

Installation et Configuration de FreeBSD comme poste de travail

Table of Contents

- Installation et Configuration de FreeBSD comme poste de travail
 - Environnement
 - Installation
 - Le Shell
 - Boot
 - Logo Orb du bootloader EFI ne s'affiche pas en bitmap
 - Permettre la mise à l'heure automatique
 - Lecteur de carte SD
 - Audio
 - Bluetooth
 - AESNI (accélération matériel du chiffage)
 - SMBus
 - Configuration avancée du réseau
 - Wifi
 - Activation du Privacy Extension d'IPv6
 - DHCP en background pendant le démarrage
 - Failover entre carte réseau filaire et wifi
 - USB Tethering
 - Firewall
 - Localisation
 - Configuration du clavier Bépo
 - Permettre à l'utilisateur d'utiliser les périphériques USB, graveur CD/DVD
 - Optimisations
 - Configurer le scheduler pour un usage «desktop»
 - Réparation fsck automatique et background fsck (sauf si installation ZFS)
 - Désactivation de l'access time sur la partition /
 - Pourcentage réservé du disque
 - Éviter la création de fichier .core
 - Installation des Mises à jour de FreeBSD
 - Utiliser DMA à la place de sendmail
 - Installation des logiciels externes (packages)
 - Remplacement du dépôt quarterly par latest
 - Utilisation de pkg
 - Installer les root CA (FreeBSD 12 ou plus vieux)
 - Mise à jour des microcodes CPU
 - Remplacer powerd par powerd++
 - Montage automatique des disques amovibles

- Environnement graphique X11
 - xorg
 - Drivers vidéo X11 pour Intel (les anciennes)
 - Drivers vidéo X11 pour nvidia
 - dbus
 - Polkit (Chapitre désormais inutile?)
 - Pavé Synaptics, Trackpoint (Lenovo) ou Elantec
 - XFCE4
 - SLIM
 - Installation
 - Thème FreeBSD
 - Disposition du clavier sous slim
 - XFCE
 - Police de caractères
 - Les applications
 - Webcam
 - Firefox
 - Chromium
 - mpv
 - LibreOffice
 - CUPS
 - Serveur TFTP
 - Émulateur de terminal série (équivalent de l'hyperterminal ou teraterm)
- Personnalisation avancée
 - Collecte d'information
 - Inventaire matériel
 - Charge système par des graphiques: systat
 - Charge système en mode compact: vmstat
 - Charge des disques: gstat
 - Consommation électrique
 - Usage de ntfs-3g dans le fstab
 - Compilation des sources
 - FreeBSD avec la pile TCP BBR
 - Compilation batch des ports

Ce document inclut l'ensemble des étapes réalisées pour obtenir FreeBSD 13 installé pour une utilisation optimisée «poste de travail» (Xfce, LibreOffice, multimédia, etc..) sur un PC portable.

Il est beaucoup plus simple d'utiliser une version déjà prête pour une utilisation «poste de travail» en utilisant **GhostBSD**, **helloSystem**, **nomadBSD**, ou **MidnightBSD**, mais je préfère avoir un «vrai» FreeBSD, entièrement recompilé et installé à la main.

Environnement

L'environnement issu de ce tutoriel, comprendra:

1. un support matériel complet: Video, LAN, Wifi, Bluetooth et une gestion de l'énergie minimale.
2. un desktop avec les logiciels suivants:
 1. **KeepassXC** pour une solution multi-plateforme stockage des mots de passe et rôle d'agent de clé SSH;
 2. **bhyve**: hyperviseur;
 3. **LibreOffice**;
 4. **Veracrypt**: une solution multiplateforme de chiffrement;
 5. **Wireshark**;
 6. **Chromium** et/ou **Firefox**;
 7. **cu** ou **tip**: émulateur de terminal VT100 sur port série.

Installation

Récupération du fichier ISO/IMG

Sur le serveur FTP officiel:

```
ftp://ftp.freebsd.org/pub/FreeBSD/releases/ISO-IMAGES/
```

Concernant l'image disque USB (memstick.img): La transférer à coup de "dd bs=64k", ou pour les Windowsiens d'Image writer.

FreeBSD Installer

L'installateur est en mode texte, mais très simple: Il suffit de sélectionner «Install» sur le premier écran, puis de suivre les conseils:

- Sélectionner le type de clavier et le hostname
- Concernant les distributions à installer: Laisser les valeurs par défaut (kernel-debug et lib32 qui sont obligatoire pour certaine applications comme virtualbox)
- Partitionnement: Si on dispose de 4Go de RAM ou moins, choisir le mode "Auto (UFS)", sinon "Auto (ZFS)" pour se simplifier la vie, mais on peut aussi manuellement:
 - Créer une première partition swap de la taille de votre RAM (histoire d'avoir assez de place pour récupérer un éventuel crash dump). Labellisée "swap" par exemple.
 - Puis créer une seconde partition avec tout le reste pour la partition root /. Labellisé "rootfs" par exemple.
 - Si la machine supporte l'assistance matérielle pour le chiffrement (AESNI), l'activation du chiffrement est une bonne idée (à noter que lors du boot le clavier sera en QWERTY)
 - Puis sélectionner le disque et «Entire Disk»:
 - Partition Editor: Si on est en mode «Guided» il suffit de faire «Exit» et «Save»
 - Attendre environ 2 minutes pour l'installation
- Puis viennent quelques questions triviales: mot de passe de l'utilisateur root, configuration du réseau (incluant le Wifi si on sélectionne cette interface) et la timezone
- Concernant les services à démarrer automatiquement: Désactiver sshd (rien à faire sur un desktop) et activer ntpd et powerd
- Créer l'utilisateur:
 - À la question «**Invite USER into other groups?**», répondre, avec un simple espace

entre les noms: wheel operator dialer video

- «**wheel**», permet de passer root par la commande su
- «**operator**», obligatoire pour utiliser la commande su et éteindre le PC
- «**dialer**», permet d'utiliser les ports séries (par «cu» et «tip»)
- «**video**», permet d'accéder à la carte graphique
- **Login class**, laisser «default» (on la changera plus tard)
- **Shell**: tcsh
- **Home directory**: Par défaut
- Puis choisir un mot de passe et confirmer la création
- De retour sur le «Final Configuration», sélectionner «Exit», puis «Reboot»



Attention au vocabulaire utilisé par FreeBSD:

* Un «slice» est une partition MBR, elles portent comme nom: Le nom du disque suivi des lettres s1, s2, s3, etc...

* Une «partition» est une sous-partition MBR, elles portent comme nom: Le nom du disque, éventuellement le nom du slice, puis une lettre (a pour l'ensemble, b pour la swap, etc...)

Le Shell

Le shell bash n'est pas inclus par défaut dans FreeBSD: C'est tcsh que l'on va utiliser.

Il est déconseillé d'utiliser un shell installé depuis les ports (bash par exemple) pour l'utilisateur root, car il risque de ne plus fonctionner en cas de problème (bibliothèque manquante par exemple ou pas d'accès à /usr/local).

Si on veut utiliser un autre shell pour le compte root: modifier plutôt le shell du compte toor (ce compte administrateur sert à ça).

Par défaut le tcsh ne propose pas de couleur et fait biper le PC il faut donc le configurer en éditant le fichier /etc/csh.cshrc communs à tous les utilisateurs:

```
cat >> /etc/csh.cshrc <<EOF
# Add color to CLI
setenv CLICOLOR true
setenv COLORTERM true
# Disable system beep
set nobeep
EOF
```

Désactiver le buzzer:

```
cat >> /etc/sysctl.conf <<EOF
# Désactive le buzzer en mode vt(4)
kern.vt.enable_bell=0
```

E0F

Boot

Diminuer le timeout du menu du boot loader

Le menu du boot loader possède un timeout de 10 secondes, ce qui est un peu long.

Voici comment le réduire à 2 secondes:

```
sysrc -f /boot/loader.conf autoboot_delay=2
```

Ou pour complètement le désactiver:

```
sysrc -f /boot/loader.conf autoboot_delay=-1
```

Logo Orb du bootloader EFI ne s'affiche pas en bitmap

Fonctionnalité la plus attendue de FreeBSD 13, absolument obligatoire: L'affichage du nom FreeBSD et du logo en bitmap.

Sauf que parfois le mot FreeBSD est bien en bitmap mais le logo Orb s'affiche uniquement en ASCII.

Pour corriger le problème, forcer la résolution de l'écran EFI à la résolution native de la machine, par exemple 1366x768, donc:

```
echo 'efi_max_resolution="1366x768"' >> /boot/loader.conf
```

Permettre la mise à l'heure automatique

Si on ne l'av pas sélectionné pendant l'installation, 2 méthodes:

- **NTP**, pour les serveurs qui ne redémarrent pas souvent (daemon en tâche de fond)

```
service ntpd enable  
sysrc ntpd_sync_on_start=yes
```

- **NTPdate**, pour les postes de travail/laptop, synchronise l'heure au démarrage uniquement

```
service ntpddate enable  
sysrc ntpdate_enable=yes
```

Les Drivers (uniquement pour FreeBSD 11 ou plus vieux)

Depuis FreeBSD 12.0 qui utilise devmatch(8) pour charger le maximum de drivers par défaut, les versions plus ancienne de FreeBSD ne le font pas: On doit lui indiquer de charger les drivers spécifiques à votre matériel.

Pour obtenir la liste des drivers non chargés, on peut par exemple utiliser la commande suivante:

```
[root@laptop]~#dmesg | grep "no driver attached"
pci12: <network> at device 0.0 (no driver attached)
pci0: <multimedia, HDA> at device 27.0 (no driver attached)
pci0: <serial bus, SMBus> at device 31.3 (no driver attached)
pci9: <base peripheral, SD host controller> at device 0.1 (no driver attached)
pci9: <mass storage> at device 0.2 (no driver attached)
```

Ici on note une puce audio (multimedia, HDA), une carte réseau, le wifi dans mon cas (network), un chipset SMBus (serial bus, SMBus) et un lecteur de carte SD.

Lecteur de carte SD

Commencer par charger manuellement les drivers pour tester qu'ils correspondent bien:

```
kldload mmc
kldload mmcsd
kldload sdhci
```

Puis vérifier:

```
[root@laptop]~#dmesg | grep sdhci
sdhci0: <Generic SD HCI> mem 0xe5220000-0xe52201ff irq 18 at device 0.1 on pci9
sdhci0: 1 slot(s) allocated
```

On ajoute le nom de ces modules à la liste des modules à charger automatiquement (variable `kld_list`) du fichier `/etc/rc.conf`.

Si on n'a pas encore déclaré cette variable dans votre `/etc/rc.conf`, il est possible d'utiliser directement cette ligne:

```
sysrc kld_list+="mmc mmcsd sdhci"
```

Audio

Les drivers audio les plus courants sont activés par défaut depuis la 9.0, mais si ce n'est pas le cas,

charger un «méta» drivers qui va charger tous les drivers audio, puis regarder celui qui est accepté:

```
[root@laptop]~#kldload snd_driver
[root@laptop]~#cat /dev/sndstat
FreeBSD Audio Driver (newpcm: 64bit 2007061600/amd64)
Installed devices:
pcm0: <Intel 82801H High Definition Audio Controller> at memory 0xf6dfc000
irq 21 kld snd_hda [20080420_0052] [MPSAFE] (1p:1v/1r:1v channels duplex
default)
```

Dans ce cas présent, c'est le drivers `snd_hda` qui correspond. Ce driver est déjà inclus par défaut dans le noyau générique depuis la 9.0, mais pour une version plus ancienne on devra déclarer son chargement:

```
[root@laptop]~#sysrc kld_list+=snd_hda
```

Bluetooth

Pour utiliser les enceintes/casques bluetooth, c'est par le package **virtual_oss** (`pkg install virtual_oss`) qu'il faut passer.

AESNI (accélération matériel du chiffage)

Ce module est inclus dans le noyau de FreeBSD 13, mais pour les version plus ancienne il faut le charger manuellement.

Le module AESNI permet d'utiliser les capacité d'accélération de chiffrement des processeurs modernes mais il n'est pas chargé automatiquement (sauf si vous avez choisi de chiffrer votre disque à l'installation).

Il est possible de charger ce drivers tardivement (en l'ajoutant dans le `kld_list` du `/etc/rc.conf`) mais il est préférable de le charger le plus vite possible (dans le `/boot/loader.conf`) pour s'assurer le module crypto soit initialisé avec cette fonctionnalité.

```
[root@laptop]~#echo 'aesni_load="YES"' >> /boot/loader.conf
```

SMBus

(ce truc-là ne sert pas à grand-chose)

Charger et vérifier la bonne prise en compte du driver:

```
[root@laptop]~#kldload ichsmb
[root@laptop]~#dmesg | grep smbus
```

```
smbus0: <System Management Bus> on ichsmb0
```

Puis ajouter ce module à la liste de module à charger automatiquement:

```
sysrc kld_list+=smbus
```

Configuration avancée du réseau

Wifi

Normalement déjà réalisé pendant la phase d'installation, mais si ce n'est pas le cas la méthode la plus facile pour la configuration du wifi est d'utiliser la fonction netconfig de bsdinstall:

```
bsdinstall netconfig
```

Activation du Privacy Extension d'IPv6

Par défaut le mode Privacy Extension d'IPv6 n'est pas activé:

```
sysrc ipv6_privacy=YES
```

DHCP en background pendant le démarrage

Par défaut le client DHCP met en pause le démarrage jusqu'à recevoir (ou déclencher un timeout) une adresse. Pour éviter on le passe en background:

```
sysrc background_dhclient=YES
```

Failover entre carte réseau filaire et wifi

Cette configuration permet de basculer ultra rapidement entre le réseau filaire (prioritaire) et wifi.

Tester en premier la configuration WPA avant de mettre en place cet exemple (qui utilise une carte wifi Intel iwn0 et une carte Ethernet Intel em0).

On commence par noter l'adresse MAC de la carte Wifi et on configure la carte Ethernet avec cette même adresse MAC (ce qui permet de présenter la même adresse quelque soit l'interface utilisée):

```
setenv MACWIFI `ifconfig wlan0 | awk '/ether/ { print $2 }`  
sysrc wlans_iwn0=wlan0
```



```
sysrc ifconfig_em0="ether $MACWIFI up "  
sysrc ifconfig_wlan0="WPA powersave"  
sysrc create_args_wlan0="country FR regdomain ETSI"  
sysrc cloned_interfaces=lagg0  
sysrc ifconfig_lagg0="up laggproto failover laggport em0 laggport wlan0  
DHCP"  
sysrc ifconfig_lagg0_ipv6="inet6 accept_rtadv"
```

Un simple «service netif restart» n'est peut-être pas suffisant si votre interface wlan0 était déjà configurée: un redémarrage risque d'être nécessaire.

USB Tethering

Rien de plus simple car il n'y a pratiquement rien à faire (avec un téléphone Android) à part connecter le smartphone, activer le partage de connexion Internet par USB et demander une IP par DHCP.

Une interface réseau urndis(4) USB Remote NDIS Ethernet device nommée ue0 va être utilisée pour cet accès.

```
ugen0.2: <Xiaomi Mi-4c> at usb0  
urndis0 on uhub2  
urndis0: <RNDIS Communications Control> on usb0  
ue0: <USB Ethernet> on urndis0  
ue0: Ethernet address: 3e:a6:64:12:9c:ef  
root@idepad:~ # dhclient ue0  
DHCPDISCOVER on ue0 to 255.255.255.255 port 67 interval 4  
DHCPOFFER from 192.168.42.129  
DHCPREQUEST on ue0 to 255.255.255.255 port 67  
DHCPACK from 192.168.42.129  
bound to 192.168.42.21 -- renewal in 1800 seconds.
```

Firewall

La configuration du firewall en mode “workstation” est assez simple:

```
service firewall enable  
sysrc firewall_type=workstation  
service ipfw start
```

Dans le cas où on souhaite héberger quelques services réseau sur la machine, il est possible de les déclarer simplement:

```
sysrc firewall_myservices="22/tcp 80/tcp 443/tcp"  
sysrc firewall_allowservices=any
```

Pour éviter d'afficher la liste des règles chargées:

```
sysrc firewall_quiet=yes
```

Et pour logger les paquets rejetés dans /var/log/security:

```
sysrc firewall_logdeny=yes
```

Et pour activer le tout:

```
service ipfw start
```

Les différentes règles et leur stats s'affichent par un "ipfw show".

Localisation

Configuration du système en français

En root, rajouter dans /etc/login.conf :

```
cat >>/etc/login.conf <<EOF
french|French Users Accounts:\
:charset=UTF-8:\
:lang=fr_FR.UTF-8:\
:tc=default:
EOF
```

Et exécuter la commande suivante :

```
# cap_mkdb /etc/login.conf
```

Puis éditer le fichier de mot de passe par la commande vipw. , Et rechercher le login :

```
# vipw
olivier:$1$00edYNK2$heBNRz65txD0j7bGtB40r.:1001:1001::0:0:Olivier Cochard-
Labbe:/home/olivier:/bin/tcsh
```

Il faut ajouter «french» entre les ::, ce qui va donner :

```
olivier:$1$00edYNK2$heBNRz65txD0j7bGtB40r.:1001:1001:french:0:0:Olivier
Cochard-Labbe:/home/olivier:/bin/tcsh
```

Puis, pour que les futurs utilisateurs utilisent cette langue par défaut, on ajoute dans /etc/adduser.conf :

```
defaultclass = french
```

Configuration du clavier Bépo

Pour l'installation du clavier Bépo en mode console, les instructions pour FreeBSD sur le wiki du site clavier-bépo: ne sont plus valable depuis la migration de syscons vers VT (il faut transformer le fichier kbd):

```
fetch http://dev.bsdrp.net/mirrors/fr.bepo.kbd
cp fr.bepo.kbd /usr/share/vt/keymaps/
echo 'keymap="fr.bepo.kbd"' >> /etc/rc.conf
```

Permettre à l'utilisateur d'utiliser les périphériques USB, graveur CD/DVD

Commencer par permettre aux utilisateurs de monter des périphériques:

```
echo vfs.usermount=1 >> /etc/sysctl.conf
sysctl vfs.usermount=1
```

Et si on n'a pas mis l'utilisateur dans le groupe «operator» et «dialer» (accès aux ports séries) il faut le faire maintenant:

```
pw group mod operator -m <username>
pw group mod dialer -m <username>
```

Maintenant, il faut donner les droits en écriture pour les périphériques USB au groupe operator en créant le fichier /etc/devfs.rules:

```
cat >>/etc/devfs.rules <<EOF
[localrules=5]
add path 'da*' mode 0660 group operator
add path 'cd*' mode 0660 group operator
add path 'usanner*' mode 0660 group operator
add path 'xpt*' mode 660 group operator
add path 'pass*' mode 660 group operator
add path 'md*' mode 0660 group operator
add path 'msdosfs/*' mode 0660 group operator
add path 'ext2fs/*' mode 0660 group operator
add path 'ntfs/*' mode 0660 group operator
add path 'usb/*' mode 0660 group operator
EOF
```

Déclarer les nouvelles règles :

```
sysrc devfs_system_ruleset=localrules
```

Activer les nouvelles règles:

```
service devfs restart
```

Et vérifier leur applications:

```
devfs rule show
```

Optimisations

Configurer le scheduler pour un usage «desktop»

On commence par optimiser le scheduler pour un usage desktop:

```
sysrc -f /etc/sysctl.conf kern.sched.preempt_thresh=224
```

Réparation fsck automatique et background fsck (sauf si installation ZFS)

Pour activer la réponse «yes» aux questions de fsck au démarrage:

```
sysrc fsck_y_enable=YES
```

Désactivation de l'access time sur la partition /

Par défaut, l'heure d'accès aux fichiers est noté, ce qui peut consommer du temps (et ça ne sert pas à grand-chose).

Éditer le fichier /etc/fstab est ajouter l'option noatime:

#Device	Mountpoint	FSType	Options	Dump	Pass
/dev/gpt/ROOT	/	ufs	rw,noatime	1	1

Pourcentage réservé du disque

Par défaut, FreeBSD réserve 8% du disque pour ses opérations système (défragmentation), si cela est

trop pour vous, vous pouvez modifier cette valeur avec «tunefs -m X».

Éviter la création de fichier .core

Pour les fichiers core générés par les applications, ajouter au fichier /etc/csh.login:

```
limit coredumpsize 0
```

Pour les fichiers core générés par le noyau, ajouter dans le /etc/sysctl.conf :

```
kern.coredump=0
```

Installation des Mises à jour de FreeBSD

Cela se fait simplement par la commande suivante:

```
freebsd-update fetch install
```

Il faut utiliser le noyau GENERIC pour que cet outil mette aussi les binaires du noyau à jour.

Utiliser DMA à la place de sendmail

DragonFly Mail-Agent (inclus à partir de FreeBSD 11) permet de remplacer sendmail lorsque le besoin est uniquement de pouvoir envoyer des emails.

```
sysrc sendmail_enable=NONE
service sendmail onestop
cat >/etc/mail/mailer.conf <<EOF
# Execute dma instead of sendmail
sendmail      /usr/libexec/dma
mailq         /usr/libexec/dma
newaliases    /usr/libexec/dma
rmail         /usr/libexec/dma
EOF

cat >>/etc/periodic.conf <<EOF
#disable some sendmail specific daily maintenance routines
daily_clean_hoststat_enable="NO"
daily_status_mail_rejects_enable="NO"
daily_status_include_submit_mailq="NO"
daily_submit_queuerun="NO"
EOF
```

Reste la configuration de DMA, ici un exemple pour un compte gmail:

```
cat >/etc/dma/dma.conf <<EOF
SMARTHOST smtp.gmail.com
PORT 587
SECURETRANSFER
STARTTLS
AUTHPATH /etc/dma/auth.conf
MASQUERADE votre-login@gmail.com
EOF

echo 'votre-login|smtp.gmail.com:votre-mot-de-passe' > /etc/dma/auth.conf
```

Installation des logiciels externes (packages)

Remplacement du dépôt quarterly par latest

Une distribution -RELEASE utilise un dépôt de package mis à jour tous les 3 mois uniquement.

Pour ne pas attendre 3 mois, on peut configurer l'usage du dépôt latest (mis à jour tous les 2 jours environs) comme ceci:

```
sed -i "" -e 's/quarterly/latest' /etc/pkg/FreeBSD.conf
```

Puis il suffit de lancer la commande “pkg” pour qu'il s'auto-installe.

Utilisation de pkg

Une fois pkg installé, commencer par mettre à jour la liste des paquets présents sur le dépôt officiel:

```
pkg update
```

Et pour installer des nouveaux, il suffira de faire:

```
pkg install truc bidule machin chouette
```

Et pour les mettre à jour:

```
pkg upgrade
```

Installer les root CA (FreeBSD 12 ou plus vieux)

Un des premiers packages à installer est celui des root CA:

```
pkg inst ca_root_nss
```

Mise à jour des microcodes CPU

Corrige les grosses failles CPU en attendant la Maj de votre BIOS/EFI:

```
pkg install devcpu-data  
service microcode_update enable  
service microcode_update start
```

Remplacer powerd par powerd++

Une version plus évoluée (gère mieux les multi-core) de powerd existe sous le nom de powerd++:

```
pkg inst powerdxx  
service powerd disable  
service powerd onestop  
service powerdxx enable  
service powerdxx start
```

Montage automatique des disques amovibles

Pour la détection d'insertion d'un disque, son analyse et le montage de celui-ci on va utiliser automount et si l'on souhaite ajouter le support d'autre file systèmes: les packages fuse-[ntfs].

```
pkg install automount fusefs-ntfs fusefs-simple-mtpfs
```

Commencer par le chargement de fuse et créer un fichier de configuration (à éditer pour faire correspondre à votre environnement):

```
sysrc kld_list+=fusefs  
sysrc kld_list+=ext2fs  
kldload fusefs  
kldload ext2fs  
cp /usr/local/etc/automount.conf.sample /usr/local/etc/automount.conf
```

Puis on relance le service devd pour qu'il charge la configuration d'automount:

```
service devd restart
```

Désormais, dans le dossier /media on doit retrouver le nom des disques USB connectés.

Environnement graphique X11

xorg

Ajouter l'utilisateur au groupe "video" pour lui permettre d'accéder au DRI:

```
pw groupmod video -m $USERNAME
```

Puis installer xorg et le xscreensaver:

```
pkg install xorg xscreensaver  
Drivers vidéo KRM-DMS (Intel i915, AMD GPU, Radeon)
```

À partir de FreeBSD 11.2 uniquement, on peut installer le meta-package drm-kmod qui va sélectionner la meilleure version des drivers (legacy, stable, next, devel):

```
pkg install drm-kmod
```

Et demander leur chargement automatique, ici un exemple pour un module i915:

```
sysrc kld_list+=" /boot/modules/i915kms.ko"  
kldload /boot/modules/i915kms.ko
```

Après un reboot, vérifier que le module du package (/boot/modules) et non de la base (/boot/kernel) est bien pris en compte:

```
olivier@x220:~ % kldstat -v | grep i915  
6      1 0xffffffff8293d000 120ca8  i915kms.ko (/boot/modules/i915kms.ko)
```

Drivers vidéo X11 pour Intel (les anciennes)

Si la machine possède une vieille puce Intel, installer quand même les bibliothèques d'accélération vidéo:

```
pkg install libva-intel-driver
```


Ensuite tester que ça fonctionne: Et surtout tester aussi le suspend/resume.

Pour les utilisateurs de carte Intel Sandybridge (ou mieux), l'activation du mode SNA peut-être bénéfique (vérifier en mesurant la différence avec gtkperf par exemple).

Pour activer ce mode, un exemple de configuration (mais qui fait rebooter ma machine quand elle passe en suspend):

```
cat > /usr/local/etc/X11/xorg.conf.d/20-video.conf <<EOF
Section "Device"
Identifier "Intel Graphics"
Driver "intel"
Option "AccelMethod" "sna"
EndSection
EOF
```

Sous X11, l'utilitaire «vainfo» devrais indiquer une liste de profil vidéo (H264, etc...) pris en charge matériellement.

Drivers vidéo X11 pour nvidia

Pour un drivers nvidia, il y a plusieurs drivers en fonction de la génération de la carte graphique (nvidia-driver-173, nvidia-driver-304, nvidia-driver-340, nvidia-driver).

Dans le cas ou l'on dispose d'un ancien ION 1:

```
pkg install nvidia-driver-304
echo 'nvidia_load="YES"' >> /boot/loader.conf
mkdir -p /usr/local/etc/X11/xorg.conf.d/
cat > /usr/local/etc/X11/xorg.conf.d/20-video.conf <<EOF
Section "Device"
Identifier "NVIDIA Card"
VendorName "NVIDIA Corporation"
Driver "nvidia"
EndSection
EOF
```

dbus

Activation de ce service

```
service dbus enable
service dbus start
```

Par défaut, FreeBSD ne monte ni procfs ni fdesc. Mais certaines applications en ont besoin. On édite le fichier /etc/fstab et lui ajouter ces 2 lignes:

proc	/proc	procfs	rw,late	0	0
fdesc	/dev/fd	fdescfs	rw,late	0	0

L'option "late" évite les problèmes avec ZFS. Puis les monter:

```
mount /proc
mount /dev/fd/
```

Polkit (Chapitre désormais inutile?)

Concernant les logiciels utilisant Polkit (xfce par exemple), voici comment permettre à l'utilisateur d'éteindre et de mettre en veille la machine pour les utilisateurs du groupe "operator"

```
cat > /usr/local/etc/polkit-1/rules.d/51.shutdown.rules <<EOF
polkit.addRule(function (action, subject) {
if ((action.id == "org.freedesktop.consolekit.system.restart" ||
action.id == "org.freedesktop.consolekit.system.stop")
&& subject.isInGroup("operator")) {
return polkit.Result.YES;
}
});
EOF

cat > /usr/local/etc/polkit-1/rules.d/52.resume.rules <<EOF
polkit.addRule(function (action, subject) {
if (action.id == "org.freedesktop.consolekit.system.suspend"
&& subject.isInGroup("operator")) {
return polkit.Result.YES;
}
});
EOF

chown -R polkitd /usr/local/etc/polkit-1/
```

Puis ensuite, corriger un problème de permission manquantes concernant les actions .

On doit voir les erreurs suivantes dans le /var/log/messages:

```
Jul 18 00:21:28 x220 dbus[20382]: [system] Rejected send message, 2 matched
rules; type="method_call", sender=":1.19" (uid=1001 pid=63293 comm="")
interface="org.freedesktop.ConsoleKit.Manager" member="CanSuspend" error
name="(unset)" requested_reply="0" destination=":1.1" (uid=0 pid=44489
comm="")
Jul 18 00:21:28 x220 dbus[20382]: [system] Rejected send message, 2 matched
rules; type="method_call", sender=":1.19" (uid=1001 pid=63293 comm="")
interface="org.freedesktop.ConsoleKit.Manager" member="CanHibernate" error
name="(unset)" requested_reply="0" destination=":1.1" (uid=0 pid=44489
```

```
comm=" ")
```

Pour corriger ce problème (et déverrouiller les actions «Mise en veille» et «Mise en veille prolongée») depuis XFCE, il faut éditer le fichier `/usr/local/etc/dbus-1/system.d/ConsoleKit.conf`, et ajouter ces 6 lignes (en gras) entre les lignes existantes:

```
<deny send_destination="org.freedesktop.ConsoleKit"
      send_interface="org.freedesktop.DBus.Properties" />
<allow send_destination="org.freedesktop.ConsoleKit"
       send_interface="org.freedesktop.ConsoleKit.Manager"
       send_member="CanSuspend"/>
<allow send_destination="org.freedesktop.ConsoleKit"
       send_interface="org.freedesktop.ConsoleKit.Manager"
       send_member="CanHibernate"/>
<allow send_destination="org.freedesktop.ConsoleKit"
       send_interface="org.freedesktop.ConsoleKit.Manager"
       send_member="Restart"/>
```

Pavé Synaptics, Trackpoint (Lenovo) ou Elantec

À partir de FreeBSD 12.1 il est fortement conseillé d'utiliser le mode EVDEV.

Mais il faut dire à celui-ci que l'on dispose de ce type d'équipement (synaptics ou elantec):

```
echo 'hw.psm.synaptics_support="1"' >> /boot/loader.conf
```

Pour un trackpoint ou un elantec il faut remplacer le mot "synaptics" par le type de pointeur.

Puis d'activer le mode evdev:

```
echo 'kern.evdev.rcpt_mask=6' >> /etc/sysctl.conf
sysctl kern.evdev.rcpt_mask=6
```

XFCE4

```
pkg install xfce4 xfce4-goodies
```

SLIM

Ceux qui n'utiliseront ni gnome ni KDE peuvent utiliser slim comme gestionnaire de session.

Installation

Voici comment le faire fonctionner correctement:

```
pkg inst slim
```

Puis créer un fichier .xinitrc de ce style pour un lancement de xfce4 par exemple:

```
echo "exec startxfce4" >.xinitrc
```

Activer son lancement automatique:

```
service slim enable
```

Thème FreeBSD

Voici comment y installer un joli thème FreeBSD:

```
pkg install slim-themes
```

Et éditer le fichier /usr/local/etc/slim.conf pour remplacer la ligne:

```
current_theme      default
```

Par

```
current_theme      fbsd
```

Disposition du clavier sous slim

Pour préciser le type de clavier sous slim, qui démarre avant votre environnement graphique (xfce, lxde, gnome, etc.) il n'y a pas d'autre choix que définir des fichiers de configuration dans /usr/local/etc/X11/xorg.conf.d/ que l'on remplit par le contenu suivant:

```
cat > /usr/local/etc/X11/xorg.conf.d/10-keyboard.conf <<EOF
Section "InputClass"
Identifier "keyboard Layout"
Driver "libinput"
MatchIsKeyboard "on"
Option "XkbLayout" "fr"
EndSection
```

EOF

Pour du bepo, il suffit de préciser la variante à la clé XkbLayout:

```
Option "XkbLayout" "fr(bepo)"
```

XFCE

Un environnement de bureau léger (pour qu'il prenne en compte .Xresources il faut le lancer avec xfce4-session et non pas avec startxfce4):

```
pkg install xfce xfce4-mixer xfce4-volumed xscreensaver
```

Police de caractères

Puis on installe de jolies polices de caractères:

```
pkg install dejavu urwfonts bitstream-vera webfonts terminus-font ubuntu-font-consolamono-ttf inconsolata-ttf
```

On vérifie ensuite quelles sont bien prises en compte dans xorg par la commande:

```
fc-list
```

Si on rencontre un problème d'anti-aliasing avec la police Helvetica:

```
cat > /usr/local/etc/fonts/local.conf <<EOF
<?xml version='1.0'?>
<!DOCTYPE fontconfig SYSTEM 'fonts.dtd'>
<fontconfig>
  <dir>~/.fonts</dir>
  <!-- do not use the embedded bitmap instead of the outline
<https://www.freebsd.org/cgi/man.cgi?query=fonts-conf&sektion=5&manpath=Free
BSD+and+Ports>
    <https://bbs.archlinux.org/viewtopic.php?id=161609> post 2 (2013)
    <https://redd.it/7kqr5l> (2017) -->
  <match target="font">
    <edit name="embeddedbitmap" mode="assign">
      <bool>>false</bool>
    </edit>
  </match>
  <!-- prefer outline e.g. TrueType instead of bitmap fonts
    <https://bbs.archlinux.org/viewtopic.php?id=161609> post 2 (2013)
```

```
<https://redd.it/4tb2dt> (2016) -->
<match target="font">
  <edit name="prefer_outline">
    <bool>true</bool>
  </edit>
</match>
<!-- reject bitmap fonts, except Ohsnapu - prefer PostScript,
TrueType et cetera
<https://forums.freebsd.org/threads/howto-nice-fonts.2021/> (2009)
<https://redd.it/4tb2dt> (2016) -->
<selectfont>
  <acceptfont>
    <pattern>
      <patelt name="family">
        <string>Ohsnapu</string>
      </patelt>
    </pattern>
  </acceptfont>
  <rejectfont>
    <pattern>
      <patelt name="scalable">
        <bool>>false</bool>
      </patelt>
    </pattern>
  </rejectfont>
</selectfont>
</fontconfig>
EOF
```

Les applications

Webcam

L'installation du support de webcam demande plusieurs étapes.

Commencer par installer les logiciels nécessaires:

```
pkg install webcamd v4l_compat libv4l
```

Puis activer webcamd:

```
service webcamd enable
```

Ajouter le module cuse à la liste kld_load:

```
sysrc kld_list+="cuse"
```

```
kldload cuse
```

Une fois terminé, ajouter l'utilisateur au groupe webcamd:

```
pw groupmod webcamd -m votre-utilisateur
```

Afficher la liste des périphériques utilisable:

```
webcamd -l
```

Et teste la webcam, ici un exemple avec une webcam d'un PC portable Lenovo:

```
~ # webcamd -N Azurrewave-Lenovo-EasyCamera  
Attached to ugen1.4[0]  
Creating /dev/video0
```

Démarrer le daemon:

```
service webcamd start
```

Pour tester la webcam, pwcview est très simple.

Si webcamd n'est pas capable de selectionner automatiquement la bonne webcam, il faut déclarer un flag en récupérant le nom de la webcam (flag -N de la sortie "webcamd -l):

```
sysrc webcamd_0_flags="-N Azurrewave-Lenovo-EasyCamera"  
sysrc webcamd_1_flags="-N vendor-0x046d-HD-Pro-Webcam-C920"
```

Firefox

On l'installe en français par ces deux commandes:

```
pkg install firefox firefox-i18n
```

Chromium

On l'installe par cette commande:

```
pkg install chromium
```

mpv

Lecteur vidéo avec excellent support d'accélération matériel VA-API (carte intel), que l'on installe, avec la librairie VDPAU/VA-API et configure comme ceci:

```
pkg install mpv libvdpau-va-gl
mkdir ~/.config/mpv
cat > ~/.config/mpv/mpv.conf <<EOF
vo=vaapi
hwdec=vaapi
fs=yes
EOF
```

LibreOffice

Comme pour les autres:

```
pkg install fr-libreoffice
```

Il est à installer l'extension «dictionnaire français» et l'extension de «correction grammaticale».

CUPS

Passons au support de l'impression par l'installation cups et à l'ajout de votre utilisateur dans le groupe cups:

```
pkg install cups cups-filters
```

```
pw groupmod cups -m $USERNAME
```

Puis, pour une imprimante locale (USB) éditer le fichier /etc/devfs.rules et ajoutez:

```
[localrules=10]
add path 'unlpt*' mode 0660 group cups
add path 'ulpt*' mode 0660 group cups
add path 'lpt*' mode 0660 group cups
add path 'usb/X.Y.Z' mode 0660 group cups
```

En remplaçant X.Y.Z par l'identifiant USB correspondant à l'imprimante (cherchez le "ugenX.Y" dans votre dmesg).

Et activer cupsd et ces règles devfs:


```
sysrc devfs_system_ruleset=localrules  
service cupsd enable
```

Il est possible d'installer des drivers supplémentaires pour cups. Comme je possède par exemple une imprimante Samsung ML-2955ND, j'installe en plus print/splix.

Puis on active le tout:

```
service devfs restart  
service cupsd restart
```

Ne reste plus qu'à ajouter l'imprimante en passant par l'interface d'administration de cups
<http://localhost:631>



Si on a activé le firewall cups ne trouvera pas automatiquement l'imprimante réseau.

Un petit tour pour récupérer le dernier PPD correspondant à l'imprimante sur openprinting.org sera sûrement obligatoire.

Gérer la présence d'un proxy HTTP/HTTPS

Pour l'utilisateur sous gnome: Système -> Préférence -> Serveur Mandataire

Pour une session sous l'utilisateur root (qui ne prend pas en compte le paramétrage de gnome), entrer ces deux commandes (la deuxième ne sert que pour l'authentification d'un proxy):

```
setenv HTTP_PROXY "10.0.0.10:8080"  
setenv HTTP_PROXY_AUTH "basic*:login:password"  
setenv http_proxy 'http://login:password@10.0.0.10:8080/'
```

Pour utiliser le proxy pour la compilation des ports:

```
echo "FETCH_ARGS=-pRr" >> /etc/make.conf  
echo "FETCH_ENV=http_proxy='http://login:password@10.0.0.10:8080/' "  
>> /etc/make.conf
```



Pour utiliser subversion, éditer le fichier ~/.subversion/servers et y ajouter les lignes suivantes:

```
http-proxy-host = proxy-hostname  
http-proxy-port = proxy-port  
http-proxy-username = username  
http-proxy-password = password
```

Pour faire des sessions SSH, installer l'utilitaire /usr/ports/net/corkscrew.

Pour utiliser les outils qui ne gèrent pas les proxy nativement (csup par exemple) il faut utiliser des tunnels SSH ou /usr/ports/net/prtunnel puis lancer le tunnel:

1) Regarder dans le /etc/standard-supfile à l'entrée default host= pour connaître le nom



du serveur cvsup utilisé

```
2) prtunnel -t http -H proxy.de.entreprise -P 3128 -D 5999
   cvsup2.FreeBSD.org 5999
```

```
3) Editer le /etc/standard-supfile (ou celui que vous utilisez) et modifier l'entrée default
   host par: default host=127.0.0.1
```

```
4) lancer le csup /etc/standard-supfile
```

```
5) Puis tuer le tunnel: killall -9 prtunnel
```

Serveur TFTP

La configuration d'un serveur TFTP est très simple:

- Créer le dossier /tftpboot
- Mettre les fichiers à partager dans le dossier /tftpboot
- Editer le fichier /etc/inetd.conf et décommenter la ligne tftp
- Activer inetd: `service inetd enable`
- Puis lancer inetd: `service inetd start`

Le noyau de FreeBSD refuse les requêtes des clients TFTP utilisant le port source 0

Et comme aucun RFC n'interdit au client TFTP d'utiliser le port 0 comme source, il existe des équipements qui le font et ne fonctionneront donc pas avec un serveur TFTP sous FreeBSD.

C'est le cas par exemple des clients TFTP du boot-loader des firewalls Juniper Netscreen ISG-1000 et ISG-2000 qui vont générer ce type de message dans votre



/var/log/messages:

```
tftpd[4894]: DATA block 1, attempt 0 failed (Error 49: Can't assign
requested address)
```

Si on veut modifier ce comportement, il faut commenter les 2 lignes suivantes de la fonction `inpcbconnectsetup()` du fichier `/usr/src/sys/netinet/in_pcb.c`:

```
if (sin->sin_port == 0)
return (EADDRNOTAVAIL);
```

Et recompiler le noyau...

Émulateur de terminal série (équivalent de l'hyperterminal ou teraterm)

Il existe 2 outils inclus dans **FreeBSD** pour cela: `cu` et `tip`

Par contre, attention car il existe deux représentations des ports séries (man uart):

- **cuauX**, pour les appels sortants
- **ttyuX**, pour les appels entrants

cu s'utilise avec des options en ligne de commande, par exemple:

```
cu -l /dev/cuau0 -s 115200
```

tip lui utilise le fichier de configuration /etc/remote.

Personalisation avancée

Collecte d'information

Inventaire matériel

La liste des équipements PCI détectés s'affiche par la commande:

```
pciconf -lv
```

Celle des équipements USB par:

```
usbconfig
```

Liste des équipements détectés par HAL:

```
lshal
```

Charge système par des graphiques: systat

Aperçu de la charge système par la commande:

```
systat -vm
```

Mais cette commande permet aussi de connaître la charge réseau actuelle avec la commande:

```
systat -ip
```

Charge système en mode compact: vmstat

Pratique pour connaître ou le goulot d'étranglement:

```
[olivier@laptop]~>vmstat
procs      memory      page      disk  faults      cpu
 r b w      avm    fre flt  re  pi  po    fr  sr ad4    in  sy    cs us sy
id
 1 0 0    4831M    209M 14907    1    1    0 14725   34    0   663 13745 3511 23   7
70
```

Et pour un rafraichissement toute les 2 secondes:

```
vmstat -w 2
```

Et pour connaitre qui consomme les interruptions matérielles (la colonne rate est importante):

```
> vmstat -i
interrupt                total        rate
irq1: atkbd0              6483          3
irq9: acpi0               3236          1
irq12: psm0              347988        167
irq14: ata0              16431          7
irq16: cbb0 uhci2+      13624982      6556
irq20: uhci0               14            0
irq22: ehci0                2            0
cpu0: timer             4154687      1999
irq256: em0               53736         25
irq257: hdac0              5797           2
cpu1: timer             4153683      1998
irq258: vgapci0           235585        113
Total                   22602624     10877
```

Sur cet exemple, on note une tempête d'interruption au niveau du ccb (cardbus), mais c'est en fait un bug lié à la carte graphique.

Charge des disques: gstat

La couche de gestion des disques est geom, l'outil pour afficher le taux d'utilisation est gstat.

Consommation électrique

Sur processeur Intel, le package “intel-pcm” dispose de l'outil pcm.x qui affiche la consommation électrique du CPU en temps réel:

Core (SKT)	EXEC	IPC	FREQ	AFREQ	L3MISS	L2MISS	L3HIT	L2HIT	
L3MPI	L2MPI	TEMP							
0	0	0.00	0.26	0.00	0.32	15 K	42 K	0.65	0.02
0.01	0.02	55							
1	0	0.00	0.71	0.00	0.32	7329	12 K	0.40	0.10

```

0.00    0.01    55
  2    0    0.00  0.30  0.00  0.32    13 K    38 K    0.63    0.04
0.01    0.02    55
  3    0    0.00  0.48  0.00  0.32    7683    15 K    0.50    0.03
0.01    0.01    55
-----
SKT    0    0.00  0.36  0.00  0.32    44 K    108 K    0.59    0.04
0.01    0.02    54
-----
TOTAL  *    0.00  0.36  0.00  0.32    44 K    108 K    0.59    0.04
0.01    0.02    N/A

```

Instructions retired: 7083 K ; Active cycles: 19 M ; Time (TSC): 2651 Mticks ; C0 (active,non-halted) core residency: 0.59 %

C1 core residency: 0.81 %; C3 core residency: 0.00 %; C6 core residency: 98.61 %; C7 core residency: 0.00 %;

C2 package residency: 48.12 %; C3 package residency: 0.00 %; C6 package residency: 49.64 %; C7 package residency: 0.00 %;

PHYSICAL CORE IPC : 0.71 => corresponds to 17.83 % utilization for cores in active state

Instructions per nominal CPU cycle: 0.00 => corresponds to 0.03 % core utilization over time interval

```

-----
|  READ  |  WRITE  |   IO   | CPU energy |
-----
SKT    0    0.29    0.00    0.29    3.76
-----

```

Usage de ntfs-3g dans le fstab

Voici un exemple de ligne du fstab pour le montage d'un disque NTFS une fois installé fuse ntfs-3g:

```

/dev/ntfs/System    /media/windows    ntfs
rw,mountprog=/usr/local/bin/ntfs-3g,late 0 0

```



umount failed: Device busy: Lorsqu'il est impossible de démonter des volumes, alors qu'on ne les utilise plus:

```

[root@laptop]~#umount /usr/obj/nanobsd.full/_mnt/
umount: unmount of /usr/obj/nanobsd.full/_mnt failed: Device busy

```

Il faut identifier le process qui empêche de démonter ce dossier, pour éventuellement le tuer:

```
[root@laptop]~#fstat /usr/obj/nanobsd.full/_mnt/
USER CMD PID FD MOUNT INUM MODE SZ|DV R/W NAME
olivier gam_server 4264 36 /usr/obj/nanobsd.full/_mnt 2 drwxr-xr-x
512 r /usr/obj/nanobsd.full/_mnt/
```



Identifier le processus qui tient le dossier:

- 1) **locate <nom du service>** permet d'identifier le programme
- 2) **pkg wich <path du programme>** permet d'identifier le paquet
- 3) **pkg info -D <nom du paquet>** permet de fournir des infos sur le paquet

Compilation des sources

FreeBSD avec la pile TCP BBR

Si on souhaite la dernière version de FreeBSD avec la pile TCP BBR¹⁾ et un OS optimisé pour le processeur:

- Téléchargement des sources

```
pkg install git
git clone -o freebsd https://git.freebsd.org/src.git -b main /usr/src` `
```

- Activation de CPUTYPE?=native et Déclaration d'un kernel spécifique

```
cat > /etc/make.conf <<EOF
CPUTYPE?=native
KERNCONF=BBR
WITH_META_MODE=yes
WITH_DIRDEPS_BUILD=yes
WITH_DIRDEPS_CACHE=yes
WITHOUT_LIB32=yes
WITHOUT_LLVM_ASSERTIONS=yes
WITH_MALLOC_PRODUCTION=yes
MALLOC_PRODUCTION=yes
DEVELOPER=yes
EOF
```

- Désactivation des fonctionnalités de débogage qui sont activé dans la version de dev

```
cat >/usr/src/sys/amd64/conf/BBR <<EOF
include GENERIC-NODEBUG
ident          BBR
options        TCPHPTS
```

```
options      RATELIMIT
makeoptions  WITH_EXTRA_TCP_STACKS=1
options      KDB_UNATTENDED
EOF
```

- Compilation et installation avec les BE (Boot Environment ZFS)

```
cd /usr/src
make -j 32 buildworld buildkernel
tools/build/beinstall.sh
```

- Redémarrage du système

```
shutdown -r now

echo "net.inet.tcp.functions_default=bbr" >> /etc/sysctl.conf
sysrc kld_list+="filemon tcp_rack tcp_bbr"
kldload tcp_bbr && sysctl net.inet.tcp.functions_default=bbr
```

Désormais pour régulièrement mettre à jour FreeBSD:

```
cd /usr/src
git pull && make -j 32 buildworld buildkernel && tools/build/beinstall.sh
shutdown -r now
```

Compilation batch des ports

Lorsqu'on veut des ports compilés avec les optimisations du processeur, l'outil **poudriere** est utilisé pour construire des packages à partir de l'arborescence des ports:

- Installation et configuration de poudriere

```
pkg install poudriere-devel
cat >> /usr/local/etc/poudriere.conf <<EOF
ZPOOL=zroot
FREEBSD_HOST=https://download.FreeBSD.org
ALLOW_MAKE_JOBS=yes
EOF
```

- Initialisation de poudriere par la création de la jail utilisé pour générer les ports et le téléchargement de l'arbre des ports

```
mkdir -p /usr/ports/distfiles
poudriere jail -c -j builder -m src=/usr/src
poudriere ports -c -U https://git.freebsd.org/ports.git -m git -B main
```

- Personnalisation du make.conf utilisé pour compiler l'ensemble des ports

```
echo 'CPUTYPE?=native' >> /usr/local/etc/poudriere.d/make.conf
```

- Création de la liste des ports (pour connaître la catégorie, exemple: `pkg rquery '%o' poudriere-devel`)

```
cat > /root/packages.list <<EOF
sysutils/tmux
devel/git
ports-mgmt/poudriere-devel
editors/vim-console
lang/python
x11/xorg-apps
x11-wm/xfce4
x11/xfce4-goodies
graphics/gimp
multimedia/vlc
graphics/drm-current-kmod
www/chromium
editors/libreoffice
french/libreoffice
security/sudo
sysutils/vm-bhyve
x11/slim
multimedia/webcamd
sysutils/ipmitool
x11/nvidia-xconfig
x11/nvidia-settings
x11-drivers/xf86-video-nv
x11/nvidia-driver
x11-drivers/xf86-video-vesa
x11-drivers/xf86-video-scfb
x11-fonts/dejavu
x11-fonts/bitstream-vera
x11-fonts/webfonts
x11-fonts/terminus-font
x11-fonts/ubuntu-font
x11-fonts/consolamono-ttf
x11-fonts/inconsolata-ttf
x11-fonts/urwfonts
EOF
```

- Lancement de la compilation des ports

```
poudriere bulk -j builder -f /root/packages.list
```

- Configuration d'un dépôt local et désactivation du dépôt officiel

```
mkdir -p /usr/local/etc/pkg/repos/
cat > /usr/local/etc/pkg/repos/local.conf <<EOF
local: {
```



```
url: "file:///usr/local/poudriere/data/packages/builder-default/",
signature_type: "none",
assume_always_yes: true,
enabled: yes
}
EOF
```

```
sed -i '' 's/yes/no/' /etc/pkg/FreeBSD.conf
```

1)

L'algorithme TCP BBR permet de considérablement augmenter le débit et réduire la latence des connexions sur les réseaux

From:

<http://vps101364.serveur-vps.net/> - **dwndoc**

Permanent link:

<http://vps101364.serveur-vps.net/doku.php?id=prive:freebsd-desktop>

Last update: **2025/02/19 10:59**

