

Guía de FreeBSD para usuarios de Linux®

John Ferrell

Revisión: [32483e929e](#)

Copyright © 2008 El Proyecto de Documentación de FreeBSD

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 trademarks 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.

2019-03-29 15:24:26 +0000 por Gabor Kovcsan.

Resumen

El objetivo de este documento es familiarizar de forma rápida a usuarios de nivel intermedio o avanzado de Linux® sobre los conceptos básicos de FreeBSD.

Traducción de Gábor Kövesdán <gabor@FreeBSD.org>.

Tabla de contenidos

1. Introducción	1
2. Shells: ¿No está Bash?	2
3. Packages y Ports: Instalación de software en FreeBSD	2
4. Arranque del sistema: ¿Dónde están los niveles de ejecución?	3
5. Configuración de la red	4
6. Firewall	5
7. Actualizando FreeBSD	5
8. procs: Desaparecido pero no olvidado	6
9. Comandos comunes	6
10. Conclusión	7

1. Introducción

En este documento se muestran las diferencias entre FreeBSD y Linux® para que usuarios de nivel intermedio o avanzado de Linux® puedan aprender rápidamente los fundamentos de FreeBSD. Es solamente una breve presentación técnica, no intenta explicar las diferencias de las “filosofías” que hay detrás de ambos sistemas operativos.

Este documento presupone que usted ya tiene FreeBSD instalado. Si no lo ha instalado o necesita ayuda con el proceso de la instalación consulte el capítulo en el que se explica la [instalación de FreeBSD](#) del Handbook.

2. Shells: ¿No está Bash?

Quienes vienen de Linux® se suelen sorprender de que Bash no sea la shell por omisión, que en FreeBSD es [tcsh\(1\)](#). De hecho Bash ni siquiera forma parte de la instalación mínima de FreeBSD. A pesar de esto tanto Bash como cualquiera de sus shell favoritas están en la [Colección de Packages y Ports](#) de FreeBSD.

Si instala otras shell puede usar [chsh\(1\)](#) como shell por omisión para sus usuarios, pero le recomendamos encarecidamente que no cambie la shell por omisión del usuario root. La razón es que las shell que no forman parte de la instalación base generalmente se instalan en `/usr/local/bin` o en `/usr/bin`. En caso de problemas es posible que los sistemas de ficheros donde están `/usr/local/bin` y `/usr/bin` no puedan montarse. En un caso así root no podría acceder a su shell por omisión, con lo que no podría iniciar su sesión. Para evitar este tipo de problemas se creó una segunda cuenta de usuario root, el usuario `toor`. Consulte la sección de seguridad de las FAQ si quiere saber más sobre [la cuenta toor](#).

3. Packages y Ports: Instalación de software en FreeBSD

Aparte del método tradicional de instalar software en UNIX® (descargar fuentes, descomprimir, editar el fuente y compilar), FreeBSD ofrece dos alternativas para instalar programas: packages y dos alternativas para instalar programas: package y ports. Una lista y ports. Hay una lista completa de los ports disponibles [aquí](#).

3.1. Packages

Los “packages” son programas precompilados, equivalentes en FreeBSD a los ficheros `.deb` en sistemas Debian/Ubuntu y `.rpm` en sistemas Red Hat/Fedora. Los packages se instalan usando [pkg_add\(1\)](#). Por ejemplo, puede instalar Apache 2.2 del siguiente modo:

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

Al usar la opción `-r`, [pkg_add\(1\)](#) automáticamente descargará el package indicado y lo instalará, así como todos aquellos que necesite para su funcionamiento. Esto son las dependencias.

```
# 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

Si usa una versión release de FreeBSD (6.2, 6.3, 7.0, etc., lo que seguramente es una instalación desde CD-ROM) `pkg_add -r` descargará packages compilados para esa release específica. Estos packages *pueden ser* versiones desactualizadas del programa. Puede usar la variable `PACKAGESITE` para ignorar este comportamiento, que es el que se espera por omisión del programa. Por ejemplo, asigne a `PACKAGESITE` el valor `ftp://ftp.freebsd.org/pub/Free-`

BSD/ports/i386/packages-6-stable/Latest/ si quiere descargar los packages más recientes compilados para la versión 6.X.

Para más información sobre los packages consulte la sección 4.4 del Handbook: [Uso del sistema de packages](#).

3.2. Ports

El segundo método que ofrece FreeBSD para instalar programas es usar la Colección de Ports. la Colección de Ports es un entorno de ficheros Makefile y parches adaptados específicamente para poder instalar desde el código fuente aplicaciones de lo más variopinto en sistemas FreeBSD. Si instala un port el sistema descargará los fuentes, aplicará los parches necesarios, compilará el código e instalará el programa (y hará exactamente lo mismo con todas las dependencias).

La Colección de Ports, también conocida como árbol de ports, está en `/usr/ports` si se instaló durante el proceso de instalación de FreeBSD. Si la Colección de Ports no está instalada puede hacerlo desde los discos de instalación mediante [sysinstall\(8\)](#) o desde los servidores de FreeBSD gracias a [csup\(1\)](#) o [portsnap\(8\)](#). Tiene instrucciones detalladas para instalar la Colección de Ports en la [sección 4.5.1](#) del Handbook.

Instalar un port es (generalmente) tan simple como entrar en el directorio del port y ejecutar el proceso de compilación. El ejemplo que se muestra a continuación instala Apache 2.2 desde la Colección de Ports:

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

Una de las principales ventajas del uso de ports para instalar software es la posibilidad de ajustar las opciones de la instalación. Por ejemplo, es posible activar `mod_ldap` dentro de su Apache 2.2 durante la instalación mediante el uso de la variable `WITH_LDAP` de [make\(1\)](#):

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

Para más información sobre la Colección de Ports consulte la sección del Handbook dedicada a [la Colección de Ports](#).

3.3. ¿Qué debería usar? ¿Ports o Packages?

Los packages son simplemente ports precompilados, así que es cuestión de elegir entre instalar desde los fuentes (ports) o instalar binarios directamente (packages). Cualquiera de los dos tiene sus propias ventajas:

- Instalación más rápida (compilar programas grandes puede llevar su tiempo).
- No hay que comprender el proceso de compilar software.
- No es necesario instalar compiladores en su sistema.
- Es posible ajustar las opciones de la instalación. (Los packages se generan normalmente con las opciones estándar. Con los ports puede ajustar diversas opciones, como disponer de módulos adicionales o cambiar la ruta por omisión.)
- Puede usar sus propios parches si tiende hacerlo.

Si no tiene requisitos especiales los packages probablemente cubran sus necesidades sin problema. Si alguna vez necesita algo más ajustado puede recurrir a los ports. Tenga en cuenta que si necesita un programa a su medida pero sigue prefiriendo usar packages puede hacerse uno su gusto mediante `make patch` y luego copie el package a otras máquinas.

4. Arranque del sistema: ¿Dónde están los niveles de ejecución?

Linux® usa el sistema de inicio SysV, mientras que FreeBSD usa el sistema [init\(8\)](#) tradicional en BSD. En el [init\(8\)](#) del estilo BSD no existen ni los niveles de ejecución ni `/etc/inittab`; en vez de esto el inicio se controla desde

rc(8). El script `/etc/rc` lee los ficheros `/etc/defaults/rc.conf` y `/etc/rc.conf` para determinar qué servicios deben arrancarse. Después de esto se arrancan los servicios a los que se hace referencia en los scripts de arranque ubicados en `/etc/rc.d/` o `/usr/local/etc/rc.d/`. Estos scripts son similares a los que hay en `/etc/init.d/` en un sistema Linux®.

?Por qué los script de arranque están en dos sitios? Los script situados en `/etc/rc.d/` son para programas del sistema “base”, como [cron\(8\)](#), [sshd\(8\)](#), [syslog\(3\)](#) y otros. Los scripts de `/usr/local/etc/rc.d/` son para software instalado por el usuario como por ejemplo Apache, Squid, etc.

?Cuál es la diferencia entre el sistema “base” y los programas instalados por el usuario? FreeBSD se desarrolla como un sistema operativo completo. En otras palabras, el kernel, las bibliotecas del sistema y las utilidades (por ejemplo [ls\(1\)](#), [cat\(1\)](#), [cp\(1\)](#), etc.) se desarrollan y se publican como un todo. A esto es a lo que nos referimos como “sistema base”. Los programas instalados por el usuario, como por ejemplo Apache, X11, Mozilla Firefox, etc. no forman parte del sistema “base”. Estos programas normalmente se instalan desde la [Colección de Packages y Ports](#). Para separarlos del sistema “base” se instalan normalmente en `/usr/local/`. De ahí que los binarios de los programas instalados por los usuario se sitúen en `/usr/local/bin/`, los ficheros de configuración en `/usr/local/etc/`, etc.

Los servicios se arrancan mediante líneas `NombreDeServicio_enable="YES"` en `/etc/rc.conf` ([rc.conf\(5\)](#)). Eche un vistazo a `/etc/defaults/rc.conf`: La configuración por omisión del sistema puede sobrescribirse desde `/etc/rc.conf`. Además, al instalar programas debe asegurarse de leer la documentación para determinar cómo arrancar los servicios que puedan estar relacionados.

El siguiente fragmento de `/etc/rc.conf` arranca [sshd\(8\)](#) y Apache 2.2. También hace que Apache arranque con SSL.

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

Una vez que un servicio aparece en `/etc/rc.conf` el servicio puede arrancarse desde la línea de comandos (sin reiniciar el sistema):

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

Si un servicio no dispone de la correspondiente entrada en `/etc/rc.conf` se puede arrancar desde la línea de comandos usando la opción `forstart`:

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

5. Configuración de la red

5.1. Interfaces de red

Los nombres de interfaces de red representan en FreeBSD el nombre del controlador seguido por un número, en lugar del esquema que usa Linux®, el genérico. *ethX*. La salida de [ifconfig\(8\)](#) que se muestra a continuación presenta dos interfaces del tipo Intel® Pro 1000 (em0 y 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
```

Una dirección IP puede ser asignada a una interfaz usando [ifconfig\(8\)](#). Para que permanezca entre reinicios, la configuración IP debe ser incluida en `/etc/rc.conf`. Las siguientes entradas en `/etc/rc.conf` especifican el hostname, la dirección IP, y el gateway por defecto:

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

En su lugar, utilice las siguientes entradas para configurar una interfaz de red con DHCP:

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

6. Firewall

FreeBSD no usa las IPTABLES de Linux® para su firewall. En su lugar, FreeBSD ofrece tres firewalls a nivel del kernel:

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

PF está desarrollado por el proyecto OpenBSD y portado a FreeBSD. PF fue creado como un reemplazo para IPFILTER y su sintaxis es similar. PF se puede combinar con [altq\(4\)](#) para proporcionar QoS.

Este ejemplo de PF permite la entrada de tráfico SSH:

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

IPFILTER es el firewall desarrollado por Darren Reed. No es específico de FreeBSD y se ha portado a varios sistemas operativos, incluidos NetBSD, OpenBSD, SunOS, HP/UX y Solaris.

La sintaxis de IPFILTER para permitir la entrada de tráfico SSH es:

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

IPFW es el cortafuegos desarrollado y mantenido por FreeBSD. Se puede combinar con [dummynet\(4\)](#) para proporcionar traffic shaping y simular diferentes tipos de conexiones.

La sintaxis de IPFW para permitir la entrada de tráfico SSH sería:

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

7. Actualizando FreeBSD

Hay dos métodos para actualizar un sistema FreeBSD: desde el código fuente o desde la actualización de los binarios.

Actualizar desde código fuente es el método más complejo pero el que ofrece mayor flexibilidad. El proceso implica la sincronización de una copia local del código fuente de FreeBSD con los servidores Subversion de FreeBSD. Una vez actualizado el código fuente, puede compilar nuevas versiones del kernel y utilidades.

Las actualizaciones de los binarios son similares a usar `yum` o `apt-get` para actualizar un sistema Linux®. En FreeBSD, `freebsd-update(8)` puede usarse para obtener las nuevas actualizaciones de los binarios e instalarlas. Estas actualizaciones pueden ser programadas usando `cron(8)`.



Nota

Cuando use `cron(8)` para programar actualizaciones, use `freebsd-update cron` en `crontab(1)` para reducir la posibilidad de que una gran cantidad de máquinas se actualicen al mismo tiempo:

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

Para obtener más información de las actualizaciones de código y binarias, consulte el [capítulo sobre la actualización](#) en el Manual de FreeBSD.

8. procfs: Desaparecido pero no olvidado

En algunas distribuciones de Linux®, puede consultar `/proc/sys/net/ipv4/ip_forward` para determinar si IP forwarding está habilitado. En FreeBSD, `sysctl(8)` se usa para ver esta y otras configuraciones del sistema.

Por ejemplo, utilice el siguiente comando para comprobar si IP forwarding está habilitado en FreeBSD:

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

Use `-a` para listar todos los ajustes del sistema:

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

Si una aplicación necesita `procfs`, añada la siguiente línea a `/etc/fstab`:

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

Incluir `noauto` evitará que `/proc` se monte automáticamente en el arranque.

Para montar el sistema de archivos sin reiniciar:

```
# mount /proc
```

9. Comandos comunes

Algunos equivalentes de los comandos comunes son los siguientes:

Comando Linux® (Red Hat/Debian)	Equivalente en FreeBSD	Objetivo
<code>yum install package / apt-get install package</code>	<code>pkg install package</code>	Instalar el paquete desde el repositorio remoto
<code>rpm -ivh package / dpkg -i package</code>	<code>pkg add package</code>	Instalar un paquete local
<code>rpm -qa / dpkg -l</code>	<code>pkg info</code>	Listar los paquetes instalados
<code>lspci</code>	<code>pciconf</code>	Listar los dispositivos PCI
<code>lsmod</code>	<code>kldstat</code>	Listar los módulos cargados en el kernel

Comando Linux® (Red Hat/Debian)	Equivalente en FreeBSD	Objetivo
modprobe	kldload / kldunload	Cargar/Descargar módulos del kernel
strace	truss	Rastrear llamadas al sistema

10. Conclusión

Este documento ha proporcionado una visión general de FreeBSD. Consulte el [Manual de FreeBSD](#) para obtener una información más detallada de los temas tratados, así como otros muchos temas que no se han tratado en este documento.

