

# Aos iniciantes em FreeBSD e UNIX

## Resumo

Parabéns por instalar o FreeBSD! Esta introdução é para iniciantes em FreeBSD e UNIX®-então vamos começar do básico.

## Índice

|   |    |
|---|----|
| 1. Logando e Deslogando .....                           | 1  |
| 2. Adicionando um usuário com privilégios de Root ..... | 2  |
| 3. Explorando .....                                     | 3  |
| 4. Obtendo ajuda e informações .....                    | 4  |
| 5. Editando textos .....                                | 5  |
| 6. Outros comandos úteis .....                          | 7  |
| 7. Próximos Passos .....                                | 8  |
| 8. Seu ambiente de trabalho .....                       | 9  |
| 9. Outros .....   | 10 |
| 10. Comentários são bem-vindos .....                    | 10 |

## 1. Logando e Deslogando

Entre com o usuário que você criou na instalação (quando você visualizar o prompt **login:**) ou como **root**. (Sua instalação do FreeBSD já terá uma conta para o **root**; esse usuário pode ir a qualquer lugar e fazer qualquer coisa, incluindo deletar arquivos essenciais do sistema, então tenha cuidado!) Os símbolos **%** e **no** prompt (o seu pode ser diferente) indicam o tipo do usuário logado, o símbolo **%** indica um usuário comum e **no** indica que o usuário é **root**.

Para se sair (e obter um novo **login:**), digite

```
# exit
```

quantas vezes forem necessárias. Pressione **enter** após digitar os comandos, e lembre-se que o UNIX® diferencia letras maiúsculas de minúsculas, então, digite **exit**, e não **EXIT**.

Para desligar o computador, digite

```
# /sbin/shutdown -h now
```

Ou para reiniciar, digite

```
# /sbin/shutdown -r now
```

ou

```
# /sbin/reboot
```

Você também pode reiniciar com `Ctrl + Alt + Delete`. Aguarde um instante para a conclusão do processo. Esta sequência de teclas é equivalente ao `/sbin/reboot` nas versões recentes do FreeBSD, e claro, este procedimento é muito, mas muito melhor do que simplesmente pressionar o botão reset do seu computador. Você não quer ter que reinstalar o sistema, não é mesmo?

## 2. Adicionando um usuário com privilégios de Root

Se você não criou um usuário quando instalou o sistema operacional e está logado como `root`, você provavelmente deve criar um usuário com

```
# adduser
```

A primeira vez que você usar o comando `adduser`, ele poderá solicitar alguns padrões para salvar. Você pode querer tornar o `cs(1)` no shell padrão ao invés do `sh(1)`, se for sugerido o `sh` como padrão. Caso contrário, basta pressionar enter para aceitar cada padrão. Esses padrões são salvos em `/etc/adduser.conf`, um arquivo editável.

Suponha que você crie um usuário `jack`, o qual possui *Jack Benimble* como nome completo. Dê a `jack` uma senha se a segurança (até mesmo crianças ao redor que podem esbarrar no teclado) for um problema. Quando perguntarem se você deseja convidar `jack` para outros grupos, digite `wheel`

```
O grupo de login é "jack". Convide o jack para outros grupos: wheel
```

Isso tornará possível efetuar login como `jack` e usar o comando `su(1)` para tornar-se `root`. Dessa forma, você não será mais repreendido por fazer login como `root`.

Você pode sair do `adduser` a qualquer momento digitando `Ctrl + C`, e no final você terá a chance de aprovar seu novo usuário ou simplesmente digitar `n` para não criá-lo. Você pode querer criar um segundo novo usuário para que ao editar os arquivos de login do `jack`, você tenha um login de reserva caso algo dê errado.

Depois de fazer isso, use `exit` para voltar a um prompt de login e efetuar login como `jack`. Em geral, é uma boa ideia fazer o máximo possível de trabalho como um usuário comum que não tenha o poder --e o risco-- do usuário `root`.

Se você já criou um usuário e deseja que o usuário seja capaz de executar o comando `su` para logar-

se como **root**, é possível efetuar login como **root** e editar o arquivo `/etc/group`, adicionando **jack** à primeira linha (no grupo **wheel**). Mas primeiro você precisa praticar com o **vi(1)**, o editor de texto - ou usar um editor de texto mais simples, como por exemplo, o **ee(1)**, instalado em versões recentes do FreeBSD.

Para excluir um usuário, use o comando **rmuser**.

## 3. Explorando

Logado como um usuário comum, explore e experimente alguns comandos que acessarão as fontes de ajuda e informações dentro do FreeBSD.

Aqui estão descritos alguns comandos e o que eles fazem:

### **id**

Diz a você quem você é!

### **pwd**

Mostra onde você está—quanto ao diretório atual de trabalho.

### **ls**

Lista os arquivos presentes no diretório atual.

### **ls -F**

Lista os arquivos do diretório atual com um **\*** depois dos executáveis, um **/** depois dos diretórios, e um **@** depois de links simbólicos.

### **ls -l**

Lista os arquivos em formatos estendidos—tamanho, data, permissões.

### **ls -a**

Lista arquivos "dot" junto com os outros. Se você é **root**, os arquivos com nome iniciando por um "ponto" serão mostradas sem a opção **-a**.

### **cd**

Altera o diretório atual de trabalho. **cd ..** te leva ao diretório antecessor do atual; note o espaço depois de **cd**. **cd /usr/local** te leva a esse mesmo diretório. **cd ~** te leva ao diretório home do usuário que você está logado—exemplo: `/usr/home/jack`. Tente **cd /cdrom**, e depois **ls**, para saber se o CDROM está montado e funcionando.

### **less filename**

Te permite ver um arquivo (chamado *filename*) sem alterá-lo. Tente **less /etc/fstab**. Digite **q** para sair.

### **cat filename**

Exibe o conteúdo do arquivo *filename* na tela. Se o conteúdo for longo a ponto de que você consiga ler apenas o final, pressione **ScrollLock** e use **up-arrow** para retroceder; você pode usar **ScrollLock** em páginas de manual também. Pressione **ScrollLock** novamente para parar de rolar.

Você também pode querer experimentar `cat` em alguns arquivos com nomes iniciados por ponto em seu diretório home — `cat .cshrc`, `cat .login`, `cat .profile`.

Você notará os aliases em `.cshrc` para alguns dos comandos `ls`. (Eles são muito convenientes). Você pode criar aliases ao editar `.cshrc`. Você pode disponibilizar esses aliases para todos os usuários do sistema, colocando-os no arquivo de configuração global do `cs`h localizado em `/etc/csh.cshrc`.

## 4. Obtendo ajuda e informações

Aqui estão algumas fontes úteis de ajuda. A palavra *Texto* deve ser substituída por algo de sua escolha—geralmente um comando ou nome de arquivo.

### `apropos texto`

Tudo que contém a palavra *texto* no `whatis` database.

### `man texto`

Exibe a página de manual referente ao *texto*. A principal fonte de documentação para os sistemas UNIX®. O comando `man ls` lhe dirá todas as maneiras de usar o comando `ls`. Pressione `Enter` para percorrer o texto, `Ctrl + B` para voltar a página anterior, `Ctrl + F` para avançar, `q` ou `Ctrl + C` para sair.

### `which texto`

Te informa em qual local no path do usuário o comando *texto* é encontrado.

### `locate texto`

Todos os paths nos quais a string *texto* é encontrada.

### `whatis texto`

Te informa o que o comando *texto* faz e sua página de manual. Digitando `whatis *` a saída do comando te informará sobre todos os binários no diretório atual.

### `whereis texto`

Encontra o arquivo *texto*, te informando seu path completo.

Você pode experimentar usar `whatis` em alguns comandos úteis e comuns como `cat`, `more`, `grep`, `mv`, `find`, `tar`, `chmod`, `chown`, `date`, e `script`. `more` te permite ler uma página por vez como no DOS, exemplo: `ls -l | more` ou `more filename`. O símbolo `*` funciona como um caractere curinga—exemplo: `ls w*` mostrará os arquivos que começam com `w`.

Alguns deles não estão funcionando muito bem? Ambos `locate(1)` e `whatis(1)` dependem de um banco de dados que é reconstruído semanalmente. Se a sua máquina não for ficar ligada nos fins de semana (e rodando FreeBSD), você pode executar os comandos a seguir para que ela execute manutenções diárias, semanais, mensais ou apenas de vez em quando. Execute-os como `root` e dê a cada comando o tempo necessário para ser finalizado antes de executar o próximo.

```
# periodic daily
output omitted
# periodic weekly
output omitted
# periodic monthly
output omitted
```

Se você se cansar de esperar, pressione **Alt** + **F2** para obter outro *console virtual*, e efetue seu login novamente. Afinal, é um sistema multi-usuário e multitarefa. No entanto, esses comandos provavelmente irão piscar mensagens na tela enquanto eles estiverem em execução; você pode digitar **clear** no prompt para limpar a tela. Uma vez executados, você pode querer olhar o conteúdo do `/var/mail/root` e do `/var/log/messages`.

A execução de tais comandos faz parte da administração do sistema—e como usuário único de um sistema UNIX®, você é seu próprio administrador de rede. Praticamente tudo o que você precisa para ser **root** é administrar o sistema. Tais responsabilidades não são abordadas nem mesmo naqueles livros gigantescos de UNIX®, os quais parecem dedicar muito espaço para menus e gerenciadores de janelas. Você pode querer obter um dos dois principais livros sobre administração de sistemas, pode ser o Evi Nemeth et.al.'s UNIX System Administration Handbook (Prentice-Hall, 1995, ISBN 0-13-15051-7)--a segunda edição com a capa vermelha; ou o livro Æleen Frisch's Essential System Administration (O'Reilly & Associates, 2002, ISBN 0-596-00343-9). Eu usei Nemeth.

## 5. Editando textos

Para configurar o seu sistema, você precisará editar arquivos de texto. A maioria deles estará no diretório `/etc`; e você precisará utilizar o comando **su** para acessar o usuário **root** do sistema e assim poder alterá-los. Você pode utilizar um editor de texto simples, como o **ee**, mas a longo prazo, é recomendado aprender a utilizar o editor de texto **vi**. Existe um excelente tutorial do editor de texto **vi** disponível em `/usr/src/contrib/nvi/docs/tutorial`, se você tiver o código fonte do sistema instalado.

Antes de editar um arquivo, é recomendado criar um backup do mesmo. Suponha que você queira editar o `/etc/rc.conf`. Você poderia usar apenas **cd /etc** para acessar o diretório `/etc` e usar o seguinte comando:

```
# cp rc.conf rc.conf.orig
```

Isso iria copiar `rc.conf` para `rc.conf.orig`, e após isso você poderia copiar `rc.conf.orig` para `rc.conf` para recuperar o original. Mas melhor ainda seria mover (renomear) e depois copiar de volta:

```
# mv rc.conf rc.conf.orig
# cp rc.conf.orig rc.conf
```

porque o comando **mv** preserva a data original e o proprietário do arquivo. Agora você pode editar o `rc.conf`. Se você quiser o original de volta, você deve então executar **mv rc.conf rc.conf.myedit**

(considerando que você quer preservar sua versão editada) e então

```
# mv rc.conf.orig rc.conf
```

para colocar as coisas de volta da maneira que estavam.

Para editar um arquivo, digite

```
# vi filename
```

Movimente-se através do texto com as teclas direcionais. `Esc` (a tecla de escape) coloca o editor `vi` em modo de comando. Aqui estão alguns comandos:

**x**

deleta a letra localizada onde o cursor está

**dd**

Apaga a linha inteira (mesmo se ela não estiver por inteiro aparecendo na tela)

**i**

Inserir um texto na posição do cursor

**a**

Inserir um texto após a posição do cursor

Uma vez que você digitou **i** ou **a**, você pode entrar com o texto. Se você pressionar **Esc** você colocará novamente o editor em modo de comando, e lá será onde você irá digitar

**:w**

para escrever suas alterações no disco e continuar editando

**:wq**

para gravar e sair

**:q!**

para sair do arquivo sem salvar as alterações

**/texto**

para mover o cursor para a primeira ocorrência da palavra *texto*; / `Enter` (a tecla Enter) para encontrar a próxima ocorrência da palavra *texto*.

**G**

para ir para o final do arquivo

**nG**

para ir para a linha *n* dentro do arquivo em edição, onde *n* é um número

**Ctrl + L**

para recarregar a tela

**Ctrl + b** and **Ctrl + f**

retrocede e avança uma tela, como fazem com **more** e **view**.

Pratique com o **vi** em seu diretório home criando um novo arquivo com **vi filename** e adicionando e excluindo texto, salvando o arquivo, e o chamando novamente. O **vi** oferece algumas surpresas porque ele realmente é muito complexo e, às vezes, você irá executar um comando que fará algo que você não espera. (Algumas pessoas realmente gostam do **vi**— é mais poderoso que o DOS EDIT— saiba mais sobre o comando **:r**). Use **Esc** uma ou mais vezes para ter certeza de que você está no modo de comando e prossiga dali quando ele lhe der problema, salve frequentemente com **:w**, e use **:q!** para sair e começar novamente (do seu último **:w**) quando você precisar.

Agora você pode usar o comando **cd** para ir para o /etc, use o comando **su** para logar como **root**, use o **vi** para editar o arquivo /etc/group, e adicione um usuário ao grupo **wheel**, assim, o usuário terá privilégios de root. Basta adicionar uma vírgula e o nome do login do usuário ao final da primeira linha do arquivo, pressione **Esc**, e use **:wq** para salvar as alterações no disco e sair. Instantaneamente eficaz. (Você não colocou um espaço após a vírgula, colocou?)

## 6. Outros comandos úteis

**df**

mostra o espaço dos sistemas de arquivos que estão montados.

**ps aux**

mostra os processos em execução. O **ps ax** é uma forma mais básica do comando.

**rm filename**

deleta o arquivo *filename*.

**rm -R dir**

remove o diretório *dir* e todos os subdiretórios — use com cuidado!

**ls -R**

lista os arquivos no diretório atual e em todos os subdiretórios; Eu usei uma variante, **ls -AFR > where.txt**, para obter uma lista de todos os arquivos no diretório / e /usr (separadamente) antes de encontrar melhores maneiras de encontrar arquivos.

**passwd**

para mudar a senha do usuário (ou a senha do usuário **root**).

**man hier**

exibe as páginas do manual para o sistema de arquivos UNIX® e sua estrutura hierárquica.

Use o comando **find** para localizar *filename* no diretório /usr e nos seus subdiretórios:

```
% find /usr -name "filename"
```

Você pode usar `*` como um caractere curinga em `"filename"` (que deve estar entre aspas). Se você utilizar o comando `find` para procurar no `/` em vez de `/usr` ele procurará o(s) arquivo(s) em todos os sistemas de arquivos montados, incluindo o CDROM e a partição DOS.

Um excelente livro que explica os comandos e demais utilitários do UNIX® é o Abrahams & Larson, *Unix for the Impatient* (Addison-Wesley, 1996, segunda edição). Há também toneladas de informações na internet sobre o UNIX®.

## 7. Próximos Passos

Agora você deve ter as ferramentas necessárias para explorar e editar arquivos, para que você possa colocar tudo em funcionamento. Há uma grande quantidade de informações no Handbook do FreeBSD (que provavelmente está no seu disco rígido) e no [site oficial do FreeBSD](#). Uma grande variedade de pacotes e ports estão no CD-ROM e no site. O Handbook traz mais informações sobre como usá-los (obtenha o pacote, se existir, com o comando `pkg_add /cdrom/packages/All/packagename`, onde *packagename* é o nome do pacote). O CDROM possui listas de pacotes e ports com descrições breves em `cdrom/packages/index`, `cdrom/packages/index.txt`, e `cdrom/ports/index`, e descrições completas em `/cdrom/ports/*/*/pkg/DESCR`, onde os `*`s representam subdiretórios de tipos de programas e nomes de programas, respectivamente.

Se você achar o handbook muito sofisticado (com isso do `ln_dir` e tudo mais) ao instalar os ports do CDROM, veja a seguir o que geralmente funciona:

Encontre o port que você deseja, digamos o `kermit`. Haverá um diretório para ele no CDROM. Copie o subdiretório para `/usr/local` (este é um bom lugar para softwares que você adicione no sistema e que devem estar disponíveis para todos os usuários) com:

```
# cp -R /cdrom/ports/comm/kermit /usr/local
```

Isto deve resultar em um subdiretório `/usr/local/kermit` que possui todos os arquivos que o subdiretório `kermit` no CDROM possui.

Em seguida, crie o diretório `/usr/ports/distfiles` se ele ainda não existir usando o comando `mkdir`. Agora verifique em `/cdrom/ports/distfiles` um arquivo com um nome que indique o port que você deseja. Copie esse arquivo para `/usr/ports/distfiles`; você pode pular esse passo em versões recentes, pois o FreeBSD fará isso para você. No caso do `kermit`, não existe um distfile.

Em seguida, use o comando `cd` para ir para o subdiretório `/usr/local/kermit` que contém o arquivo `Makefile`. Digite

```
# make all install
```

Durante este processo, o port irá usar o FTP para obter os arquivos compactados de que precisa e



que ele não encontrou no CDROM ou no `/usr/ports/distfiles`. Se a sua rede ainda não está configurada e não havia nenhum arquivo para o port em `/cdrom/ports/distfiles`, você terá que pegar o distfile usando outra máquina e copiá-lo manualmente para `/usr/ports/distfiles`. Leia o Makefile (com `cat` ou `more` ou `view`) para descobrir onde ir (o principal site de distribuição) para obter o arquivo e o nome dele. (Use transferências binárias de arquivos!) Em seguida volte para o diretório `/usr/local/kernit`, encontre o diretório com Makefile, e digite `make all install`.

## 8. Seu ambiente de trabalho

Seu shell é a parte mais importante do seu ambiente de trabalho. O shell é o que interpreta os comandos que você digita na linha de comando e, assim, se comunica com o restante do sistema operacional. Você também pode escrever scripts de shell, que consiste em uma série de comandos para serem executados sem intervenção.

Dois shells vem instalados com o FreeBSD: `csch` e `sh`. `csch` é bom para trabalhos na linha de comando, mas scripts devem ser escritos com o `sh` (ou `bash`). Você pode descobrir qual shell você utiliza executando o comando `echo $SHELL`.

O `csch` é bom, mas o `tcsh` faz tudo que o `csch` faz e ainda mais. Ele permite recuperar comandos com as teclas de seta e editá-los. Ele permite usar a tecla tab para completar o nome dos arquivos (o `csch` usa a tecla `Esc`), e permite alternar o diretório em que você estava por último com o comando `cd -`. E também é muito mais fácil alterar o seu prompt com o `tcsh`. Ele tornará sua vida muito mais fácil.

Aqui estão os dois passos para instalar um novo shell:

1. Instale o shell como um port ou um pacote, como faria com qualquer outro port ou pacote.
2. Use o comando `chsh` para alterar permanentemente seu shell para `tcsh`, ou digite `tcsh` no prompt para alterar seu shell sem precisar efetuar login novamente.



Pode ser perigoso alterar o shell do usuário `root` para outro diferente do `sh` ou `csch` nas primeiras versões do FreeBSD e também em muitas outras versões do UNIX®; talvez você não tenha um shell funcional quando o sistema entrar em modo de single user. A solução é usar `su -m` para se tornar `root`, o que lhe dará o `tcsh` como `root`, por que o shell faz parte do ambiente. Você pode tornar isso permanente adicionando-o ao seu arquivo `.tcshrc` como um alias, para isso, execute:

```
alias su su -m
```

Quando o `tcsh` é iniciado, ele lê os arquivos `/etc/csh.cshrc` e `/etc/csh.login`, assim como o `csch`. Ele também lerá o arquivo `.login` no seu diretório home e também o arquivo `.cshrc`, a menos que você forneça um arquivo `.tcshrc`. Você pode fazer isso simplesmente copiando o `.cshrc` para `.tcshrc`.

Agora que você instalou o `tcsh`, você pode ajustar o seu prompt. Você pode encontrar os detalhes na página de manual do `tcsh`, mas aqui está uma linha para colocar em seu `.tcshrc` que informará quantos comandos você digitou, qual o horário atual, e em qual diretório você está. Ele também produz um `>` se você for um usuário comum e `#` se você for `root`, mas o `tcsh` fará isso em qualquer caso:

```
set prompt = "%h %t %~ %# "
```

Isso deve ir no mesmo lugar em que está a linha de prompt atual caso exista uma, ou logo abaixo de "if(\$?prompt) then" caso ela não exista. Comente a linha antiga; você sempre pode voltar a ela se preferir. Não esqueça os espaços e as aspas. Você pode forçar a releitura do `.tcshrc` digitando `source .tcshrc`.

Você pode obter uma listagem de outras variáveis de ambiente que estão configuradas digitando `env` no prompt. O resultado mostrará seu editor padrão, pager e tipo de terminal, entre possivelmente muitas outras variáveis. Um comando útil se você efetuar o seu login a partir de um local remoto e não puder executar um programa porque o terminal não é capaz de fazê-lo é o `setenv TERM vt100`.

## 9. Outros

Como `root`, você pode desmontar o CDROM com o comando `/sbin/umount /cdrom`, retire-o da unidade, insira outro e monte-o com `/sbin/mount_cd9660 /dev/cd0a /cdrom` considerando que `cd0a` é o nome do dispositivo da sua unidade de CDROM. As versões mais recentes do FreeBSD permitem montar o CDROM apenas com o comando `/sbin/mount /cdrom`.

Usar o live filesystem — o segundo dos discos de CDROM do FreeBSD — é útil se você tiver espaço limitado. O que está no live filesystem varia de release para release. Você pode tentar jogar jogos a partir do CDROM. Isso envolve o uso do `lndir`, que é instalado com o X Window System, para informar ao(s) programa(s) onde encontrar os arquivos necessários, porque eles estão no sistema de arquivos do `/cdrom` em vez do `/usr` e seus subdiretórios, que é onde eles devem estar. Leia o manual deste utilitário executando `man lndir`.

## 10. Comentários são bem-vindos

Caso você venha a usar este guia, eu estaria interessada em saber onde ele pode não ter ficado claro e o que foi deixado de fora que você acha que deveria ser incluído, e se ele foi útil para você. Meus agradecimentos a Eugene W. Stark, professor de ciência da computação na SUNY-Stony Brook, e também ao John Fieber pelos comentários úteis.

Annelise Anderson, [andrsn@andrsn.stanford.edu](mailto:andrsn@andrsn.stanford.edu)