Para iniciantes tanto no FreeBSD quanto no UNIX®

Resumo

Parabéns por instalar o FreeBSD! Esta introdução é destinada a usuários iniciantes tanto no FreeBSD quanto no UNIX® - então ela começa com o básico.

Índice

1. Logando e Deslogando 1
2. Adicionando um Usuário com Privilégios de Root 2
3. Explorando o Sistema 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

Faça login (quando você ver `login:`) com o usuário que você criou durante a instalação ou como `root`. (Sua instalação do FreeBSD já terá uma conta para o usuário `root`, o qual pode ir a qualquer lugar e fazer qualquer coisa, incluindo excluir arquivos essenciais, então tenha cuidado!) Os símbolos % e # abaixo representam o prompt (o seu pode ser diferente), sendo que o % indica um usuário comum e o # indica o usuário `root`.

Para sair (e obter um novo prompt `login:`), digite

```
# exit
```

quantas vezes for necessário. Sim, pressione `enter` após os comandos e lembre-se de que o UNIX® é sensível a maiúsculas e minúsculas - use `exit`, 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 o sistema com `Ctrl` + `Alt` + `Delete`. Dê um pouco de tempo para que o sistema execute suas tarefas. Isso é equivalente a `/sbin/reboot` nas versões recentes do FreeBSD e é muito, muito melhor do que pressionar o botão de reset. Você não quer ter que reinstalar tudo, certo?

2. Adicionando um Usuário com Privilégios de Root

Se você não criou nenhum usuário durante a instalação do sistema e, portanto, está conectado como `root`, é provável que você queira criar um usuário agora com o seguinte comando

```
# adduser
```

A primeira vez que você usa o comando `adduser`, ele pode solicitar algumas opções padrão para salvar. Você pode querer definir o shell padrão como csh(1) em vez de sh(1), se ele sugerir `sh` como padrão. Caso contrário, basta pressionar Enter para aceitar cada opção padrão. Essas opções padrão são salvas em /etc/adduser.conf, um arquivo que pode ser editado.

Suponha que você crie um usuário chamado `jack` com o nome completo *Jack Benimble*. Defina uma senha para `jack` se a segurança for uma preocupação (mesmo que sejam apenas crianças por perto que possam bater no teclado). Quando o comando `adduser` perguntar se você deseja convidar `jack` para outros grupos, digite `wheel`

```
Login group is "jack". Invite jack into other groups: wheel
```

Isso tornará possível fazer login como `jack` e usar o comando su(1) para se tornar `root`. Então, você não receberá mais repreensões por fazer login como `root`.

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

Depois de ter feito isso, use o comando `exit` para voltar ao prompt de login e faça login como `jack`. Em geral, é uma boa ideia fazer o máximo de trabalho possível como um usuário comum que não

possui o poder - e o risco - de `root`.

Se você já criou um usuário e deseja que o usuário possa fazer `su` para `root`, faça login como `root` e edite o arquivo /etc/group, adicionando `jack` à primeira linha (o grupo `wheel`). Mas primeiro, você precisa praticar o uso do editor de texto vi(1), ou usar um editor de texto mais simples, o ee(1), instalado em versões recentes do FreeBSD.

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

3. Explorando o Sistema

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

Aqui estão alguns comandos e o que eles fazem:

`id`
: Diz quem você é!

`pwd`
: Exibe o diretório de trabalho atual.

`ls`
: Lista os arquivos presentes no diretório atual.

`ls -F`
: Lista os arquivos no diretório atual com um * após os arquivos executáveis, uma `/` após os diretórios e um `@` após os links simbólicos.

`ls -l`
: Lista os arquivos no diretório atual em formato longo, mostrando informações como tamanho, data e hora de modificação, permissões de arquivo e proprietário/grupo.

`ls -a`
: Lista todos os arquivos no diretório atual, incluindo arquivos ocultos (com ponto no início do nome). Se você estiver logado como `root`, os arquivos ocultos serão listados sem a necessidade do uso do parâmetro `-a`.

`cd`
: O comando `cd` é usado para mudar de diretório. `cd ..` volta um nível; observe o espaço após o `cd`. `cd /usr/local` navega até esse diretório. `cd ~` navega até o diretório home do usuário logado, por exemplo, /usr/home/jack. Experimente o comando `cd /cdrom`, e em seguida `ls`, para descobrir se o seu CDROM está montado e funcionando.

`less filename`
: Permite visualizar o conteúdo de um arquivo (chamado *filename*) sem modificá-lo. Experimente o comando `less /etc/fstab`. Digite `q` para sair do modo de visualização.

`cat filename`

Exibe o conteúdo do arquivo *filename* na tela. Se o arquivo for muito longo e você conseguir visualizar apenas o final, pressione `ScrollLock` e use a tecla de seta para cima (`up-arrow`) para rolar para trás. Você também pode usar `ScrollLock` com páginas de manual (man pages). Pressione `ScrollLock` novamente para sair do modo de rolagem. Você pode experimentar o comando `cat` em alguns dos arquivos ocultos em seu diretório home, por exemplo: `cat .cshrc`, `cat .login`, `cat .profile`.

Você notará que existem aliases no arquivo .cshrc para alguns dos comandos `ls` (eles são muito convenientes). Você pode criar outros aliases editando o arquivo .cshrc. Você pode tornar esses aliases disponíveis para todos os usuários do sistema colocando-os no arquivo de configuração global do `csh`, o /etc/csh.cshrc.

4. Obtendo ajuda e informações

Aqui estão algumas fontes úteis de ajuda. O termo *Text* significa algo de sua escolha que você digita, geralmente um comando ou nome de arquivo.

`apropos text`

Irá retornar todos os comandos e arquivos que contenham a palavra-chave especificada *text* no `banco de dados whatis`.

`man text`

A página de manual para *text*. A principal fonte de documentação para sistemas UNIX®. `man ls` lhe dirá todas as maneiras de usar o comando `ls`. Pressione `Enter` para mover pelo texto, `Ctrl` + `B` para voltar uma página, `Ctrl` + `F` para avançar, `q` ou `Ctrl` + `C` para sair .

`which text`

Mostra o caminho completo para o executável do comando *text* especificado.

`locate text`

Todos os caminhos onde a string *text* é encontrada.

`whatis text`

Informa o que o comando *text* faz e sua página de manual. Digitando `whatis *` irá falar sobre todos os binários no diretório atual.

`whereis text`

Localiza o arquivo *text*, fornecendo seu caminho completo.

Você pode tentar usar `whatis` em alguns comandos úteis comuns como `cat`, `more`, `grep`, `mv`, `find`, `tar`, `chmod`, `chown`, `date` , e `script`. `more` permite que você leia uma página por vez como no DOS, por exemplo, `ls -l | more` ou `more filename`. O * funciona como um curinga, por 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 sua máquina não vai ficar ligada no fim de semana (e rodando o FreeBSD), você pode querer executar os comandos para manutenção diária,

semanal e mensal de vez em quando. Execute-os como `root` e, por enquanto, dê a cada um tempo para terminar antes de iniciar o próximo.

```
# periodic daily
saída omitida
# periodic weekly
saída omitida
# periodic monthly
saída omitida
```

Se você cansar de esperar, pressione `Alt` + `F2` para obter outro *virtual console* e faça login novamente. Afinal, é um sistema multiusuário e multitarefa. No entanto, esses comandos provavelmente exibirão mensagens em sua tela enquanto estiverem sendo executados; você pode digitar `clear` no prompt para limpar a tela. Uma vez executados, você pode querer olhar para /var/mail/root e /var/log/messages.

Executar esses comandos faz parte da administração do sistema - e como único usuário de um sistema UNIX®, você é seu próprio administrador de sistema. Praticamente tudo o que você precisa fazer como `root` é administração do sistema. Essas responsabilidades não são bem cobertas, mesmo nos grandes e grossos livros sobre UNIX®, que parecem dedicar muito espaço a abrir menus em gerenciadores de janelas. Se você deseja aprofundar seus conhecimentos em administração de sistemas, pode ser útil obter um dos dois principais livros sobre o assunto. Os dois livros recomendados são "UNIX System Administration Handbook" de Evi Nemeth et al. (Prentice-Hall, 1995, ISBN 0-13-15051-7) - a segunda edição com a capa vermelha; ou "Essential System Administration" de Æleen Frisch (O'Reilly & Associates, 2002, ISBN 0-596-00343-9). Ambos os livros são excelentes recursos para administradores de sistemas e fornecem informações detalhadas sobre administração de sistemas UNIX®. Eu usei o livro da Nemeth.

5. Editando textos

Para configurar o sistema, você precisa editar arquivos de texto. A maioria deles estará no diretório /etc, e você precisará usar o comando `su` para se tornar `root` e poder alterá-los. Você pode usar o editor de texto `ee` para edição mais simples, mas a longo prazo, vale a pena aprender o editor de texto `vi`. Há um excelente tutorial sobre o vi em /usr/src/contrib/nvi/docs/tutorial, se você tiver as fontes do sistema instaladas.

Antes de editar um arquivo, é recomendável fazer backup dele. Suponha que você deseje editar o arquivo /etc/rc.conf. Você pode simplesmente usar o comando `cd /etc` para ir para o diretório /etc e executar o seguinte comando:

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

Isso copiaria o arquivo rc.conf para rc.conf.orig, e posteriormente você poderia copiar rc.conf.orig de volta para rc.conf para recuperar o original. Mas uma opção ainda melhor 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 e o proprietário original do arquivo. Agora você pode editar o arquivo rc.conf. Se quiser recuperar o original, você pode executar o comando `mv rc.conf rc.conf.myedit` (assumindo que você deseja preservar a versão editada) e depois executar o seguinte comando

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

para colocar as coisas de volta da maneira que estavam.

Para editar um arquivo, digite

```
# vi filename
```

Para mover-se pelo texto, use as teclas de seta. Pressione `Esc` (a tecla de escape) para entrar no modo de comando do `vi`. Aqui estão alguns comandos:

`x`
: deleta a letra localizada onde o cursor está

`dd`
: Apaga a linha inteira (mesmo se ela estiver aparecendo quebrada na tela em varias linhas)

`i`
: Insere um texto na posição do cursor

`a`
: Insere um texto após a posição do cursor

Depois de digitar `i` ou `a`, você pode inserir texto. `Esc` coloca você de volta no modo de comando onde você pode digitar

`:w`
: para gravar suas alterações no disco e continuar editando

`:wq`
: para gravar e sair

`:q!`
: para sair do arquivo sem salvar as alterações

`/text`
: para mover o cursor para *text*; `/` `Enter` (a tecla enter) para encontrar a próxima ocorrência de *text*.

`G`

para ir para o final do arquivo

`nG`

para ir para a linha *n* no arquivo, onde *n* é um número

`Ctrl` + `L`

para redesenhar a tela

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

para voltar e avançar uma tela, como fazem o `more` e o `view`.

Pratique com `vi` no seu diretório pessoal, criando um novo arquivo com o comando `vi nome_do_arquivo`, adicionando e excluindo texto, salvando o arquivo e chamando-o novamente. O `vi` oferece algumas surpresas porque é realmente bastante complexo e às vezes você inadvertidamente emite um comando que fará algo que você não espera. (Algumas pessoas realmente gostam do `vi` - é mais poderoso que o EDIT do DOS - descubra sobre o `:r`.) Use `Esc` uma ou mais vezes para ter certeza de que está no modo de comando e prossiga a partir daí quando ele lhe der problemas, salve frequentemente com `:w` e use `:q!` para sair e começar de novo (a partir do último `:w`) quando precisar.

Agora você pode fazer `cd` para /etc, `su` para `root`, usar `vi` para editar o arquivo /etc/group e adicionar um usuário ao grupo `wheel` para que ele tenha privilégios de root. Basta adicionar uma vírgula e o nome de login do usuário ao final da primeira linha do arquivo, pressionar `Esc` e usar `:wq` para gravar o arquivo 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. ps ax é uma forma mais simplificada.

`rm filename`

remove o arquivo *filename*.

`rm -R dir`

remove um diretório *dir* e todos os subdiretórios - use com cuidado!

`ls -R`

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

`passwd`

para alterar a senha do usuário (ou a senha do `root`)

`man hier`

página de manual no sistema de arquivos UNIX®

Use o `find` para localizar o arquivo filename em /usr ou qualquer um de seus subdiretórios com

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

Você pode usar * como curinga em `"filename"` (que deve estar entre aspas). Se você disser ao `find` para procurar em / ao invés de /usr ele irá procurar o(s) arquivo(s) em todos os sistemas de arquivos montados, incluindo o CD-ROM e a partição DOS.

Um livro excelente que explica os comandos e utilitários do UNIX® é Abrahams & Larson, Unix for the Impacient (2ª ed., Addison-Wesley, 1996). Também há muitas informações sobre o UNIX® na Internet.

7. Próximos Passos

Você agora deve ter as ferramentas necessárias para navegar e editar arquivos, para que possa colocar tudo em funcionamento. Existe uma grande quantidade de informações no Handbook do FreeBSD (que provavelmente está no seu disco rígido) e no site do FreeBSD. Uma ampla variedade de pacotes e ports estão no CDROM, bem como no site. O Handbook explica mais sobre como usá-los (obtenha o pacote se existir, com `pkg add nomedopacote`, onde nomedopacote é o nome do arquivo do pacote). O CDROM contém listas dos pacotes e ports com breves descrições em cdrom/packages/index, cdrom/packages/index.txt, e cdrom/ports/index, e com descrições mais detalhadas 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 `lndir` 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 adiciona no sistema e que devem estar disponíveis para todos os usuários) com:

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

Isso deve resultar em um subdiretório /usr/local/kermit que contém todos os arquivos que o subdiretório `kermit` no CD-ROM possui.

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

Então utilize o `cd` para ir para o subdiretório /usr/local/kermit que contém o arquivo Makefile. E execute

```
# make all install
```

Durante esse processo, o port fará o download via FTP de quaisquer arquivos compactados que precise e que não foram encontrados no CDROM ou no diretório /usr/ports/distfiles. Se você não tiver a rede funcionando ainda e não houver um arquivo para o port em /cdrom/ports/distfiles, você precisará obter o distfile usando outra máquina e copiá-lo para /usr/ports/distfiles. Leia o Makefile (com `cat` ou `more` ou `view`) para descobrir onde obter o arquivo (o site de distribuição principal) e qual é o nome dele. (Use transferências de arquivo binário!) Em seguida, volte para /usr/local/kermit, encontre o diretório com o Makefile e execute o `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 vêm instalados com o FreeBSD: `csh` e `sh`. O `csh` é bom para trabalho na linha de comando, mas os scripts devem ser escritos com `sh` (ou `bash`). Você pode descobrir qual shell você tem digitando `echo $SHELL`.

O shell `csh` é bom, mas o `tcsh` faz tudo o que o `csh` faz e muito mais. Ele permite que você recupere comandos com as teclas de seta e edite-os. Ele completa os nomes dos arquivos com a tecla de tabulação (o `csh` usa `Esc`) e permite alternar para o último diretório em que você estava com `cd -`. Também é muito mais fácil alterar seu prompt com `tcsh`. Ele torna a vida muito mais fácil.

Aqui estão os três 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 `chsh` para alterar seu shell para `tcsh` permanentemente, ou digite `tcsh` no prompt para alterar seu shell sem fazer login novamente.



Pode ser perigoso mudar o shell `root` para algo diferente de `sh` ou `csh` nas primeiras versões do FreeBSD e muitas outras versões do UNIX®; você pode não ter um shell funcionando quando o sistema o coloca no modo de usuário único. A solução é usar `su -m` para se tornar `root`, o que lhe dará o `tcsh` como `root`, porque o shell faz parte do ambiente. Você pode torná-lo permanente adicionando-o ao seu arquivo .tcshrc como um alias com:

```
alias su su -m
```

Quando o `tcsh` iniciar, ele lerá os arquivos /etc/csh.cshrc e /etc/csh.login, assim como o `csh`. Ele também lerá .login em seu diretório inicial e bem como o .cshrc, a menos que você forneça um .tcshrc. Isso pode ser feito simplesmente copiando .cshrc para .tcshrc.

Agora que você instalou o `tcsh`, você pode ajustar seu prompt. Você pode encontrar os detalhes na página de manual do `tcsh`, mas aqui está uma linha para colocar no seu .tcshrc que lhe dirá quantos comandos você digitou, que horas são e em qual diretório você está. Ele também produz um `>` se você for um usuário comum e um # se você for `root`, mas o tsch fará isso em qualquer caso:

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

Isso deve ir no mesmo lugar que a linha de configuração do prompt existente, se houver, ou sob "if($?prompt) then" se não. Comente a linha antiga; você sempre pode voltar para ele, se preferir. Não se esqueça dos espaços e aspas. Você pode reaplicar as opções do .tcshrc digitando `source .tcshrc`.

Você pode obter uma lista das outras variáveis de ambiente que foram definidas digitando `env` no prompt. O resultado mostrará seu editor, pager e tipo de terminal padrão, entre possivelmente muitos outros. Um comando útil se você efetuar login de um local remoto e não puder executar um programa porque o terminal não é capaz é `setenv TERM vt100`.

9. Outros

Como `root`, você pode desmontar o CDROM com `/sbin/umount /cdrom`, retirá-lo da unidade, inserir outro e montá-lo com `/sbin/mount_cd9660 /dev/cd0a /cdrom` assumindo que cd0a é o nome do dispositivo para sua unidade de CD-ROM. As versões mais recentes do FreeBSD permitem que você monte o CDROM apenas com `/sbin/mount /cdrom`.

Usar o sistema de arquivos ao vivo (live filesystem), o segundo dos discos do CDROM do FreeBSD, é útil se você tem espaço limitado. O que está no sistema de arquivos ao vivo varia de versão para versão. Você pode tentar jogar jogos 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, pois eles estão em /cdrom em vez de em /usr e seus subdiretórios, onde são esperados. Leia `man lndir`.

10. Comentários são bem-vindos

Se você usou este guia, ficaria interessado em saber onde não ficou claro e o que foi deixado de fora que você acha que deveria ser incluído, e se foi útil. Meus agradecimentos a Eugene W. Stark, professor de ciência da computação da SUNY-Stony Brook, e John Fieber por seus comentários úteis.

Annelise Anderson, andrsn@andrsn.stanford.edu