

Guia rápido de FreeBSD para usuários de Linux®

John Ferrell

Revision: [52220](#)

Copyright © 2008 Projeto de Documentação do FreeBSD

FreeBSD is a registered trademark of the FreeBSD Foundation.

Linux is a registered trademark of Linus Torvalds.

Intel, Celeron, Centrino, Core, EtherExpress, i386, i486, Itanium, Pentium, and Xeon are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Red Hat, RPM, are trademarks or registered trademarks of Red Hat, Inc. in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this document, and the FreeBSD Project was aware of the trademark claim, the designations have been followed by the “™” or the “®” symbol.

2018-09-06 00:43:08 by ebrandi.

Resumo

Este documento tem a intenção de familiarizar rapidamente usuários intermediários ou avançados do Linux® com o básico do FreeBSD.

Índice

1. Introdução	1
2. Shell Padrão	2
3. Coleção de Pacotes e Árvore de Ports: Instalando novos programas no FreeBSD	2
4. Inicialização do Sistema	3
5. Configuração de Rede	4
6. Firewall	4
7. Atualizando o FreeBSD	5
8. procfs: É passado, mas foi não esquecido	5
9. Comandos Comuns	6
10. Conclusão	6

1. Introdução

Esse documento destaca algumas diferenças técnicas entre o FreeBSD e o Linux® para que os usuários intermediários ou avançados do Linux® possam se familiarizar rapidamente com o básico do FreeBSD.

Este documento assume que o FreeBSD já está instalado. Acesse o link do capítulo [Instalando o FreeBSD](#) no Handbook do FreeBSD para obter ajuda no processo de instalação.

2. Shell Padrão

Os usuários do Linux® são geralmente surpreendidos quando descobrem que o Bash não é o shell padrão do FreeBSD. De fato, o Bash não é incluído na instalação padrão. Ao invés disto, o FreeBSD usa o `tcsh(1)` como shell padrão para o usuário root, e o `sh(1)`, um shell compatível com o Bourne shell, como shell padrão para os demais usuários. O `sh(1)` é muito similar ao Bash mas com um conjunto de funcionalidades muito menor. Geralmente os scripts shell escritos para o `sh(1)` irão ser executados no Bash, mas o contrário não.

Entretanto, o Bash e outros shells estão disponíveis para a instalação usando a [Coleção de Pacotes e a Árvore de Ports](#).

Depois de instalar um novo shell, use o comando `chsh(1)` para trocar o shell padrão do usuário. É recomendado que o shell padrão do usuário root se mantenha inalterado uma vez que os shells que não fazem parte da base do sistema são instalados em `/usr/local/bin`. No caso de ocorrer algum problema com o sistema de arquivos onde o diretório `/usr/local/bin` está localizado e este não puder ser utilizado, o usuário root poderá não ter acesso ao shell padrão, evitando que o usuário root entre no sistema para corrigir o problema.

3. Coleção de Pacotes e Árvore de Ports: Instalando novos programas no FreeBSD

FreeBSD provê dois métodos para a instalação de novos aplicativos: pacotes binários e através da compilação do código fonte (Árvore de ports). Cada método tem seu benefício:

- Instalação rápida quando comparada com a compilação de grandes aplicativos .
- Não há a necessidade de compreender como compilar um software.
- Não é necessário a instalação de um compilador.
- Possibilidade de customizar as opções de instalação
- Possibilidade de se aplicar patches customizados

Se a instalação da aplicação não necessitar de opções customizadas, a instalação via pacotes é suficiente. Compile o port sempre que o aplicativo exigir a personalização das opções padrão. Se você necessita de pacotes customizados, você poderá obtê-los através da compilação a partir do ports utilizando o comando `make` Pacotes .

A lista completa da árvore de ports e dos pacotes pode ser encontrada [aqui](#).

3.1. Pacotes

Pacotes são aplicações pré-compiladas, os equivalentes no FreeBSD aos arquivos `.deb` nos sistemas baseados no Debian/Ubuntu e aos arquivos `.rpm` nos sistemas baseados no Fedora/Red Hat. Pacotes são instalados através do comando `pkg`. Por exemplo, o comando a seguir instala o Apache 2.4:

```
# pkg install apache24
```

Para mais informações sobre pacotes, veja a seção 5.4 do Handbook do FreeBSD: [Usando pkgng para gerenciar pacotes binários](#).

3.2. Árvore de Ports

A árvore de ports do FreeBSD é um framework de `Makefiles` e patches customizados especificamente para a instalação através do código fonte no FreeBSD. Quando um port é instalado, o sistema irá buscar o código fonte, aplicar qualquer patch que seja necessário, compilar o código, instalar a aplicação e qualquer outra dependência da qual ela necessite.

A Coleção de Ports, algumas vezes referenciada como a árvore de ports, pode ser instalada em `/usr/ports` usando o comando `portsnap(8)`. Instruções detalhadas para instalação da Coleção de Ports podem ser encontradas na [seção 5.5](#) do Handbook do FreeBSD.

Para compilar um port, mude para o diretório do port e inicie o processo de compilação. O exemplo abaixo instala o Apache 2.4 através da Coleção de Ports:

```
# cd /usr/ports/www/apache24
# make install clean
```

Um dos benefícios de usar a árvore de ports para realizar a instalação de um software é a possibilidade de customizar as opções de instalação. O exemplo a seguir, especifica que o módulo `mod_ldap` também deve ser instalado:

```
# cd /usr/ports/www/apache24
# make WITH_LDAP="YES" install clean
```

Veja [Usando a Coleção de Ports](#) para mais informações.

4. Inicialização do Sistema

Muitas distribuições Linux® usam o sistema `init` do SysV, enquanto o FreeBSD usa o tradicional `init(8)` estilo BSD. Por utilizar este sistema `init(8)`, não existe níveis diferenciados de execução e o arquivo `/etc/inittab` não existe. Ao invés disto, a inicialização é controlada por meio de scripts `rc(8)`. Na inicialização do sistema, o `/etc/rc` lê o arquivo `/etc/rc.conf` e o arquivo `/etc/defaults/rc.conf` para determinar quais são os serviços que devem ser inicializados. Os serviços especificados são inicializados pela execução de scripts de inicialização localizados em `/etc/rc.d/` e `/usr/local/etc/rc.d/`. Esses scripts são similares aos contidos no diretório `/etc/init.d/` dos sistemas Linux®.

Os scripts encontrados no diretório `/etc/rc.d/` fazem parte das aplicações da “base” do sistema, tais como `cron(8)`, `sshd(8)`, e `syslog(3)`. Os scripts encontrados no diretório `/usr/local/etc/rc.d/` correspondem aos aplicativos instalados pelo usuário, como por exemplo: Apache e Squid.

Uma vez que o FreeBSD é desenvolvido como um sistema completo, aplicações instaladas pelos usuários não são consideradas parte do sistema “base”. As aplicações dos usuários são geralmente instaladas por meio [dos Pacotes Binários ou da Coleção de Ports](#). Para mantê-los separados da base do sistema, eles são instalados sob o diretório `/usr/local/`. Portanto, os aplicativos binários instalados pelos usuários localizam-se em `/usr/local/bin/`, e os arquivos de configuração em `/usr/local/etc/`.

Os serviços são habilitados pela adição de uma entrada no arquivo `/etc/rc.conf`. As configurações padrões são encontradas no arquivo `/etc/defaults/rc.conf` e essas configurações padrões são sobre postas pelas configurações realizadas no arquivo `/etc/rc.conf`. Veja o manual do [rc.conf\(5\)](#) para maiores informações sobre as entradas disponíveis. Quando você instalar aplicações adicionais, leia as mensagens de instalação da aplicação para determinar como habilitar os serviços associados a essa aplicação.

As seguintes entradas no arquivo `/etc/rc.conf` habilitam o `sshd(8)`, o Apache 2.4, e especifica que o Apache deve ser inicializado com SSL ativado.

```
# enable SSHD
sshd_enable="YES"
# enable Apache with SSL
apache24_enable="YES"
apache24_flags="-DSSL"
```

Uma vez que o serviço tenha sido habilitado no arquivo `/etc/rc.conf`, ele pode ser inicializado sem a necessidade de uma reinicialização do sistema.

```
# service sshd start
# service apache24 start
```

Se o serviço não tiver sido habilitado, ele poderá ser inicializado a partir da linha de comando usando a opção `onestart`:

```
# service sshd onestart
```

5. Configuração de Rede

Diferente da identificação genérica *ethX* usada pelo Linux® para identificar a interface de rede, o FreeBSD usa o nome do driver seguido por um número. A seguinte saída do comando `ifconfig(8)` mostra duas interfaces de rede Intel® Pro 1000 (`em0` e `em1`):

```
% ifconfig
em0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    options=b<RXCSUM, TXCSUM, VLAN_MTU>
    inet 10.10.10.100 netmask 0xfffff00 broadcast 10.10.10.255
    ether 00:50:56:a7:70:b2
    media: Ethernet autoselect (1000baseTX <full-duplex>)
    status: active
em1: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    options=b<RXCSUM, TXCSUM, VLAN_MTU>
    inet 192.168.10.222 netmask 0xfffff00 broadcast 192.168.10.255
    ether 00:50:56:a7:03:2b
    media: Ethernet autoselect (1000baseTX <full-duplex>)
    status: active
```

Um endereço IP pode ser designado à uma interface utilizando o comando `ifconfig(8)`. Para torna-lo definitivo e persistente entre as reinicializações, o endereço IP deve ser incluído no arquivo `/etc/rc.conf`. A seguinte entrada no arquivo `/etc/rc.conf` especifica o nome da máquina, o endereço IP e o gateway padrão da rede:

```
hostname="server1.example.com"
ifconfig_em0="inet 10.10.10.100 netmask 255.255.255.0"
defaultrouter="10.10.10.1"
```

Use a seguinte entrada para configurar uma interface para obter sua configuração por meio do DHCP:

```
hostname="server1.example.com"
ifconfig_em0="DHCP"
```

6. Firewall

O sistema FreeBSD não utiliza o Linux® IPTABLES como seu firewall, o FreeBSD oferece três tipos de firewall a nível de kernel:

- [PF](#)
- [IPFILTER](#)
- [IPFW](#)

O PF é desenvolvido pelo projeto OpenBSD e portado para o FreeBSD. O PF foi criado para substituir o IPFILTER e sua sintaxe é similar ao IPFILTER. O PF pode ser utilizado em conjunto com `altq(4)` para prover funcionalidade de QoS.

O exemplo abaixo mostra uma regra do PF para permitir conexões de entrada do SSH:

```
pass in on $ext_if inet proto tcp from any to ($ext_if) port 22
```

O IPFILTER é uma aplicação de firewall desenvolvida por Darren Reed. Ela não é específica para o FreeBSD e foi portado para diversos sistemas operacionais, incluindo NetBSD, OpenBSD, SunOS, HP/UX, e Solaris.

A sintaxe do IPFILTER para permitir conexões de entrada do SSH é:

```
pass in on $ext_if proto tcp from any to any port = 22
```

O IPFW é o firewall desenvolvido e mantido pelo FreeBSD. Ele pode ser utilizado em conjunto com o [dummynet\(4\)](#) para prover a funcionalidade de traffic shaping e simular diferentes tipos de conexões de rede.

A sintaxe do IPFW para permitir conexões de entrada do SSH é:

```
ipfw add allow tcp from any to me 22 in via $ext_if
```

7. Atualizando o FreeBSD

Existem dois métodos para realizar a atualização em um sistema FreeBSD: a partir do código fonte ou atualização binária.

Atualizar através do código fonte é a forma mais trabalhosa, mas ela oferece uma grande flexibilidade. O processo envolve a sincronização da cópia local do código fonte do FreeBSD com os servidores de Subversion do FreeBSD. Uma vez que o código fonte local esteja atualizado, uma nova versão do kernel e da userland poderão ser compiladas.

Atualização binária é similar ao uso do comando `yum` ou `apt-get` para atualizar um sistema Linux®. No FreeBSD, o comando [freebsd-update\(8\)](#) pode ser utilizado para buscar uma nova atualização binária e a instalá-la. Estas atualizações podem ser agendada usando o [cron\(8\)](#).



Nota

Quando utilizar o [cron\(8\)](#) para agendar as atualizações, use o comando `freebsd-update cron` no arquivo [crontab\(1\)](#) para reduzir a possibilidade de que um grande número de máquinas busquem a atualização ao mesmo tempo:

```
0 3 * * * root /usr/sbin/freebsd-update cron
```

Para maiores informações sobre a atualização por meio do código fonte ou dos updates binários, acesse o [capítulo sobre atualização](#) do Handbook do FreeBSD .

8. procfs: É passado, mas foi não esquecido

Em algumas distribuições do Linux®, você pode consultar o `/proc/sys/net/ipv4/ip_forward` para verificar se o encaminhamento de IP está ou não habilitado. No FreeBSD, o comando [sysctl\(8\)](#) é utilizado para ver o status desta e também de outras variáveis do sistema.

Por exemplo, use o seguinte comando para determinar se o encaminhamento de IP está ou não habilitado.

```
% sysctl net.inet.ip.forwarding
net.inet.ip.forwarding: 0
```

Use a opção `-a` para ver todas as variáveis do sistema:

```
% sysctl -a | more
```

Se alguma aplicação necessitar do procfs, adicione a seguinte entrada no arquivo `/etc/fstab` :

```
proc          /proc          procfs        rw,noauto     0             0
```

Incluindo a opção `noauto` irá prevenir que o `/proc` seja montado automaticamente durante a inicialização do sistema.

Para montar o sistema de arquivos sem reinicializar:

```
# mount /proc
```

9. Comandos Comuns

Alguns comandos comuns e equivalentes são os seguintes:

Comandos do Linux® (Red Hat/Debian)	Equivalente no FreeBSD	Propósito
<code>yum install package / apt-get install package</code>	<code>pkg install package</code>	Instalação de um pacote binário de um repositório remoto
<code>rpm -ivh package / dpkg -i package</code>	<code>pkg add package</code>	Instalação de um pacote local.
<code>rpm -qa / dpkg -l</code>	<code>pkg info</code>	Listar os pacotes instalados (Pacotes binários e através da árvore de ports)
<code>lspci</code>	<code>pciconf</code>	Lista os dispositivos PCI
<code>lsmod</code>	<code>kldstat</code>	Lista os módulos do kernel que foram carregados
<code>modprobe</code>	<code>kldload / kldunload</code>	Carrega/Descarrega módulos do kernel.
<code>strace</code>	<code>truss</code>	Rastreia chamadas do sistema

10. Conclusão

Este documento forneceu uma visão geral do FreeBSD. Veja o [Handbook do FreeBSD](#) para uma cobertura mais profunda desses tópicos, assim como outros não cobertos por este documento.