Guia Rápido do FreeBSD para Usuários Linux®

John Ferrell

Revisão: 43184

Copyright © 2008 The FreeBSD Documentation Project

FreeBSD is a registered trademark of the FreeBSD Foundation.

Linux is a registered trademark of Linus Torvalds.

Intel, Celeron, 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.

2013-11-13 por hrs.

Resumo

O objetivo deste documento é familiarizar rapidamente os usuários intermediários e avançados de Linux® com o FreeBSD.

Índice

1. Introdução 2
2. `Shells`: Sem Bash? 2
3. Pacotes e `Ports`: Adicionando programas no FreeBSD 2
4. Inicialização do Sistema: Onde estão os *run-levels*? 5

5. Configuração da rede 6
6. Firewall 7
7. Atualizando o FreeBSD 7
8. procfs: Morto, mas vivo na memória 8
9. Comandos Comuns 9
10. Conclusão 10

1. Introdução

Este documento irá destacar as diferenças entre FreeBSD e Linux® para que os usuários intermediários e avançados possam rapidamente se familiarizar com os conceitos básicos do FreeBSD. Esta é apenas uma rápida introdução técnica, ela não tenta discutir as diferenças "filosóficas" entre os dois sistemas operacionais.

Este documento assume que você já tem o FreeBSD instalado. Se você não tem o FreeBSD instalado ou precisa de ajuda com o processo de instalação, por favor, consulte o capítulo Instalando o FreeBSD no Handbook.

2. Shells: Sem Bash?

Usuários vindos do Linux® são frequentemente surpreendidos por não encontrarem o Bash como o `shell` padrão no FreeBSD. De fato, o Bash nem mesmo está presente na instalação padrão. Em vez disso, o FreeBSD usa o tcsh(1) como `shell` padrão. Embora o Bash e seus outros `shells` favoritos estejam disponíveis na Coleção de `Ports` do FreeBSD.

Se você instalar outros `shells`, o chsh(1) poderá ser usado para definir o `shell` padrão dos usuários. Contudo, é recomendável que o `shell` padrão do `root` permaneça inalterado. A razão para isso é que `shells` não incluídos na base do sistema são normalmente instalados em `/usr/local/bin` ou `/usr/bin`. Caso ocorra um problema no sistema de arquivos no qual estão localizados o `/usr/local/bin` e o `/usr/bin`, eles não poderão ser montados. Neste caso, o usuário `root` não teria acesso ao seu `shell` padrão, o que o impediria de efetuar login. Por este motivo uma segunda conta `root`, a conta `toor`, foi criada para uso com `shells` que não fazem parte da base do sistema. Leia o `FAQ` de segurança para obter informações sobre a conta toor.

3. Pacotes e Ports: Adicionando programas no FreeBSD

Além do tradicional método UNIX® de instalação de programas (baixar o código fonte, extrair, editar o código fonte, e compilar), o FreeBSD oferece dois outros métodos para instalar aplicações: pacotes e `ports`. Uma lista completa de todos os `ports` e pacotes disponíveis pode ser encontrada aqui.

3.1. Pacotes

Pacotes são aplicações pré-compiladas, o equivalente no FreeBSD ao `.deb` nos sistemas baseados no Debian/Ubuntu e ao `.rpm` nos sistemas baseados no Red Hat/Fedora. Pacotes são instalados usando pkg_add(1). Por exemplo, o seguinte comando instala o Apache 2.2:

```
# pkg_add -/tmp/apache-2.2.6_2.tbz
```

Usar a opção `-r` dirá ao pkg_add(1) para baixar automaticamente o pacote e instalá-lo, juntamente com quaisquer dependências que ele possua:

```
# pkg_add --r apache22
Fetching ftp://ftp.freebsd.org/pub/FreeBSD/ports/i386/packages-6.2-
release/Latest/apache22.tbz... Done.
Fetching ftp://ftp.freebsd.org/pub/FreeBSD/ports/i386/packages-6.2-
release/All/expat-2.0.0_1.tbz... Done.
Fetching ftp://ftp.freebsd.org/pub/FreeBSD/ports/i386/packages-6.2-
release/All/perl-5.8.8_1.tbz... Done.
[snip]

To run apache www server from startup, add apache22_enable="YES"
in your -/etc/rc.conf. Extra options can be found in startup script.
```

Nota

Se você está rodando uma versão de `release` do FreeBSD (6.2, 6.3, 7.0, etc., geralmente instalada a partir de um CD-ROM) o `pkg_add -r` vai baixar o pacote compilado especificamente para esta versão. Este pacote *pode não* ser a versão mais atual da aplicação. Você pode usar a variável `PACKAGESITE` para sobrescrever este comportamento padrão. Por exemplo, ajuste `PACKAGESITE` para `ftp://ftp.freebsd.org/pub/FreeBSD/ports/i386/packages-6-stable/Latest/` para baixar os pacotes mais recentes compilados para a série 6.X.

Para mais informações sobre pacotes, por favor, consulte a seção 4.4 do Handbook do FreeBSD: Usando o Sistema de Pacotes.

3.2. `Ports`

O segundo método para instalação de aplicações no FreeBSD é a Coleção de `Ports`. A Coleção de `Ports` é um *framework* de `Makefiles` e *patches* especialmente customizados para a instalação de vários programas a partir do código fonte no FreeBSD. Ao instalar um `port` o sistema irá baixar o código fonte, aplicar qualquer *patch* necessário, compilar

o código, e instalar a aplicação. O mesmo processo será aplicado para todas as suas dependências.

A Coleção de Ports, por vezes designada como a árvore de ports, pode ser encontrada em /usr/ports . Isto assumindo que a Coleção de Ports foi instalada durante o processo de instalação do FreeBSD. Se a Coleção de Ports não foi instalada, ela pode ser adicionada a partir dos discos de instalação usando sysinstall(8), ou baixada dos servidores do FreeBSD usando csup(1) ou portsnap(8). Instruções detalhadas para a instalação da Coleção de Ports podem ser encontradas na seção 4.5.1 do Handbook.

A instalação de um port é tão simples (geralmente) quanto entrar no diretório do port desejado e iniciar o processo de compilação. O exemplo seguinte instala o Apache 2.2 a partir da Coleção de Ports:

```
# cd -/usr/ports/www/apache22
# make install clean
```

Um grande benefício do uso do ports para instalar programas é a possibilidade de personalizar as opções de instalação. Por exemplo, ao instalar o Apache 2.2 a partir do ports, você poderá habilitar o mod_ldap definindo a variável WITH_LDAP ao executar make(1):

```
# cd -/usr/ports/www/apache22
# make WITH_LDAP="YES" install clean
```

Por favor, leia a seção 4.5 do Handbook do FreeBSD, Usando a Coleção de Ports, para maiores informações sobre a Coleção de Ports.

3.3. Ports ou pacotes, qual eu devo usar?

Pacotes são apenas ports pré-compilados, então na prática é uma questão de instalarmos a partir do código fonte (ports) contra instalarmos de um pacote binário. Cada método tem seus próprios benefícios:

- Instalação rápida (a compilação de grandes aplicações pode ser um tanto demorada).
- Você não precisar saber como compilar o programa.
- Não é necessário instalar compiladores no seu sistema.
- Possibilidade de personalizar as opções de instalação. (Pacotes normalmente são compilados com as opções padrões. Com o ports você pode personalizar várias opções, como a compilação de módulos adicionais ou a mudança do *path* de instalação padrão.)
- Você pode aplicar seus próprios *patches* se assim desejar.

Se você não tem qualquer requisito especial, o sistema de pacotes provavelmente vai se adequar muito bem à sua situação. Se você for precisar personalizar a instalação, o ports é a melhor opção. (E lembre-se, se você precisa personalizar a instalação, mas prefere

pacotes, você pode compilar um pacote personalizado a partir do `ports` usando `make package` e, em seguida, copiar o pacote para outros servidores.)

4. Inicialização do Sistema: Onde estão os run-levels?

O Linux® usa o sistema `SysV init`, enquanto o FreeBSD usa o tradicional `BSD-style` init(8). Sob o `BSD-style` init(8) não existem *run-levels* e nem `/etc/inittab`, em vez disso a inicialização é controlada pelo utilitário rc(8). O `script /etc/rc` lê `/etc/defaults/rc.conf` e `/etc/rc.conf` para determinar quais serviços serão iniciados. Os serviços especificados são, então, inicializados rodando os `scripts` de inicialização correspondentes em `/etc/rc.d/` e `/usr/local/etc/rc.d/`. Esses `scripts` são similares aos `scripts` localizados em `/etc/init.d/` nos sistemas Linux®.

> *Por que existem dois locais para `scripts` de inicialização de serviços?* Os `scripts` encontrados em `/etc/rc.d/` são para aplicações que são parte da "base" do sistema. (cron(8), sshd(8), syslog(3), e outros.) Os `scripts` em `/usr/local/etc/rc.d/` são para aplicações instaladas pelo usuário, como Apache, Squid, etc.
>
> *Qual é a diferença entre a "base" do sistema e as aplicações instaladas pelo usuário?* O FreeBSD é desenvolvido como um sistema operacional completo. Em outras palavras, o `kernel`, bibliotecas do sistema, e utilitários de nível de usuário (como ls(1), cat(1), cp(1), etc.) são desenvolvidos juntos e lançados como um só. Isso é designado como "base" do sistema. As aplicações instaladas pelo usuário são aplicações que não fazem parte da "base" do sistema, como Apache, X11, Mozilla Firefox, etc. Estas aplicações instaladas pelo usuário são geralmente instaladas usando os Pacotes e a Coleção de `Ports`. A fim de mantê-las separadas da "base" do sistema, as aplicações dos usuário são normalmente instaladas sob `/usr/local/`. Portanto, os binários instalados pelo usuário residem em `/usr/local/bin/`, arquivos de configuração em `/usr/local/etc/`, e assim por diante.

Os Serviços são ativados espeficificando `NomeDoServiço_enable="YES"` em `/etc/rc.conf` (rc.conf(5)). Dê uma olhada em `/etc/defaults/rc.conf` para visualizar os padrões do sistema, essas configurações padrões podem ser sobrescritas por configurações em `/etc/rc.conf`. Quando instalar aplicações adicionais não deixe de analisar a documentação para determinar como ativar qualquer serviço associado.

O seguinte trecho do `/etc/rc.conf` ativa o sshd(8) e o Apache 2.2. Ele também determina que o Apache deve ser iniciado com SSL.

```
# enable SSHD
sshd_enable="YES"
# enable Apache with SSL
apache22_enable="YES"
apache22_flags="-DSSL"
```

Uma vez que o serviço foi ativado em `/etc/rc.conf` , ele pode ser inicializado pela linha de comando (sem precisar reinicializar o sistema):

```
# /etc/rc.d/sshd start
```

Se o serviço não foi ativado, ele pode ser inicializado pela linha de comando usando `forcestart` :

```
# /etc/rc.d/sshd forcestart
```

5. Configuração da rede

5.1. Interfaces de Rede

Em vez do identificador genérico *ethX*, que o Linux® usa para identificar uma interface de rede, o FreeBSD usa o nome do driver do dispositivo de rede seguido por um número como identificador. A seguinte saída do 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 0xffffff00 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 0xffffff00 broadcast ↺
192.168.10.255
        ether 00:50:56:a7:03:2b
        media: Ethernet autoselect (1000baseTX <full-duplex>)
        status: active
```

5.2. Configuração IP

Um endereço IP pode ser atribuído a uma interface de rede usando ifconfig(8). No entanto, para mantê-lo persistente entre as reinicializações, a configuração deve ser incluída em `/etc/rc.conf` . O seguinte exemplo configura o `hostname` , o endereço IP, e o `gateway` padrão:

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

Use a seguinte sintaxe para configurar a interface para DHCP:

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

6. `Firewall`

Como o IPTABLES no Linux®, o FreeBSD também oferece um `firewall` ao nível de `kernel`; atualmente o FreeBSD oferece três opções de `firewalls`:

- IPFIREWALL
- IPFILTER
- PF

O IPFIREWALL, ou IPFW (o comando para gerenciar um conjunto de regras IPFW é ipfw(8)), é o `firewall` desenvolvido e mantido pelos desenvolvedores do FreeBSD. O IPFW pode ser integrado com dummynet(4) para prover a capacidade de controle de tráfego e simular diferentes tipos de conexões de rede.

Amostra de uma regra do IPFW para permitir uma conexão de entrada do SSH:

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

IPFILTER é um aplicativo de `firewall` desenvolvido por Darren Reed. Ele não é específico para o FreeBSD e foi portado para vários sistemas operacionais, incluindo NetBSD, OpenBSD, SunOS, HP/UX, e Solaris.

Amostra do comando IPFILTER para permitir uma conexão de entrada do SSH:

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

O último aplicativo de `firewall`, PF, é desenvolvido pelo projeto OpenBSD. O PF foi criado como um substituto para o IPFILTER. Como tal, a sintaxe do PF é muito similar à do IPFILTER. O PF pode ser integrado com altq(4) para prover recursos de QoS.

Amostra do comando PF para permitir uma conexão de entrada do SSH:

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

7. Atualizando o FreeBSD

Existem três métodos para atualizar um sistema FreeBSD: a partir do código fonte, atualização binária, e a partir dos discos de instalação.

A atualização a partir do código fonte é a mais demorada, mas por outro lado é a que oferece a maior flexibilidade. O processo envolve a sincronização de uma cópia local do código fonte do sistema a partir dos servidores Subversion do FreeBSD. Uma vez que o código fonte local está atualizado, você pode compilar a nova versão do `kernel` e dos aplicativos de nível de usuário. Para maiores informações sobre atualizações a partir do código fonte veja o capítulo sobre atualização no Handbook do FreeBSD.

As atualizações binárias são similares ao uso do `yum` ou `apt-get` para atualizar sistemas Linux®. O comando freebsd-update(8) vai baixar e instalar as novas atualizações. As atualizações podem ser agendadas usando cron(8).

Nota

Se você utilizar o cron(8) para agendar as atualizações, por favor, certifique-se de usar `freebsd-update cron` em seu crontab(1) para reduzir a possibilidade de que um grande número de máquinas busquem as atualizações todas ao mesmo tempo.

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

O último método de atualização, a partir dos discos de instalação, é um processo bastante simples. Efetue o `boot` a partir dos discos de instalação e selecione a opção para atualizar.

8. procfs: Morto, mas vivo na memória

No Linux®, para determinar se o encaminhamento IP está ativado, você pode olhar em `/proc/sys/net/ipv4/ip_forward`. No FreeBSD você precisa usar o sysctl(8) para ver esta e outras opções do sistema, pois o procfs(5) tornou-se obsoleto nas versões mais recentes do FreeBSD. (Embora `sysctl` também esteja disponível no Linux®.)

No exemplo do encaminhamento IP, você poderia usar o seguinte comando para determinar se ele está ativado no seu sistema FreeBSD:

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

A opção `-a` é utilizada para listar todas as configurações do sistema:

```
% sysctl --a
kern.ostype: FreeBSD
kern.osrelease: 6.2-RELEASE-p9
kern.osrevision: 199506
kern.version: FreeBSD 6.2-RELEASE-p9 #0: Thu Nov 29 04:07:33 UTC 2007
    root@i386-builder.daemonology.net:/usr/obj/usr/src/sys/GENERIC

kern.maxvnodes: 17517
kern.maxproc: 1988
kern.maxfiles: 3976
kern.argmax: 262144
kern.securelevel: --1
kern.hostname: server1
kern.hostid: 0
```

```
kern.clockrate: { hz = 1000, tick = 1000, profhz = 666, stathz = ↺
133 -}
kern.posix1version: 200112
...
```

Nota

Alguns dos valores do `sysctl` estão disponíveis somente para leitura.

Existem ocasiões nas quais o `procfs` é necessário, como na execução de programas antigos, no uso do truss(1) para rastrear chamadas de sistema, e para possibilitar a Compatibilidade Binária com Linux. (Embora a Compatibilidade Binária com Linux use seu próprio `procfs`, linprocfs(5).) Se você precisar montar o `procfs`, você pode adicionar a seguinte entrada no `/etc/fstab`:

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

Nota

`noauto` vai prevenir `/proc` de ser montado automaticamente durante o `boot`.

E então monte o `procfs` com:

```
# mount -/proc
```

9. Comandos Comuns

9.1. Gerenciamento de Pacotes

Comando no Linux® (Red Hat/Debian)	Equivalente no FreeBSD	propósito
`yum install pacote` / `apt-get install pacote`	`pkg_add -r pacote`	Instala o *`pacote`* a partir do repositório remoto
`rpm -ivh pacote` / `dpkg -i pacote`	`pkg_add -v pacote`	Instala um pacote
`rpm -qa` / `dpkg -l`	`pkg_info`	Lista de pacotes instalados

9.2. Gerenciamento do Sistema

Comando no Linux®	Equivalente no FreeBSD	Propósito
`lspci`	`pciconf`	Lista de dispositivos PCI
`lsmod`	`kldstat`	Lista de módulos do `kernel` carregados
`modprobe`	`kldload` / `kldunload`	Carrega/descarrega módulos do `kernel`
`strace`	`truss`	Rastrear chamadas de sistema

10. Conclusão

Esperamos que este documento tenha fornecido para você o suficiente para começar a utilizar o FreeBSD. Certifique-se de dar uma olhada no Handbook do FreeBSD para se aprofundar nos tópicos abordados, assim como nos muitos tópicos não mencionados neste documento.