./configure \
    --sysconfdir=/etc/aide \
    --with-config_file=\
        /etc/aide/aide.conf \
    --with-mhash
$ make
(como root)
# make install
```

## 4. Configuração do arquivo de política

Com tudo instalado devidamente instalado, resta agora configurar o AIDE. A sintaxe do arquivo de política é simples, para funcionar basta setar duas variáveis de sistema e depois só regras para diretórios e/ou arquivos. Abaixo segue uma tabela de descrição para as regras:

```
p -> permissão de arquivo
i -> inode
n -> quantidade de vínculos
u -> propriedade do usuário
g -> propriedade do grupo
s -> tamanho do arquivo
m -> última alteração
a -> último acesso
c -> alteração de inode
S -> alteração no tamanho
```

Concatenação de regras:

```
E -> tudo é ignorado
L -> p+i+n+u+g
R -> p+i+n+u+g+s+m+c+md5
> -> p+i+n+u+g+S
```

Isso é como se fosse uma alias ou um grupo de regras, para simplificar a vida e ficar mais fácil a visualização das mesmas.

Podem ser utilizados vários algoritmos de hash junto às regras: md5, sha1, md1600, tiger, crc32, haval e gost. Os algoritmos mais usados e confiáveis são: md5 e sha1. Mas você pode escolher e usar qual desejar.

Agora que já sabemos do que consiste o arquivo de configuração, vamos criá-lo. Como root, crie o diretório:

```
# mkdir /etc/aide
```

Crie o arquivo `/etc/aide/aide.conf` com o seguinte conteúdo:

```
#AIDE conf
  database=file:/etc/aide/aide.db
  database_out=file:/etc/aide/aide.db.new

  Regra = p+i+n+u+g+s+b+m+c+md5+sha1

  /etc p+i+u+g
  /bin Regra
  /sbin Regra
  /var Regra
  !/var/log/*
  !/var/spool/*
  !/var/adm/utmp$
```

Setamos as duas variáveis do banco que contêm as informações do sistema, definimos uma regra padrão chamada 'Regra', diretórios com a regra criada e outros com regras diferentes. O ponto de exclamação '!' funciona como uma negação, os arquivos de existem dentro do diretório `/var/log/`, `/var/spool/` e o arquivos que comecem `utmp` dentro de `/var/adm` não serão checados

## 5. Checando a integridade e analisando os logs

Agora que já temos o AIDE instalado e configurado vamos a funcionalidade :) Primeiro vamos criar o banco de dados, com o seguinte comando:

```
# aide -i
```

Irá criar o arquivo `/etc/aide/aide.db.new` baseado no seu arquivo de política.

**DICA:** Remova os arquivos de configurações e mova o banco de dados para um lugar seguro e de somente leitura, geralmente em outra máquina.

Vamos então, checar a integridade dos arquivos. A primeira vez que você criar o banco de dados você terá que fazer uma cópia dele para `aide.db` dentro do diretório de configuração, assim:

```
# cp /etc/aide/aide.db.new \
    /etc/aide/aide.db
```

E depois:

```
# aide -C > aide.log
```

Dependendo das suas políticas a checagem pode demorar um pouco. Neste momento o AIDE analisa o arquivo `aide.db.new` com o `aide.db` e gera um log das mudanças. Estamos redirecionando a saída para o arquivo `aide.log` para depois podermos analisá-lo. Um exemplo dos logs gerados pelo AIDE pode ser vista no quadro logo abaixo, preservando inclusive a formatação original.

continua

```
AIDE found differences between database and filesystem!!
Start timestamp: 2005-01-31 20:01:01
Summary:
Total number of files=7299,added files=2,removed files=0,changed files=1

Added files:
added:/etc/aide/aide.db
added:/sbin/lala
Changed files:
changed:/sbin
Detailed information about changes:

Directory: /sbin
  Size      : 6768                , 6792

  Mtime     : 1997-10-06 08:10:09      , 2005-01-31 20:00:52

  Ctime     : 2005-01-31 01:41:08      , 2005-01-31 20:00:52
```

continua

O log é bastante fácil de se entender, ele começa mostrando timestamp de geração do log em seguida de um sumário contendo: total de número de arquivos, arquivos adicionados, arquivos removidos e arquivos mudados. E logo abaixo os detalhes dos arquivos que sofreram 'mutações' :)

Para simplificar o uso do AIDE, acrescente-o no cron com o intervalo de tempo que desejar, de no mínimo uma e máximo quatro horas.

## 6. Por que não o Tripwire ?

O tripwire é um software de verificação de integridade de arquivos, assim como o AIDE. Existem duas versões: uma versão de código aberto e outra de código fechado. Infelizmente não pude testar sua versão de código aberto pois requer gcc 2.96 para compilação e não possui no momento nenhum sistema com essa gcc.

Infelizmente (de novo) não pude testar a versão de código fechado pois iria pagar por um software com o fim apenas de testá-lo.

Algumas 'features' do tripwire são melhores do que a do AIDE, como: mandar e-mail após verificação de integridade e um log mais recheado.

## 7. Conclusão

Sistemas de verificação de integridade de arquivos e diretórios ajudam aumentar mais a segurança de seus servidores, protegendo-o contra rootkits, backdoors, cavalos de tróia entre outras 'pragas'. Uma ótima solução é o AIDE e a lista TODO enchem os olhos com as novas 'features', é esperar e comprovar :)

Leandro Toledo <toledo@core-dumped.org>

## ERRATA

Na edição especial 8.5, lançada durante o Terceiro Encontro Nacional LinuxChix, foi publicada uma matéria sobre como reordenar interfaces de rede.

Infelizmente a matéria continha um pequeno erro. Onde diz para fazer:

```
# ifconfig eth0 down
# ifconfig eth1 down
# nameif -s
```

Existe um problema, não é possível passar a interface eth0 para eth1 enquanto a eth1 continua lá, muito menos passar a eth1 para eth0. Com isso, muitos leitores conseguiram um erro.

Então, o ideal é fazer:

```
# ifconfig eth0 down
# ifconfig eth1 down
# nameif eth2 00:E0:7D:87:8E:20
# nameif eth0 00:02:55:32:21:02
# nameif eth1 00:E0:7D:87:8E:20
```

Assim, primeiro passamos a eth0 para eth2, depois a eth1 para eth0 e, por fim, a eth2 para eth1. Usando uma variável "auxiliar" (a eth2), como utilizamos para trocar o valor de duas variáveis em programação.

Usar o nameif -s durante o boot irá funcionar perfeitamente, pois ele será utilizado antes de qualquer interface estar configurada, ou seja, não é necessário se preocupar com essa parte do artigo.

got **slack** ?

<http://store.slackware.com>

# A autenticação do SQUID via Samba

Este roteiro foi criado para auxiliar a configuração de um servidor SQUID para tornar automática a autenticação de um usuário utilizando Internet Explorer.

O SQUID possui um módulo de autenticação (ntlm\_auth) que permite que usuários utilizem o mesmo login/senha tanto para logar no domínio, quanto para autenticar-se no browser.

Se o usuário utilizar S.O Windows com browser Internet Explorer, esta autenticação pode ser automatizada, ou seja, após o usuário se logar no domínio, não será necessário informar novamente o login/senha para poder navegar.

As configurações contidas neste documento foram criadas em um servidor **slackware 9.1**, glibc-2.3.2, kernel 2.4.24, samba-3.0.2a e squid-2.5.STABLE5.

Este processo funcionou em um ambiente onde não é utilizado TRANSPARENT PROXY, ainda não fiz testes se esta solução funciona com Transparent Proxy, gostaria de ter um feedback de quem testar.

## 1. Baixe os sources do Samba e Squid:

```
# wget -c -t0 \
  http://us2.samba.org/samba/\
  ftp/samba-latest.tar.gz
# wget -c -t0 \
  http://www.squid-cache.org/\
  Versions/v2/2.5/\
  squid-2.5.STABLE5.tar.gz
```

## 2. Descompacte e compile:

### 2.1 - SAMBA

```
# tar xvfz samba-latest.tar.gz
# cd samba-3.0.2a/sources
# ./configure --prefix=/usr \
  --sysconfdir=/etc/samba \
  --with-privatedir=\
  /etc/samba/private \
  --with-lockdir=/var/lock \
  --with-piddir=/var/run \
  --with-configdir=/etc/samba \
  --with-automount --with-smbmount \
  --with-quotas --with-winbind \
  --with-winbind-auth-challenge
# make
# make install
```

### 2.2 - SQUID

```
# cd squid-2.5.STABLE5
# ./configure --prefix=/usr \
  --sysconfdir=/etc/squid \
  --enable-delay-pools \
  --enable-snmp --enable-htcp \
  --enable-ssl \
  --enable-default-err-language=\
  Portuguese \
  --enable-linux-netfilter \
  --enable-auth=ntlm,basic \
  --enable-external-acl-helpers=\
  wbinfo_group \
  --with-samba-sources=../samba-3.0.2a
# make
# make install
```

continua

[A-Za-z0-9]+[Zz]ine  
existem várias...

### 3. Configurando os serviços:

#### 3.1 - Servidor SAMBA (PDC)

No meu caso o SAMBA utilizado como servidor de domínio (PDC) utiliza a mesma versão do compilado no servidor SQUID, ou seja, samba-3.0.2a. No servidor (PDC) é necessário adicionar o usuário root ao smbpasswd:

```
# smbpasswd -a root
# smbpasswd -e root
```

Não testei se esta solução funciona com versões diferentes do samba no servidor e no cliente. Caso alguém teste, gostaria de ter um feedback.

#### 3.2 - Servidor SQUID

##### 3.2.1 - Configuração do SAMBA (winbind)

Esta máquina deverá rodar os serviços smbd, nmbd e winbindd além, é claro, do squid. Para configurar o SAMBA nesta máquina, que servirá apenas para fazer as consultas de usuário e senha via SMB, você deverá editar o arquivo /etc/smb.conf:

```
workgroup = SEU_WORKGROUP
server string = Squid Server
netbios name = squid
log level = 2
log file = /var/log/samba/%m.log
max log size = 50
security = domain
password server = \
    NOME-ou-IP_do_Servidor_SAMBA_(PDC)
socket options = \
    TCP_NODELAY \
    SO_RCVBUF=8192 \
    SO_SNDBUF=8192
encrypt passwords = yes
winbind uid = 10000-20000
winbind gid = 10000-20000
winbind enum users = yes
winbind enum groups = yes
template homedir = /dev/null
template shell = /dev/null
winbind use default domain = yes
```

continua

##### 3.2.2 - Configuração do SQUID

As seguintes linhas devem ser adicionadas ao /etc/squid/squid.conf:

```
# Grupo que o usuario que roda o
# SQUID (nobody) faz parte
cache_effective_group nobody

# ACL indicando que AuthorizedUsers
# requer autenticao para navegar
acl AuthorizedUsers proxy_auth REQUIRED

# Libera acesso para ACL AuthorizedUsers
http_access allow AuthorizedUsers

# Configuracao p/ o Internet Explorer
# autenticar automaticamente
auth_param ntlm program \
    /usr/bin/ntlm_auth \
        --helper-protocol=\
            squid-2.5-ntlmssp
auth_param ntlm children 5
auth_param ntlm max_challenge_reuses 0
auth_param ntlm max_challenge_lifetime \
    20 minutes

# Configuracao p/ Netscape, Mozilla etc.
# que nao autenticam automaticamente,
# sendo necessário informa login/senha
auth_param basic \
    program /usr/bin/ntlm_auth \
        --helper-protocol=\
            squid-2.5-basic
auth_param basic children 5
auth_param basic realm \
    Squid Proxy. Enter with \
        username/password
auth_param basic credentialsttl 2 hours
```

#### IMPORTANTE:

ao editar o squid.conf, procure no arquivo onde está cada parâmetro explicitado (no exemplo acima) e o substitua, pois, a localização de cada um destes parâmetros no arquivo de configuração é muito importante.

continua

# slackware

## zine

só existe uma.

### 3.2.3 - Configuração e teste do Winbind

Inicie os seguintes serviços:

```
# smb -D
# nmbd -D
# winbindd
```

É necessário inserir a máquina SQUID no domínio do SAMBA (PDC), para isto, siga os seguintes passos:

```
# net join -W WORKGROUP -U root
```

No comando acima, deve-se informar a senha cadastrada para o usuário root no smbpasswd do PDC. Se tudo estiver correto, você receberá uma mensagem de "bem-vindo ao domínio".

Para testar se o Winbind está funcionando perfeitamente, execute os comandos abaixo:

```
# wbinform -t
checking the trust secret via RPC \
calls succeeded
```

Se você recebeu uma mensagem como a acima citada, as configurações estão OK.

```
# wbinform -u
```

O comando acima deverá mostrar uma lista dos usuários cadastrados no SAMBA (PDC).

```
# wbinform -a WORKGROUP\mane%senha
plaintext password authentication succeeded
challenge/response password \
authentication succeeded
```

O comando acima testa a autenticação do usuário no domínio, que no neste caso, USUÁRIO=mane, SENHA=senha e DOMÍNIO=WORKGROUP. Se você receber as mensagens como no exemplo, então está tudo PERFEITO.

Para o squid ter acesso às informações do Winbind é necessário alterar o grupo do diretório /var/lock/winbindd\_privileged para o mesmo explicitado no squid.conf em cache\_effective\_group, que, no exemplo aqui citado, é o grupo nobody, assim:

```
# chgrp nobody \
/var/lock/winbindd_privileged
```

### 4. Finalizando:

Inicie o squid, sendo que a primeira vez que se inicia o squid, é necessário antes, criar os diretórios de cache, para isto:

```
# squid -z
```

Criado os diretórios de cache, inicie o SQUID, com o seguinte comando:

```
# squid -D
```

Em uma estação Windows, logue no domínio, configure o Internet Explorer para utilizar servidor Proxy, feche o Browser. Ao abrir o browser novamente verá que não será solicitado senha para navegar.

Agora que tudo funcionou, lembre-se de incluir o no seu rc.local uma chamada para o squid. Afinal, você não vai querer ter que reinicia-lo manualmente (ou correr o risco de esquecer de inicia-lo) no próximo boot do seu server.

No caso do samba, existe um pacote que vem no próprio slackware, lembre-se de removê-lo antes de instalar o nosso, compilado com suporte ao winbind.

Pensando ainda em facilitar a sua vida, lembre-se de fazer um pacote de ambos os programas, assim, se precisar deles em uma próxima vez, os pacotes já estarão prontos.

Frederico R. C. Costa  
<fred@ufgnet.ufg.br>

← continua

Artigos técnicos,  
escritos por técnicos  
e para técnicos.

# Backups usando fitas DAT

Como todos nós sabemos, o backup é uma prática extremamente importante mas nem sempre priorizada pelas empresas, bem, pelo menos até a primeira perda dos dados... :) Um dos métodos mais usuais para realização de backups é através da utilização de unidades de fitas DAT.

O Linux nomeia os dispositivos de fita como /dev/stx, ou /dev/nstx, onde "x" pode ser igual a 0,1,2 e assim por diante, de acordo com o número de unidades que o seu sistema possui. Como de costume, é bem possível que, para "economizar um troco", seu cliente tenha adquirido uma unidade IDE, desse modo, para que o dispositivo funcione, você deve realizar uma emulação SCSI, da mesma maneira como é realizado com gravadores de CD.

```
# dmesg | grep -i tape
hdd: Seagate STT20000A, ATAPI TAPE drive
```

Sabendo o dispositivo da gravadora, basta adicionar a seguinte linha no seu lilo.conf:

```
append="hdd=ide-scsi"
```

Em seguida, execute o lilo e reinicie a máquina...

Agora, o dispositivo já pode ser acessado através do arquivo de dispositivo: /dev/st0

Se o seu dispositivo for SCSI e sua controladora estiver configurada e funcionando, provavelmente ela deve ser acessada diretamente pelo arquivo acima sem a necessidade de nenhum "ajuste".

Agora, para testar a unidade, nada melhor que um backup:

```
# tar -cjvf /dev/st0 /diretório_backup
```

Backup de múltiplos diretórios e/ou arquivos (\*):

```
# tar -cjv \
  -T /backup/lista_para_backup.txt \
  -f /dev/st0
```

Para restaurar (todo) o backup:

```
# tar -xjvf /dev/st0
```

O arquivo lista\_para\_backup.txt, contém a lista de arquivos e/ou diretórios que irão compor o backup, como por exemplo:

```
/home/postgres/backup/
/arquivos_rede
/etc/passwd
/var/log/
```

A flag "j" utilizada nas instruções de criação/restauração, compacta os dados com o bzip2 antes da cópia para a fita.

Você também pode obter informações sobre o cron para agendar backups ou ainda utilizar ferramentas gráficas de backup, que na opinião de alguns, são mais fáceis de gerenciar. Sua escolha vai depender do período em que os backups serão realizados.

Para habilitar um backup diário, crie um arquivo dentro do /etc/cron.daily chamado backup\_diário, com o seguinte conteúdo:

```
# Limpa a fita
mt -f /dev/st0 erase

# Faz backup do banco (por exemplo)
sudo -u postgres \
  /usr/local/pgsql/bin/pg_dumpall > \
  ~postgres/backup/backup_banco
```

```
# Registrando a data
date > ~postgres/backup/backup_date.txt
```

```
# Grava na fita
tar -cjv -T \
  ~postgres/backup/backup.txt \
  -f /dev/st0
```

Em seguida dê ao arquivo permissão de execução com o comando chmod. Os processos para backup semanal, mensal, de hora em hora, etc são análogos e poderão ser realizados facilmente, não deixem de dar uma olhada em:

```
cron.daily
cron.hourly
cron.monthly
cron.weekly
```

Todos dentro do /etc. Um crontab -l mostra os horários e dias em que os scripts de cada um desses diretórios são executados. Um man crond e crontab também não fazem mal a ninguém... :)

Existem inúmeras maneiras de se realizar esse tipo de backup, bem como, um grande número de ferramentas modo texto/gráficas disponíveis para esse fim. Acredito que esse breve tutorial seja suficiente e funcional para a maior parte dos casos.