

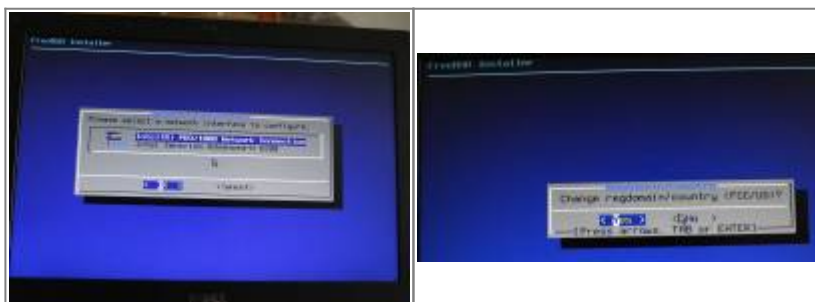
# Installation de FreeBSD 12.0 / 12.1

Sur laptop, DELL Latitude E5430 et DELL Latitude E6410

Testé avec un schéma de partition UFS et ZFS, en mode UEFI et LEGACY.

## Installation

Récupérer l'iso de FreeBSD. La graver sur une clef USB et booter dessus. Suivre l'installation pas à pas.



Faire attention au stade de configuration réseau de choisir pour ETSI / FR "country FR regdomain ETSI"



"country FR regdomain ETSI"

Voici les options que j'ai choisies lors de **System Configuration**



Rebooter sur FreeBSD

## Francisation

Editer le fichier **/etc/login.conf**

```
default:\n    :passwd_format=sha512:\n    :copyright=/etc/COPYRIGHT:\n    :welcome=/etc/motd:\n
```

```

: setenv=MAIL=/var/mail/$,BLOCKSIZE=K:\
: path=/sbin /bin /usr/sbin /usr/bin /usr/local/sbin /usr/local/bin
~/bin:\
: nologin=/var/run/nologin:\
: cputime=unlimited:\
: datasize=unlimited:\
: stacksize=unlimited:\
: memorylocked=64K:\
: memoryuse=unlimited:\
: filesize=unlimited:\
: coredumpsize=unlimited:\
: openfiles=unlimited:\
: maxproc=unlimited:\
: sbsize=unlimited:\
: vmemoryuse=unlimited:\
: swapuse=unlimited:\
: pseudoterminals=unlimited:\
: kqueues=unlimited:\
: umtxp=unlimited:\
: priority=0:\
: ignoretime@:\
: charset=UTF-8:\
: lang=fr_FR.UTF-8:\
: umask=022:

```

Ajouter pour que les locales soient prises en compte

```
: charset=UTF-8:\n: lang=fr FR.UTF-8:\n
```

## Le recharger

```
cap mkdb /etc/login.conf
```

## Initialisation de pkg



## AJOUTER un user

```
# adduser
```

L'inclure dans les groupes : **wheel operator video**

Editer .login\_conf

```
[ragnarok@asgard-freebsd:~] $ cat .login_conf
# $FreeBSD: releng/12.0/share/skel/dot.login_conf 77995 2001-06-10 17:08:53Z
ache $
#
# see login.conf(5)
#
me:\
    :charset=UTF-8:\
    :lang=fr_FR.UTF-8:
```

## Gestion du son

```
# kldload snd_driver
```

```
cat /dev/sndstat
Installed devices:
pcm0: <IDT 92HD93BXX (Analog 2.0+HP/2.0)> (play/rec) default
pcm1: <IDT 92HD93BXX (Analog)> (play/rec)
pcm2: <Intel Panther Point (HDMI/DP 8ch)> (play)
pcm3: <Intel Panther Point (HDMI/DP 8ch)> (play)
pcm4: <Intel Panther Point (HDMI/DP 8ch)> (play)
No devices installed from userspace.
```

```
# sysrc kld_list+=snd_hda
```

## Affichage : drm-kmod

Installer le paquet

```
pkg install drm-kmod
```

Initialisation au boot

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

Le charger

```
kldload /boot/modules/i915kms.ko
```

Vérifier

```
kldstat -v | grep i915
```

# Network Manager

## Network Manager

```
pkg show networkmgr
WWW           : https://github.com/GhostBSD/networkmgr
Comment      : FreeBSD/GhostBSD network connection manager
Annotations  :
  repo_type   : binary
  repository  : FreeBSD
Flat size    : 1.71MiB
Description  :
NetworkMgr is an open source, Network Manager based on the look of the Linux
Network Manager user interface. It use ifconfig and netif if make all work.
```

Il faudra ensuite configurer DOA pour que ça fonctionne

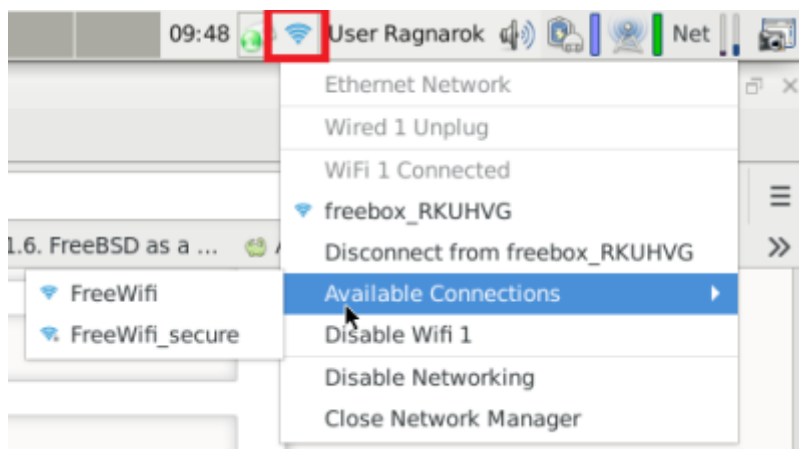
```
cat /usr/local/etc/doas.conf

permit nopass keepenv root
permit :wheel
permit nopass keepenv :wheel cmd netcardmgr
permit nopass keepenv :wheel cmd ifconfig
permit nopass keepenv :wheel cmd service
```

Se delogguer, l'utilisateur doit faire partie du groupe **wheel**.

Gestion de la connexion par ce fichier

```
cat /etc/wpa_supplicant.conf
```



## CLI graphique de gestion

```
pkg install wifimgr
```



## Devices : permissions

Permettra en autres l'accès aux périphériques amovibles (CDR,DVD). Nécessaire pour xfburner par exemple.

```
cat /etc/devfs.conf
```

```
# Allow all users to access optical media
```

```
perm    /dev/acd0      0666
perm    /dev/acd1      0666
perm    /dev/cd0       0666
perm    /dev/cd1       0666
```

```
# Allow all USB Devices to be mounted
```

```
perm    /dev/da0      0666
perm    /dev/da1      0666
perm    /dev/da2      0666
perm    /dev/da3      0666
perm    /dev/da4      0666
perm    /dev/da5      0666
```

```
# Misc other devices
```

```
perm    /dev/pass0    0666
perm    /dev/xpt0     0666
perm    /dev/uscanner0 0666
perm    /dev/video0   0666
perm    /dev/tuner0   0666
perm    /dev/dvb/adapt0/demux0 0666
perm    /dev/dvb/adapt0/dvr 0666
perm    /dev/dvb/adapt0/frontend0 0666
```

```
cat /etc/devfs.rules
```

```
[devfsrules_common=7]
```

```
add path 'ad[0-9]*'      mode 666
add path 'ada[0-9]*'    mode 666
add path 'da[0-9]*'      mode 666
add path 'acd[0-9]*'     mode 666
```

```

add path 'cd[0-9]*'      mode 666
add path 'mmc[0-9]*'     mode 666
add path 'pass[0-9]*'    mode 666
add path 'xpt[0-9]*'     mode 666
add path 'ugen[0-9]*'    mode 666
add path 'usbctl'        mode 666
add path 'usb/*'         mode 666
add path 'lpt[0-9]*'     mode 666
add path 'ulpt[0-9]*'    mode 666
add path 'unlpt[0-9]*'   mode 666
add path 'fd[0-9]*'      mode 666
add path 'uscan[0-9]*'   mode 666
add path 'video[0-9]*'   mode 666
add path 'tuner[0-9]*'   mode 666
add path 'dvb/*'         mode 666
add path 'cx88*' mode 0660
add path 'cx23885*' mode 0660 # CX23885-family stream configuration
device
add path 'iicdev*' mode 0660
add path 'uvisor[0-9]*' mode 0660

```

Mettre à jour le **/etc/rc.conf** avec cette entrée

```
devfs_system_ruleset="devfsrules_common"
```

Prendre en compte les modifications sans rebooter

```
/etc/rc.d/devfs restart
```

## Pour la lecture des DVD

Il faudra penser à installer le paquet libdvdcss

```
# pkg install libdvdcss
```

## Pour la monter un [ DVD | CD ]

```
dmesg | egrep -i 'cd|dvd'
```

Retournera le device un cd

```
mount -t udf /dev/cd0 /mnt
```

un dvd

```
mount -t udf /dev/cd0 /mnt
```

(PS besoin de monter le DVD sous VLC, il le gère)

Pour ejecter le CD | DVD

```
eject /dev/cd0
```

## Powerd++

Gestion des processeurs multicores.

Desinstaller **powerd**

```
# service powerd stop
Stopping powerd.
Waiting for PIDS: 50127.
# service powerd disable
powerd disabled in /etc/rc.conf
```

Ensuite on peut installer et lancer **powerd++** :

```
# service powerdxx enable
powerdxx enabled in /etc/rc.conf
# service powerdxx start
Starting powerdxx.
```

## X11 / XFCE

```
pkg install xorg xfce xfce4-goodies xfce4-mixer xfce4-volumed xf86-video-
intel file-roller
```

## Pour activer le suspend et resume

Créer ces 2 fichiers

**/usr/local/etc/polkit-1/rules.d/51.shutdown.rules**

```
polkit.addRule(function (action, subject) {
  if ((action.id == "org.freedesktop.consolekit.system.restart" ||
    action.id == "org.freedesktop.consolekit.system.stop")
    && subject.isInGroup("PUTYOURGROUPHERE")) {
    return polkit.Result.YES;
  }
});
```

**/usr/local/etc/polkit-1/rules.d/52.resume.rules**

```
polkit.addRule(function (action, subject) {  
    if (action.id == "org.freedesktop.consolekit.system.suspend"  
        && subject.isInGroup("PUTYOURGROUPHERE")) {  
        return polkit.Result.YES;  
    }  
});
```

puis

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

## Démarrer xfce

On aura besoin de dbus

```
Pkg install dbus  
service dbus onestart
```

Démarrage automatique, ajouter

```
cat /etc/rc.conf  
dbus_enable="YES"
```

On utilisera xinit pour le lacement de XFCE Créer dans le Home un **.xinitrc**

```
# SET PROPER locale  
export LANG="fr_FR.UTF-8"  
export LC_ALL="fr_FR.UTF-8"  
export LC_MESSAGES="fr_FR.UTF-8"  
  
# Clavier français  
setxkbmap fr &  
# Lancer xfce  
/usr/local/bin/startxfce4
```

Lier xinitrc et xsession

```
ln -s ~/.xinitrc ~/.xsession
```

Enfin lancer la session X

```
xinit .xinitrc
```

Pour démarrer la session de XFCE, j'ai choisi la façon manuelle sinon on peut passer, entre autre par SLIM. J'ai donc créer cet alias

```
alias x='xinit ~/.xinitrc'
```



et la commande **x** suffit au démarrage de XFCE

Sinon : jeter un oeil à SLIM : [Slim](#)

## anti-aliasing avec la police Helvetica

Editer ce fichier ainsi : **/usr/local/etc/fonts/local.conf**

```
<?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=FreeBSD+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>
```

## Gestionnaire WIFI

Bien que par défaut, on n'en a pas besoin. On peut vouloir utiliser un outil style Gestionnaire Wifi

```
pkg install wifimgr
```

Sinon la(es) connexion(s) se gère(nt) via :

```
cat /etc/wpa_supplicant.conf
```

## Barre de Menu pour XFCE

J'utilise Plank, simple et convivial.

```
# pkg install plank
```

Ajouter cette commande au démarrage de XFCE

## Paramétrage de Xorg

### Carte video

Créer ces 2 fichiers

```
# cat /usr/local/etc/X11/xorg.conf.d/card.conf
Section "Device"
    Identifier "Card0"
    Driver "intel"
    Option "DPMS"
    Option      "Backlight"  "intel_backlight"
EndSection
```

### Le clavier

```
# cat /usr/local/etc/X11/xorg.conf.d/keyboard.conf
Section "InputDevice"
    Identifier "Keyboard0"
```

```
Driver "kbd"  
Option "XkbLayout" "fr"  
Option "XkbOptions" "terminate:ctrl_alt_bksp,ctrl:nocaps"  
EndSection
```

## Configuration spécifique

### timeout boot

Diminuer le timeout du menu du boot loader à 2 secondes

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

### Acc. chiff.

Accélération de chiffrement des processeurs modernes

```
#echo 'aesni_load="YES"' >> /boot/loader.conf
```

Module à charger dans /boot et non /etc/rc.conf pour un chargement plus rapide

### Scheduler

Configuration pour une utilisation pour desktop/laptop

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

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

```
sysrc fsck_y_enable=YES
```

### Micro codes CPU

Utiliser micro codes CPU

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

### ACPI

## CPU et TEMP

device driver for Intel Core on-die digital thermal sensor cpuctl pseudo device

```
sysrc kld_list+=coretemp
sysrc kld_list+=cpuctl
```

## Fn Keys : DELL latitude E5430

Ai rencontré un soucis avec les touches **Fn + brightness** seulement sur le Latitude E5430. Avec le Latitude E6410 pas de soucis.

Les touches **FN et luminosité -up/down** ne fonctionnent pas. De plus **xbindkeys** ne retourne aucun code avec l'association **Fn + Brightness UP / Down** (Pas de Mod 2 + quelque chose)

Solution adoptée.

Installer :

```
pkg install xbindkeys intel-backlight
```

Fichier de configuration :

```
cat .xbindkeysrc
# PERSONNAL XBINDKEYS
"intel_backlight decr 10"
Alt + Down

"intel_backlight incr 10"
Alt + Up
```

Lancer xbindkeys au démarrage de la session ~/.xinitrc

Ajouter cette section avant le démarrage de XFCE dans

```
# Démarrer xbindkeys
/usr/local/bin/xbindkeys &
```

## XFCE / FLUXBOX / OPENBOX

Pour avoir le choix entre plus Windows Manager : ici fluxbox, openbox

```
pkg install openbox openbox-themes obconf fluxbox fluxbox-tenr-styles-pack
icewm wmakerconf windowmaker blackbox
```

Modifier le fichier ~/.xinitrc de cette façon

```
# GESTIONNAIRE X
```

```
DEFAULT_SESSION=startxfce4

case $1 in
o)
    exec openbox
    ;;
f)
    exec fluxbox
    ;;
i)
    icewmbg &
    icewmtray &
    exec icewm
    ;;
w)
    exec wmaker
    ;;
b)
    exec blackbox
    ;;
*)
    exec $DEFAULT_SESSION
    ;;
esac
```

Dans votre shell, créer l'alias suivant

```
alias x='xinit ~/.xinitrc'
```

A la connexion, un **x** lancer xfce, **x o** openbox **x b** blackbox....

## Monter les périphériques amovibles

Attention la version d'automount des repos ne permet pas de monter du FAT32. Utiliser cette procédure.



Attention sous FreeBSD **12.1**, **fusefs-ext4fuse** n'est plus disponible. Utiliser **sysutils/fusefs-ext2** à la place (ce module permet le montage des ext2, ext3 et ext4 en lecture / écriture).

## Installer git

```
# pkg install git
```

```
# git clone https://github.com/vermaden/automount.git
```

## Initialisation des ports

(nécessaires pour compiler exfat-utils/exFAT et sysutils/fusefs-exfat)

```
# portsnap fetch
# portsnap extract
```

Compilation de fusefs-exfat

```
# cd /usr/ports/sysutils/fusefs-exfat/
# make install clean
```

Accepter la licence et toutes les options par défaut

Compilation de exfat-utils

```
# cd /usr/ports/sysutils/exfat-utils/
# make install clean
```

Même procédure

Installer le reste des dépendances via pkg

```
# pkg install fusefs-ntfs fusefs-ext4fuse fusefs-hfsfuse fusefs-lkl fusefs-simple-mtpfs zenity
```

Détarer automount-1.6.1.tar.gz récupéré du git , entrez dans le dossier

Editer ce fichier de cette façon

```
$ cat /usr/local/etc/automount.conf
USERMOUNT=YES
ATIME=NO
REMOVEDIRS=YES
FM="thunar"
USER=ragnarok
ENCODING=fr_FR.UTF-8
CODEPAGE=cp850
```

Installation manuelle d'automount

```
# cp automount.conf /usr/local/etc/automount.conf
# cp automount_devd.conf /usr/local/etc/devd/automount_devd.conf
# cp automount /usr/local/sbin/automount
# chmod +x /usr/local/sbin/automount
# /etc/rc.d/devd restart
```

Et hop, ça fonctionne.....

## Gérer le log d'automount

A la base je n'avais pas pensé à cela, mais avec le temps j'observe un log d'automount de 350 Mo ! Il faut donc utiliser [newsyslog](#) afin de le gérer.

```
su -  
cd /etc/newsyslog.conf.d/  
nano automount.conf
```

avec le contenu suivant :

```
# newsyslog for automount  
# Cyrille  
/var/log/automount.log 640 7 1000 * JC
```

Dès lors une archive sera créée dès que le log sera supérieur à 1 000Kb et 7 archives seront conservées.

## Dropbox



J'ai laissé tomber **dropbox**. J'ai opté pour l'installation de **Seafile** sur mon VPS et le client GUI **seafile-gui** dispo dans les repos de FreeBSD, Linux sur les postes clients. Fait très bien le travail demandé ;)

Pas de portage Dropbox sous FreeBSD Le kernel de FreeBSD ne prend pas en charge **inotify** donc pas de dropbox... En tout cas dropbox ne fournit pas de client FreeBSD. Reste à trouver une alternative ou utiliser wine et le client windows (paraît il)... Il y a aussi [libnotify](#) mais pas trop cherché dans ce sens ....

En alternative, il y a [rclone](#) qui fait très bien le travail mais en ligne de commande. Mais reste très gérable.

```
pkg info rclone  
rclone-1.43  
Name           : rclone  
Version        : 1.43  
Installed on   : Sat Feb 23 19:02:31 2019 CET  
Origin         : net/rclone  
Architecture   : FreeBSD:12:amd64  
Prefix         : /usr/local  
Categories     : net  
Licenses       : MIT  
Maintainer     : wg@FreeBSD.org
```

```
WWW      : https://rclone.org/
Comment  : Sync files to and from various cloud services
Annotations :
  FreeBSD_version: 1200086
  repo_type      : binary
  repository     : FreeBSD
Flat size : 26.1MiB
Description
Sync files to and from Google Drive, S3, Swift, Cloudfiles, Dropbox and
Google Cloud Storage

WWW: https://rclone.org/
```

On l'installe

```
# pkg install rclone
```

Puis on le configure

```
$ rclone config
```

Toute la doc est là [Configurer rclone pour Dropbox](#)

L'option dropbox est la **7**, on valide le tout et on finit par être redirigé vers une page **d'association dropbox/rclone** qu'on renseigne et valide.

Sur mon système mon répertoire **dropbox** et dans mon **/home/\$USER/Dropbox** et contient 2 dossiers essentiels qui seront mis à jour soit de mon **PC vers la Dropbox** ou de la **Dropbox vers mon PC**.

J'ai écrit ce script qui fonctionne sur 3 arguments \* le premier : **IN** ou **OUT** : la synchronisation se fait vers la Dropbox ou depuis la Dropbox \* le second : **répertoire à mettre à jour** : COURS ou DIR \* le dernier : **dry** ou **write** : **dry** : on simule, **write** on met à jour

Puis 1 alias :

```
alias dropbox='/home/ragnarok/SCRIPTS/dropbox.sh'
```

Pour mettre à jour le répertoire DIR depuis la Dropbox vers mon PC

```
dropbox IN DIR write
```

Pour simuler une synchronisation du répertoire COURS de mon PC vers la Dropbox

```
dropbox OUT COURS dry
```

Et c'est tout ! Le script ci-dessous :

```
#!/bin/sh
echo == DEBUG ==
echo "Nom du script : $0";
```



```
echo argument 1 : $1
echo argument 2 : $2
echo "Tous les arguments : $* ($# arguments)"
echo == FIN DEBUG ==

# =====
case $# in
0) echo "Aucun parametre"
    echo "Syntaxe :  $0 IN/OUT DIR/COURS" ;;
1) echo "Il manque un paramètre !"
    echo "Syntaxe :  $0 IN/OUT DIR/COURS" ;;
2) echo "2 parametres passes au programme : $1 et $2" ;;
3) echo "Le 3eme argument ne peut être que --dry-run ou write" ;;
*) echo "TROP DE PARAMETRES !"
    echo "Syntaxe :  $0 IN/OUT DIR/COURS" ;;
esac

# =====
if [ $1 == "IN" ]
then
    echo "Depuis la dropbox";
    source="dropbox:/" ;
    dest="/home/ragnarok/Dropbox/";

elif [ $1 == "OUT" ]
then
    echo "Vers la dropbox" ;
    source="/home/ragnarok/Dropbox/" ;
    dest="dropbox:/";

else
    echo "$1 : Soit IN soit OUT";
    exit;
fi

# =====
if [ $2 == "DIR" ]
then
    echo "Repertoire de DIR";
    dir="DIR-2018-2019";
elif [ $2 == "COURS" ]
then
    echo "Repertoire de COURS"
    dir="COURS-2018-2019";
elif [ $2 == "ecole" ]
then
    echo "Repertoire Ecole"
    dir="ecole";
else
    echo "$2 : Soit DIR soit COURS soit ecole";
    exit;
```

```

    fi
# =====

if [ $3 == "dry" ]
then
    echo "Option --dry-run activée";
    option="--dry-run";
elif [ $3 == "write" ]
then
    echo "Option: aucune. En action";
    option="";
else
    echo "$3 : Ne peut être que dry ou write";
    exit;
fi
# =====

echo "rclone sync $source$dir $dest$dir $option --progress"
rclone sync $source$dir $dest$dir $option --progress

echo " ----- Done.";
```

## WINE

Attention il faut installer le paquet i386 même si on est en 64bit, sinon erreur de segmentation.

Donc

```
pkg install i386-wine
```

Puis peaufinage via

```
$ winecfg
```

Accepter tout ce qu'il propose de télécharger et installer.

Fonctionne nickel, mais les polices sont vraiment bizarres.

Éditer ~/.config/fontconfig/fonts.conf de cette manière :

```

<?xml version="1.0"?>
<!DOCTYPE fontconfig SYSTEM "fonts.dtd">

<fontconfig>

  <!-- antialias all fonts -->
  <match target="font">
    <edit name="antialias" mode="assign"><bool>true</bool></edit>
    <edit name="hinting" mode="assign"><bool>true</bool></edit>
```

```
<edit name="hintstyle" mode="assign"><const>hintslight</const></edit>
<edit name="rgba" mode="assign"><const>rgb</const> </edit>
</match>

</fontconfig>
```

## XFCE / Archives

Si vous avez le message d'erreur "Aucun gestionnaire d'archive trouvé" en essayant d'extraire ou de compresser un dossier via Thunar

```
pkg install xarchiver
```

## Arduino

Surtout ne pas installer la version depuis les ports, elle ne passe pas (l'IDE d'arduino ne sait pas compiler) et c'est hyper casse-pieds pour la désinstaller. Préférer la version précompilée (**pkg**).

```
pkg install arduino
```

Si nécessaire, inclure votre \$USER aux groupes **operator** et **dialer**

```
# sudo pw groupmod operator -m $USER
# sudo pw groupmod dialer -m $USER
```

Donner l'accès aux périphériques USB via **/etc/devfs.rules**

```
# cat /etc/devfs.rules
#
# Allow operators access to usb devices.
#
[operator_usb=5]
add path usbctl mode 0660 group operator
add path 'usb/*' mode 0660 group operator
add path 'ugen*' mode 0660 group operator
```

Prise en compte des changements

```
# sudo sysrc devfs_system_ruleset=operator_usb
```

Relance du service devfs

```
# sudo service devfs restart
```

On branche sa carte et on vérifie sa détection

```
$ usbconfig
```

```
...
```

```
ugen0.2: <Arduino (www.arduino.cc) product 0x0043> at usb0, cfg=0 md=HOST  
spd=FULL (12Mbps) pwr=0N (100mA)
```



## Bash

### bash comme shell par défaut



Attention ne pas changer le shell du root

```
# chsh -s /usr/local/bin/bash {username}
```

```
grep ragnarok /etc/passwd
```

### Cas d'un système UFS

```
cat /etc/fstab  
## append the following to /etc/fstab file ##  
fdesc /dev/fd fdescfs rw 0 0
```

### Cas d'un système ZFS (fdescfs doit être monté après la racine zfs)

```
cat /etc/fstab  
## append the following to /etc/fstab file ##  
fdesc /dev/fd fdescfs rw,auto,late 0 0
```

## Autocomplétion

add the following to your ~/.bashrc or ~/.bash\_profile file:

```
[[ $PS1 && -f /usr/local/share/bash-completion/bash_completion.sh ]] && \
. /usr/local/share/bash-completion/bash_completion.sh
```

```
# pkg install bash-completion
```

## Alias

Créer un fichier **.bash\_aliases** touch .bash\_aliases Et y mettre ses alias. Si non fait, insérer cette section dans le .bashrc

```
# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.
if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi
```

## VirtualBox

Jeter un oeil : [VirtualBox sous FreeBSD](#)

## DUAL BOOT FreeBSD / Linux

J'avais une Linux Debian SID d'installée sur mon PC.

```
$ sudo fdisk -l
Disque /dev/sda : 111,8 GiB, 120034123776 octets, 234441648 secteurs
Unités : secteur de 1 × 512 = 512 octets
Taille de secteur (logique / physique) : 512 octets / 512 octets
taille d'E/S (minimale / optimale) : 512 octets / 512 octets
Type d'étiquette de disque : dos
Identifiant de disque : 0x0c3bb11c

Périphérique Amorçage Début Fin Secteurs Taille Id Type
/dev/sda1 * 2048 143362047 143360000 68,4G 83 Linux
/dev/sda2 226338814 234440703 8101890 3,9G 5 Étendue
/dev/sda3 143362048 226338813 82976766 39,6G a5 FreeBSD
/dev/sda5 226338816 234440703 8101888 3,9G 82 partition
d'échange
```

Donc on voit

- sda1 : Partition primaire : Racine linux /
- sda2 : Partition étendue
- sda 3 : FreeBSD
- sda5 : swap Linux

**Configuration du grub sous LINUX** Sous LINUX, éditer le fichier suivant : **/etc/grub.d/40\_custom** et créer l'entrée ci-dessous

Attention hd0,3 car freeBSD est sur /dev/sda3, s'il était sur le 1, mettre hd0,1, sur le 2 hd0,2, sur la seconde partition d'un deuxième disque hd1,2... Donc à adapter à votre configuration

```
#!/bin/sh
exec tail -n +3 $0
# This file provides an easy way to add custom menu entries.  Simply type
# the
# menu entries you want to add after this comment.  Be careful not to change
# the 'exec tail' line above.
menuentry "FreeBSD 12" {
    set root='(hd0,3)'
    kfreebsd /boot/loader
}
```

Prendre en compte les modifications de la configuration du grub

```
# update-grub
Création du fichier de configuration GRUB...
Image Linux trouvée : /boot/vmlinuz-4.10.0-38-generic
Image mémoire initiale trouvée : /boot/initrd.img-4.10.0-38-generic
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
unknown Linux distribution trouvé sur /dev/sda3
```

Rebooter et la nouvelle entrée est fonctionnelle dans le grub.

## SSD Config [UFS]

Rien d'extraordinaire à faire, juste **activer le trim** sur les partitions montées Chez moi ada0p2 (/) et ada0p4 (/home)

Rebooter et démarrer en mode : **Boot Single User** (le n°2). (Il ne faut pas que les partitions soient montées)

```
tunefs -t enable ada0p2
tunefs -t enable ada0p4
```

Rebooter

Vérifier que ce soit pris en compte

```
tunefs -p /dev/ada0p4 | grep trim
tunefs: POSIX.1e ACLs: (-a) disabled
tunefs: NFSv4 ACLs: (-N) disabled
tunefs: MAC multilabel: (-l) disabled
tunefs: soft updates: (-n) enabled
tunefs: soft update journaling: (-j) enabled
tunefs: gjournal: (-J) disabled
tunefs: trim: (-t) enabled
tunefs: maximum blocks per file in a cylinder group: (-e) 4096
tunefs: average file size: (-f) 16384
tunefs: average number of files in a directory: (-s) 64
tunefs: minimum percentage of free space: (-m) 8%
tunefs: space to hold for metadata blocks: (-k) 6408
tunefs: optimization preference: (-o) time
tunefs: volume label: (-L)
```

## Problème de boot UFS

Pour schéma de disque **UFS** Au boot du système, j'avais le message "**Invalid partition table !**" Une pression sur ENTER lançait cependant le système. Voici la marche suivit par régler le problème (déclarer *proprement* la partition active)

```
gpart show
```

```
=>      40  468862048  ada0  GPT  (224G)
        40      1024      1  freebsd-boot  (512K)
       1064  25164800      2  freebsd-ufs   (12G)
      25165864 16777216      3  freebsd-swap (8.0G)
      41943080 398458880      4  freebsd-ufs  (190G)
      440401960 28460128      5  freebsd-ufs  (14G)
```

Dans un premier temps, régénérer le secteur de boot

```
cd /boot
gpart bootcode -b pmbr -p gptboot -i1 ada0
```

Si ne fonctionne toujours pas après un redémarrage, re-préciser la partition **active**.

```
[ragnarok@asgard-freebsd:~] $ sudo fdisk -a
***** Working on device /dev/ada0 *****
parameters extracted from in-core disklabel are:
cylinders=465141 heads=16 sectors/track=63 (1008 blks/cyl)
Figures below won't work with BIOS for partitions not in cyl 1
parameters to be used for BIOS calculations are:
cylinders=465141 heads=16 sectors/track=63 (1008 blks/cyl)
Media sector size is 512
Warning: BIOS sector numbering starts with sector 1
Information from DOS bootblock is:
```

```

The data for partition 1 is:
sysid 238 (0xee), (EFI GPT)
  start 1, size 468862127 (228936 Meg), flag 0
  beg: cyl 0/ head 0/ sector 2;
  end: cyl 1023/ head 255/ sector 63

The data for partition 2 is:
<UNUSED>
The data for partition 3 is:
<UNUSED>
The data for partition 4 is:
<UNUSED>
Do you want to change the active partition? [n] y
Supply a decimal value for "active partition"

```

On précise le numéro de la partition où booter et valide le reste.

On reboote et ça devrait passer

## ZFS Configuration

Si vous avez choisi cette option, vous devriez avoir cette sortie.

```

[ragnarok@asgard-freeBSD:~] $ zfs list
NAME                                USED  AVAIL  REFER  MOUNTPOINT
zroot                               9,06G  202G   88K    /zroot
zroot/R00T                          3,81G  202G   88K    none
zroot/R00T/default                  3,81G  202G   3,81G  /
zroot/tmp                           208K   202G  208K    /tmp
zroot/usr                           5,24G  202G   88K    /usr
zroot/usr/home                      4,49G  202G   4,49G  /usr/home
zroot/usr/ports                     763M   202G  763M    /usr/ports
zroot/usr/src                       88K   202G   88K    /usr/src
zroot/var                           680K   202G   88K    /var
zroot/var/audit                     88K   202G   88K    /var/audit
zroot/var/crash                     88K   202G   88K    /var/crash
zroot/var/log                       216K   202G  216K    /var/log
zroot/var/mail                      112K   202G  112K    /var/mail
zroot/var/tmp                       88K   202G   88K    /var/tmp

```

```

[ragnarok@asgard-freeBSD:~] $ zfs get -r canmount zroot
NAME                                PROPERTY  VALUE      SOURCE
zroot                               canmount  on         default
zroot/R00T                          canmount  on         default
zroot/R00T/default                  canmount  noauto     local
zroot/tmp                           canmount  on         default
zroot/usr                           canmount  off        local
zroot/usr/home                      canmount  on         default
zroot/usr/ports                     canmount  on         default

```



zroot/usr/src	canmount	on	default
zroot/var	canmount	off	local
zroot/var/audit	canmount	on	default
zroot/var/crash	canmount	on	default
zroot/var/log	canmount	on	default
zroot/var/mail	canmount	on	default
zroot/var/tmp	canmount	on	default

Si problème de montage au boot, vérifier que **/etc/rc.conf** contienne bien cette ligne

```
zfs_enable="YES"
```

Normalement le TRIM est pris en compte par défaut

```
root@asgard-freeBSD:~ # sysctl vfs.zfs.trim
vfs.zfs.trim.max_interval: 1
vfs.zfs.trim.timeout: 30
vfs.zfs.trim.txg_delay: 32
vfs.zfs.trim.enabled: 1
```

```
root@asgard-freeBSD:~ # sysctl -d kstat.zfs.misc.zio_trim
kstat.zfs.misc.zio_trim:
kstat.zfs.misc.zio_trim.failed: Number of TRIM requests that failed for
reasons other than not supported
kstat.zfs.misc.zio_trim.unsupported: Number of TRIM requests that failed
because TRIM is not supported
kstat.zfs.misc.zio_trim.success: Number of successful TRIM requests
kstat.zfs.misc.zio_trim.bytes: Number of bytes successfully TRIMmed
```

```
root@asgard-freeBSD:~ # sysctl -a |grep _trim
kern.cam.nda.max_trim: 256
kstat.zfs.misc.zio_trim.failed: 0
kstat.zfs.misc.zio_trim.unsupported: 0
kstat.zfs.misc.zio_trim.success: 34651
kstat.zfs.misc.zio_trim.bytes: 2751168512
```

Optimisation en lecture / écriture (mais tout dépend de l'utilisation que vous en faites)

```
zpool add 'nom_du_zpool' cache 'nom_du_device_ssd'
zpool add 'nom_du_zpool' log 'nom_du_device_ssd'
```

## Gestion des fichiers core

### core générés par le kernel

Dans **/etc/sysctl.conf** Ajouter cette option

```
kern.coredump=0
```

Ou celle-ci : Reroutage des fichiers core

```
kern.corefile=/dev/null
```

OU dumpdev="NO" to your **/etc/rc.conf**

## core générés par les applications

CSH limit coredumpsize 0 dans **/etc/csh.login**

BASH

ulimit -c 0 dans **.bash\_profile**

## Gestion des paquets PKG

### Gestion des mises à jour (PKG)

Récupérer la liste des mises à jour

```
# pkg update
```

La déployer

```
# pkg upgrade
```

### Geler / dégeler un paquet

Lister les paquets gelés

```
# pkg -l
```

Geler un paquet ===

```
# pkg lock -y LePaquet
```

Dégeler un paquet

```
# pkg unlock LePaquet
```

### Installer la dernière version des paquets de pkg

Créer ce dossier

```
mkdir -p /usr/local/etc/pkg/repos
```

et ce fichier

```
cat /usr/local/etc/pkg/repos/FreeBSD.conf
FreeBSD: {
  url: "pkg+http://pkg.FreeBSD.org/${ABI}/latest"
}
```

Recharger la base de données des paquets

```
pkg upgrade -f
```

Mettre à jour

## Vérifier quel dépôt est utilisé

```
pkg -vv
```

## Informations sur pkg

```
pkg info pkg
```

# Gestion des ports

Pour ceux qui préfèrent compiler les paquets par eux.

## via portsnap

Première install

```
# portsnap fetch
# portsnap extract
```

Pour les fois suivantes

```
# portsnap fetch update
```

Installer un programme

```
# cd /usr/ports/shells/bash
```

Installer

```
# make install clean
```

Le supprimer

```
# make deinstall
```

## portmaster et mise à jour des ports

Pour mettre à jour les logiciels des ports, le plus simple est d'utiliser portmaster

```
# cd /usr/ports/ports-mgmt/portmaster  
# make install clean
```

Lister les paquets à mettre à jour

```
# portmaster -L
```

Les mettre à jour

```
# portmaster -a
```

Portmaster peut aussi être utilisé pour gérer l'installation d'un programme. Exemple

```
# portmaster shells/bash
```

## Patches de sécurité

freebsd-version

**-k** Print the version and **patch** level of the installed kernel. Unlike **uname(1)**, if a new kernel has been installed but the system has not yet rebooted, freebsd-version will print the version and **patch** level of the new kernel.

**-r** Print the version and **patch** level of the running kernel. Unlike **uname(1)**, this is unaffected by environment variables.

**-u** Print the version and **patch** level of the installed userland. These are hardcoded into freebsd-version during the build.

```
root@asgard-freeBSD:~ # freebsd-update fetch  
src component not installed, skipped  
Looking up update.FreeBSD.org mirrors... 3 mirrors found.  
Fetching public key from update4.freebsd.org... done.  
Fetching metadata signature for 12.0-RELEASE from update4.freebsd.org...  
done.
```

```
Fetching metadata index... done.  
Fetching 2 metadata files... done.  
Inspecting system... done.  
Preparing to download files... done.
```

```
root@asgard-freeBSD:~ # freebsd-update install  
src component not installed, skipped  
Installing updates... done.
```

```
root@asgard-freeBSD:~ # uname -a  
FreeBSD asgard-freeBSD 12.0-RELEASE FreeBSD 12.0-RELEASE r341666 GENERIC  
amd64
```

```
root@asgard-freeBSD:~ # freebsd-version -u  
12.0-RELEASE-p5
```

```
root@asgard-freeBSD:~ # freebsd-version -k  
12.0-RELEASE-p4
```

## Upgrader FreeBSD 12.0 à 12.1

```
# freebsd-update fetch  
# freebsd-update install  
# freebsd-update upgrade -r 12.1-RELEASE  
# freebsd-update install
```

A cette étape, rebooter le système :

```
# shutdown -r now
```

Après ce reboot, freebsd-update(8) doit être relancer avec les nouveaux composants de l'userland:

```
# freebsd-update install
```

A cette étape les utilisateurs sont invités par freebsd-update(8) à reconstruire leurs applications tiers (comme celles compilées depuis les sources (/usr/port/) afin de faire correspondre les librairies système.

Une fois fait, relancer freebsd-update(8) afin de supprimer les libraires inutiles

```
# freebsd-update install
```

Enfin reboter sous **FreeBSD 12.1-RELEASE**

```
# shutdown -r now
```

Une fois rebooter, il est recommandé de régénérer le paquet pkg afin d'éviter de futurs problèmes lors des installations à venir.

```
# pkg-static install -f pkg
```

## Sources

Je n'ai pas pondu tout cela tout seul, j'ai lu pas mal de docs et les forums BSD. Voici les sources indispensables à mon avis :

- [O. Cochard FreeBSD](#)
- [Mine d'or d'informations](#)
- [GIT de Vermaden](#)
- [FreeBSD on a laptop](#)
- [Post installation freeBSD laptop](#)
- [FreeBSD Install BASH Shell Using pkg command](#)
- [A FreeBSD 11 Desktop How-to](#)

Et surtout

- [Site FreeBSD](#)
- [Forums FreeBSD](#)
- [HandBook Français de FreeBSD](#)
- [Forum FUG-FR](#)
- [ZFS](#)

From:

<https://cbiot.fr/dokuwiki/> - **Cyrille BIOT**

Permanent link:

<https://cbiot.fr/dokuwiki/installfreebsd>

Last update: **2020/06/06 17:12**

