

FreeBSD

1.Introduction:

FreeBSD est un système d'exploitation UNIX libre. Le nom vient de l'association d'une part de *free* qui signifie à la fois libre (liberté) et gratuit dans l'anglais courant, et d'autre part de *BSD* (Berkeley Software Distribution), l'UNIX® développé à l'Université Berkeley de Californie. Free prend un sens plus connoté dans ce nom : il signifie que le logiciel peut être utilisé gratuitement même pour un usage commercial, que les sources complètes sont disponibles et utilisables avec un minimum de restrictions quant à leur usage, leur distribution et leur incorporation dans un autre projet (commercial ou non), et enfin que n'importe qui est libre de soumettre son code source pour enlever un bogue (bug) ou améliorer le logiciel, ce code étant incorporé aux sources après accord.

L'objectif du projet FreeBSD est de fournir un logiciel pour n'importe quelle utilisation, ce avec le moins de restrictions possibles.

2.Configuration principale

L'emplacement principal pour les données de configuration du système est le fichier `/etc/rc.conf`. Ce fichier contient une large gamme d'informations de configuration, principalement utilisées au démarrage du système pour configurer ce dernier. Son nom le sous-entend; c'est l'information de configuration pour les fichiers `rc*`.

Un administrateur devrait ajouter des entrées dans le fichier `rc.conf` pour remplacer les valeurs par défaut du fichier `/etc/defaults/rc.conf`. Les fichiers de valeurs par défaut ne devraient pas être copiés directement tels quels dans `/etc` - ils contiennent des valeurs par défaut, et non pas des exemples. Tout changement spécifique au système devrait être fait dans le fichier `rc.conf`.

Un certain nombre de stratégies peuvent être appliquées dans le cas d'applications en grappe pour séparer la configuration d'un site de celle d'un système afin de réduire le travail d'administration. L'approche recommandée est de placer la configuration propre au site dans un autre fichier comme `/etc/rc.conf.site`, puis ensuite inclure ce fichier dans `/etc/rc.conf`, qui ne contiendra seulement que les informations spécifiques au système.

Comme `rc.conf` est lu par `sh` il est assez trivial d'effectuer cela. Par exemple:

`rc.conf:`

```
. /etc/rc.conf.site
hostname="node15.example.com"
network_interfaces="fxp0 lo0"
ifconfig_fxp0="inet 10.1.1.1"
```

- rc.conf.site:

```
defaultrouter="10.1.1.254"  
saver="daemon"  
blanktime="100"
```

Le fichier rc.conf.site peut être distribué à l'ensemble des systèmes en utilisant rsync ou un programme semblable, tandis que le fichier rc.conf reste unique.

Mettre à jour le système en employant `sysinstall` ou `make world` n'écrasera pas le fichier rc.conf, les informations de configuration du système ne seront donc pas perdues.

3. Installation des logiciels sous FreeBSD:

Une petite introduction à comment on peut installer des logiciels sous FreeBSD.

Il existe deux méthodes différentes pour installer des logiciels sous FreeBSD :

- les paquets,
- les ports.

Dans le premier cas il s'agit d'archives contenant des binaires qui seront directement mis en place sur le système. Dans le second, les sources des logiciels voulus seront téléchargées, compilées, en prenant en compte les options voulus, puis installées.

Ensuite du point de vue du système il n'y a aucune différence entre les deux méthodes d'installation, les fichiers sont gérés tout pareil dans les deux cas.

Les paquets

Cette première méthode offre une grande rapidité, mais peu de souplesse : on est obligé d'accepter les binaires et on ne peut choisir des options de compilation.

On procède de la sorte pour installer :

```
pkg_add -r nomdupaquet
```

Et le paquet est directement téléchargé et installé, avec ses dépendances. Il y a une notion importante, qui est que l'URL de recherche est constituée en partie de la version du système, et que l'on ne peut la changer. Donc si vous avez installé une version PRERELEASE, dans quelques mois, cette commande ne marchera plus sauf si vous passez à une version STABLE ou CURRENT.

Les ports

On peut les installer avec `sysinstall`. Il s'agit d'une collection de répertoires contenant des Makefiles, permettant de télécharger, compiler et installer des logiciels. Tout cela se trouve dans `/usr/ports`.

On localise d'abord le logiciel. Disons que pour cela on a :

```
make search key="monsoft"
```

Mais que cela est souvent trop verbeux et que l'on a pas ce que l'on veut. Sinon on peut utiliser la complétion du shell qui un moyen plus efficace :

```
ls -d */*monsoft*
```

Après on va dans ce répertoire. On peut le nettoyer d'une éventuelle précédente installation :

```
make clean
```

Puis compiler le logiciel :

```
make
```

Là, on peut avoir des options dans un menu en curses. Si ultérieurement on veut consulter de nouveau ces options, on peut faire `make config`. Pour certains logiciels, les options sont des variables du Makefile, par exemple :

```
make PYTHON_VERSION=python2.3
```

Si on passe une variable au Makefile, il faut toujours le faire par la suite. Si on est sûr de le vouloir tout le temps, on peut la mettre dans le fichier `/etc/make.conf`.

Pour installer :

```
make install
```

Si on veut désinstaller un logiciel :

```
make deinstall
```

Le réinstaller :

```
make reinstall
```

Normalement pour les ports bien fait, les fichiers de configurations ne sont pas enlevés quand on désinstalle, et ne sont pas écrasés lorsque l'on réinstalle. C'est bien pour les mises à jour.

On peut très bien partager ce répertoire entre plusieurs machines via NFS. Si certaines d'entre elles n'ont pas internet, on peut télécharger les sources et les dépendances d'un port avec respectivement :

```
make fetch
```

Et :

```
make fetch-recursive
```

Si l'on est derrière un proxy, les variables d'environnement `http_proxy` et `ftp_proxy` peuvent aider. Si on veut les garder d'une manière plus définitive on peut ajouter dans le fichier `/etc/make.conf` :

```
FETCH_ENV= HTTP_PROXY=http://proxy:3128 FTP_PROXY=http://proxy:3128
```

Les ports utiles

- `cvsup` pour mettre à jour l'arborescence contenant les ports,
- `portupgrade` permettant de mettre à jour tout les ports contenu sur le système. Je déconseille vivement le `portupgrade -a`, qui a de forte chance

de ne pas marcher. Il faut mettre à jour le système petit morceau par petit morceau suivant les besoins.

Les autres commandes utiles

Si on veut avoir des nouvelles d'un logiciel installé :

```
pkg_info monsoft\*
```

Plus brièvement :

- `port_info` appelé tout seul va lister tout les ports installés avec une petite description,
- `port_version` va lister les ports installés avec leur version et indiquer s'ils ont besoin d'une mise à jour.