

Guide rapide pour dŽbuter avec FreeBSD ^ l'attention des utilisateurs de Linux

RŽsumŽ

Ce document a pour but de familiariser rapidement les utilisateurs de Linux de niveau intermŽdiaire ^ avancŽ avec les fondamentaux de FreeBSD.

Version fran•aise de Frederic Culot <culot@freebsd.org>.

Table des mati•res

1. Introduction	1
2. InterprŽteurs de commandes: Pas de Bash?	2
3. Paquetages et logiciels portŽs: ajouter des logiciels sous FreeBSD	2
4. DŽmarrage Syst•me: o• sont les niveaux d'exŽcution (run-levels)?	4
5. Configuration RŽseau	5
6. Pare-feu	6
7. Mettre ^ jour FreeBSD	7
8. procfs: disparu mais pas oubliŽ	8
9. Commandes Usuelles	9
10. Conclusion	9

1. Introduction

Ce document met en Žvidence les diffŽrences entre FreeBSD et Linux de telle sorte que les utilisateurs de Linux d'un niveau intermŽdiaire jusqu'• avancŽ puissent se familiariser rapidement avec les fondamentaux de FreeBSD. Il s'agit ici simplement d'une courte introduction technique qui n'a pas pour objectif d'expliquer les diffŽrences "philosophiques" entre les deux syst•mes d'exploitation.

Ce document part du principe que vous avez dŽj installŽ FreeBSD. Si vous n'avez pas installŽ FreeBSD ou que vous avez besoin d'aide pour mener ^ terme le processus d'installation, vous pouvez vous rŽfŽrer au chapitre [Installer FreeBSD](#) du Manuel de RŽfŽrence FreeBSD.

2. Interpréteurs de commandes: Pas de Bash?

Ceux qui ont l'habitude de Linux sont souvent surpris de voir que Bash n'est pas l'interpréteur de commandes par défaut de FreeBSD. En fait, Bash n'est même pas présent dans l'installation par défaut. À la place, FreeBSD utilise [tsh\(1\)](#) comme interpréteur par défaut. Cependant, Bash ainsi que vos autres interpréteurs de commandes favoris sont disponibles dans les [Paquetages et logiciels portés](#) de FreeBSD.

Si vous installez d'autres interpréteurs de commandes vous pouvez utiliser [chsh\(1\)](#) pour définir un interpréteur par défaut pour un utilisateur. Il est cependant recommandé de ne pas modifier l'interpréteur de commandes par défaut de `root`. La raison en est que les interpréteurs de commandes qui ne sont pas inclus dans la distribution de base sont normalement installés dans `/usr/local/bin` ou `/usr/bin`. Dans le cas d'un problème les systèmes de fichiers contenant `/usr/local/bin` et `/usr/bin` peuvent ne pas être montés. Dans ce cas `root` n'aurait pas accès à son interpréteur de commandes par défaut, ce qui empêcherait `root` de pouvoir se connecter. Pour cette raison un second compte `toor`, a été créé pour l'utilisation avec des interpréteurs de commandes qui ne sont pas ceux par défaut. Vous pouvez consulter les questions fréquemment posées sur la sécurité concernant le [compte toor](#) pour plus d'information.

3. Paquetages et logiciels portés: ajouter des logiciels sous FreeBSD

En plus de la traditionnelle méthode UNIX d'installation de logiciels (télécharger les sources, extraire, éditer le code source, et compiler), FreeBSD offre deux autres méthodes pour installer des applications: les paquetages et les logiciels portés. Une liste complète de tous les logiciels portés et paquetages disponibles se trouve [ici](#).

3.1. Paquetages

Les paquetages sont des applications pré-compilées, les équivalents FreeBSD des fichiers `.deb` pour les systèmes basés sur Debian/Ubuntu et des fichiers `.rpm` pour les systèmes basés sur Red Hat/Fedora. Par exemple, la commande suivante installe Apache 2.2:

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

Utiliser l'option `-r` indique à [pkg\(add\)](#) de télécharger automatiquement le paquetage et de l'installer, ainsi que toutes ses dépendances:

```
# 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.



Si vous utilisez une version RELEASE de FreeBSD (6.2, 6.3, 7.0, etc., g n ralement install e depuis un CD-ROM) `pkg_add -r` t l chagera les paquetages compil s sp cifiquement pour cette version. Ces paquetages peuvent *ne pas*  tre la version la plus r cente de l'application. Vous pouvez utiliser la variable `PACKAGESITE` pour surcharger ce comportement par d faut. Par exemple, assignez `ftp://ftp.freebsd.org/pub/FreeBSD/ports/i386/packages-6-stable/Latest/ ^ PACKAGESITE` pour t l chager les paquetages les plus r cents construits pour les versions 6.X.

Pour plus d'information concernant les paquetages vous pouvez vous r f rer   la section 4.4 du Manuel FreeBSD: [Utiliser le syst me des logiciels pr -compil s](#).

3.2. Les logiciels port s

Le catalogue des logiciels port s est la seconde m thode propos e par FreeBSD pour installer des applications. Le catalogue des logiciels port s a pour architecture un ensemble de Makefiles et de fichiers correctifs sp cifiquement adapt s pour pouvoir installer depuis les sources des applications diverses sur FreeBSD. Lors de l'installation d'un logiciel port  le syst me t l chagera le code source, appliquera tous les correctifs n cessaires, compilera le code, et installera l'application (et fera de m me pour toutes ses d pendances).

Le catalogue des logiciels port s, parfois appel e l'arbre des ports (ports tree en anglais), peut  tre trouv  dans `/usr/ports`. Nous partons ici du principe que le catalogue des logiciels port s a  t  install  pendant le processus d'installation de FreeBSD. Si le catalogue des logiciels port s n'  pas  t  install , il peut  tre ajout    partir des disques d'installation en utilisant `sysinstall(8)`, ou bien r cup r  depuis les serveurs FreeBSD en utilisant `csup(1)` ou `portsnap(8)`. Des instructions d taill es concernant l'installation du catalogue des logiciels port s peuvent  tre consult es dans la [section 4.5.1](#) du Manuel.

Installer un logiciel port  est aussi simple (en g n ral) que de se d placer dans le r pertoire du logiciel port  et de lancer le processus de compilation. L'exemple suivant installe Apache 2.2 depuis le catalogue des logiciels port s:

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

Un des avantages majeurs d'utiliser les logiciels port s pour installer des logiciels est de pouvoir adapter les options d'installation. Par exemple, lors de l'installation d'Apache 2.2 depuis une version port e, il vous est possible d'activer `mod_ldap` en fixant la variable `WITH_LDAP` de `make(1)`:

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

Vous pouvez vous référer à la section 4.5 du Manuel FreeBSD, [Utiliser le catalogue des logiciels portés](#), pour obtenir plus d'information concernant le catalogue des logiciels portés.

3.3. Logiciel porté ou paquetage, lequel devrais-je utiliser?

Les paquetages sont simplement des logiciels portés pré-compilés, donc il s'agit vraiment de choisir entre une installation depuis les sources (logiciels portés) ou une installation utilisant des binaires pré-compilés. Chaque méthode présente ses avantages:

Les paquetages (binaires)

- ¥ Installation plus rapide (compiler de grosses applications peut prendre du temps).
- ¥ Inutile de comprendre comment compiler un logiciel.
- ¥ Inutile d'installer des compilateurs sur votre système.

Les logiciels portés (sources)

- ¥ Possibilité d'adapter les options d'installation (les paquetages sont normalement compilés avec les options standards alors qu'avec les logiciels portés vous pouvez adapter diverses options comme la compilation de modules additionnels ou le changement de l'emplacement par défaut).
- ¥ Vous pouvez appliquer vos propres fichiers correctifs si vous le souhaitez.

Si vous n'avez pas de besoins particuliers, les paquetages seront probablement tout à fait adaptés à votre situation. S'il vous arrivait d'avoir des besoins spécifiques, les logiciels portés seraient plus appropriés (et n'oubliez pas que si vous devez effectuer des adaptations mais que vous préférez les paquetages, vous pouvez toujours compiler un paquetage personnalisé en utilisant `make`package`` et en copiant le paquetage sur d'autres serveurs).

4. Démarrage Système: où sont les niveaux d'exécution (run-levels)?

Linux utilise le système d'initialisation SysV alors que FreeBSD utilise le style traditionnel BSD avec `init(8)`. Avec `init(8)` il n'existe pas de niveaux d'exécution et pas de `/etc/inittab`, mais à la place le démarrage est contrôlé par l'utilitaire `rc(8)`. Le script `/etc/rc` lit `/etc/defaults/rc.conf` et `/etc/rc.conf` pour déterminer quels services doivent être démarrés. Les services déclarés sont alors démarrés en lançant les scripts d'initialisation du service considérés que l'on trouve dans `/etc/rc.d/` et `/usr/local/etc/rc.d/`. Ces scripts sont similaires aux scripts que l'on trouve dans `/etc/init.d/` sur les systèmes Linux.

*Pourquoi existe-t-il deux emplacements distincts pour les scripts d'initialisation de services ?
Les scripts que l'on trouve dans `/etc/rc.d/` sont pour les applications qui font partie du système*

de base ([cron\(8\)](#), [sshd\(8\)](#), [syslog\(3\)](#), et d'autres). Les scripts dans `/usr/local/etc/rc.d/` sont pour les applications installées par les utilisateurs telles que Apache, Squid, etc.

Quelle est la différence entre le système de "base" et les applications installées par l'utilisateur? FreeBSD est développé comme un système d'exploitation complet. En d'autres termes, le noyau, les bibliothèques système, et les utilitaires présents dans l'espace utilisateur (tels que [ls\(1\)](#), [cat\(1\)](#), [cp\(1\)](#), etc.) sont développés et distribués conjointement. C'est ce qui fait référence en parlant de système de "base". Les applications installées par l'utilisateur sont des applications qui ne font pas partie du système de "base", telles que Apache, X11, Mozilla Firefox, etc. Ces applications sont généralement installées en utilisant le [Catalogue des logiciels ports et les paquetages](#) de FreeBSD. Afin de les conserver à l'extérieur du système de "base", les applications installées par l'utilisateur sont normalement placées dans `/usr/local/`. Ainsi les binaires installés par l'utilisateur se trouvent dans `/usr/local/bin/`, les fichiers de configuration dans `/usr/local/etc/`, et ainsi de suite.

Les services sont activés en spécifiant `NomDuService_enable="YES"` dans `/etc/rc.conf` ([rc.conf\(5\)](#)). Pour vous faire une idée, vous pouvez consulter les valeurs par défaut du système dans `/etc/defaults/rc.conf`, ces valeurs par défaut sont surchargées par celles trouvées dans `/etc/rc.conf`. De plus, lors de l'installation de logiciels additionnels, veuillez consulter leur documentation pour déterminer de quelle manière sont activés les services associés, le cas échéant.

Le bout de code suivant extrait de `/etc/rc.conf` active [sshd\(8\)](#) et Apache 2.2. Il spécifie également que Apache devrait être lancé avec SSL.

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

Dès lors qu'un service a été activé dans `/etc/rc.conf`, ce service peut être démarré depuis la ligne de commande (sans avoir à redémarrer le système):

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

Si un service n'a pas été activé il peut être démarré depuis la ligne de commande en utilisant `forstart`:

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

5. Configuration Réseau

5.1. Interfaces Réseau

À la place d'un identifiant graphique *ethX* que Linux utilise pour identifier une interface réseau, FreeBSD utilise le nom du pilote suivi d'un nombre en tant qu'identifiant. La sortie suivante de `ifconfig(8)` montre deux interfaces réseau Intel Pro 1000 (em0 et em1):

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

5.2. Configuration IP

Une adresse IP peut être assignée à une interface en utilisant `ifconfig(8)`. Cependant, pour assurer la persistance même après un redémarrage, la configuration IP doit être enregistrée dans `/etc/rc.conf`. Dans l'exemple suivant sont spécifiés le nom de la machine, l'adresse IP, et la passerelle par défaut:

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

Utilisez ceci pour configurer une interface pour DHCP:

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

6. Pare-feu

Tout comme IPTABLES pour Linux, FreeBSD offre également un pare-feu au niveau du noyau; en fait FreeBSD offre trois pare-feux distincts:

¥ [IPFIREWALL](#)

¥ [IPFILTER](#)

¥ [PF](#)

IPFIREWALL ou IPFW (la commande pour g rer un jeu de r gles IPFW est [ipfw\(8\)](#)) est le pare-feu d velopp  et maintenu par les d veloppeurs FreeBSD. IPFW peut  tre coupl  avec [dummynet\(4\)](#) pour fournir des possibilit s de r gulation du trafic et simuler diff rents types de connexions r seau.

Voici un exemple de r gle IPFW pour autoriser SSH en entr e:

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

IPFILTER est le pare-feu applicatif d velopp  par Darren Reed. Celui-ci n est pas sp cifique   FreeBSD et a  t  port  sur de nombreux syst mes d exploitation tels que NetBSD, OpenBSD, SunOS, HP/UX, et Solaris.

Voici un exemple de r gle IPFILTER pour autoriser SSH en entr e:

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

Le dernier pare-feu, PF, est d velopp  par le projet OpenBSD. PF a  t  cr   pour remplacer IPFILTER, ce qui fait que la syntaxe de PF est tr s similaire   celle de IPFILTER. PF peut  tre coupl  avec [altq\(4\)](#) pour fournir des possibilit s de QoS.

Voici un exemple de r gle PF pour autoriser SSH en entr e:

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

7. Mettre   jour FreeBSD

Il existe trois m thodes diff rentes pour mettre   jour un syst me FreeBSD:   partir des sources, les mises   jour binaires, et les disques d installation.

Mettre   jour depuis les sources est la m thode la plus compliqu e mais elle offre la plus grande flexibilit . Le processus implique de synchroniser une copie locale du code source de FreeBSD avec les serveurs CVS (Concurrent Versioning System - Syst me de gestion de Versions Concurrentes) de FreeBSD. Une fois que le code source local est   jour vous pouvez compiler les nouvelles versions du noyau et du syst me de base. Pour plus d informations concernant les mises   jour depuis les sources vous pouvez consulter [le chapitre sur la mise   jour](#) du manuel FreeBSD.

Les mises   jour binaires ressemblent aux commandes [yum](#) ou [apt-get](#) utilis es pour mettre   jour un syst me Linux. La commande [freebsd-update\(8\)](#) t l chargera les nouvelles mises   jour et les installera. Les mises   jour peuvent  tre programm es par l'interm diaire de [cron\(8\)](#).

!

Si vous utilisez [cron\(8\)](#) pour programmer vos mises   jour, assurez-vous d' utiliser la commande [freebsd-update cron](#) dans votre [crontab\(1\)](#) pour r duire le nombre de machines r cup rant les mises   jour au m me moment.

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

La dernière méthode, qui permet de mettre à jour en utilisant les disques d'installation, est simple: démarrez depuis les disques d'installation et sélectionnez l'option de mise à jour.

8. procfs: disparu mais pas oublié

Avec Linux, il vous est peut-être arrivé de consulter `/proc/sys/net/ipv4/ip_forward` pour déterminer si le routage IP était activé. Avec FreeBSD vous devriez utiliser `sysctl(8)` pour voir ce réglage ainsi que d'autres réglages système parce que `procfs(5)` a été abandonné dans les versions actuelles de FreeBSD (notez que `sysctl` est disponible aussi sous Linux).

Dans l'exemple du routage IP voici ce que vous utiliseriez pour déterminer si le routage IP est activé sur votre système FreeBSD:

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

L'option `-a` est utilisée pour lister tous les réglages du système:

```
% 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
É 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
...
```



Certaines de ces valeurs `sysctl` sont uniquement accessibles en lecture.

`procfs` est parfois nécessaire comme pour faire fonctionner de vieux logiciels, pour examiner des appels système en utilisant `truss(1)`, et pour la [Compatibilité Binaire avec Linux](#) (celle-ci utilise cependant son propre `procfs`, `linprocfs(5)`). Si vous avez besoin de monter `procfs` vous pouvez ajouter la ligne suivante dans `/etc/fstab`:

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



`noauto` fera en sorte que `/proc` ne soit pas monté automatiquement lors du démarrage.

Et ensuite montez `procfs` avec:

```
# mount /proc
```

9. Commandes Usuelles

9.1. Gestion des Paquetages

Commande Linux [®] (Red Hat/Debian)	Equivalent FreeBSD	R [™] le
<code>yum install paquetage / apt-get install paquetage</code>	<code>pkg_add -r paquetage</code>	Installation de <i>paquetage</i> depuis un dépôt distant
<code>rpm -ivh paquetage / dpkg -i paquetage</code>	<code>pkg_add -v paquetage</code>	Installation de <i>paquetage</i>
<code>rpm -qa / dpkg -l</code>	<code>pkg_info</code>	Lister les paquetages installés

9.2. Gestion System

Commande Linux [®]	Equivalent FreeBSD	R [™] le
<code>lspci</code>	<code>pci conf</code>	Lister les périphériques PCI
<code>lsmod</code>	<code>kl dstat</code>	Lister les modules noyau chargés
<code>modprobe</code>	<code>kl load / kl unload</code>	Charger/Décharger les modules noyau
<code>strace</code>	<code>truss</code>	Examiner les appels system

10. Conclusion

Nous espérons que ce document vous a fourni suffisamment d'informations pour que vous puissiez faire vos premiers pas avec FreeBSD. Vous pouvez consulter [le Manuel FreeBSD](#) pour une couverture plus complète des sujets abordés ici ainsi que de tous les autres sujets non abordés dans ce document.