

XEN: Mise en oeuvre sur ARCHLINUX

Table of Contents

- [XEN: Mise en oeuvre sur ARCHLINUX](#)
- [Archlinux](#)
 - [Booter sur l'iso d'ArchLinux](#)
 - [Pré-requis](#)
 - [Réseau](#)
 - [Matériel](#)
 - [Partitionnement](#)
 - [Montage des partitions](#)
 - [Pacman](#)
 - [Installation du système de base](#)
 - [Configuration](#)
 - [fstab](#)
 - [chroot](#)
 - [locales](#)
 - [hostname](#)
 - [vconsole.conf](#)
 - [locale.conf](#)
 - [time-zone](#)
 - [pacman.conf](#)
 - [Réseau](#)
 - [Utilisateurs](#)
 - [bootloader : grub](#)
 - [Démontage et reboot](#)
 - [Produits complémentaires](#)
- [Environnement graphique : X](#)
 - [Installation du serveur X](#)
 - [Drivers graphiques ATI standards](#)
 - [Drivers propriétaires \(catalyst\) pour les cartes ATI](#)
 - [Configuration du serveur](#)
 - [Configuration double Head avec des écrans indépendants](#)
 - [Installation d'OpenBox](#)
 - [Configuration](#)
 - [Paramétrage du lancement](#)
 - [Personnalisation du bureau](#)
 - [Installation de lightdm](#)
 - [Installation de displaylink](#)
 - [Installer udlfb](#)
 - [Installer xf86-video-fbdev-for-displaylink](#)
 - [Processus de boot en mode graphique](#)

- [Personnaliser GRUB](#)
- [fb splash](#)
- [Configuration de multitouch](#)
- [Installation de modules complémentaires](#)
 - [Network Manager](#)
 - [Chromium](#)
 - [linux-headers](#)
 - [Yaourt](#)
- [XEN](#)
 - [Pré requis](#)
 - [Evaluation](#)
 - [Compatibilité](#)
 - [Installation](#)
 - [Obtention de Xen](#)
 - [Configuration de Xen](#)
 - [Configuration Bootloader](#)
 - [Polkit](#)
 - [Service systemd](#)
 - [Interface d'administration graphique](#)
 - [Installation](#)
 - [PolicyKit authorisation](#)
 - [Étapes finales](#)

Xen est un hyperviseur à source ouverte de type 1 ou baremetal, qui permet d'exécuter de nombreuses instances d'un système d'exploitation, voire de systèmes d'exploitation différents, en parallèle sur une seule machine (ou hôte). Xen est le seul hyperviseur de type 1 disponible en source ouverte.

L'hyperviseur Xen est une couche mince de logiciels qui émule une architecture d'ordinateur permettant à plusieurs systèmes d'exploitation de fonctionner simultanément. L'hyperviseur est démarré par le chargeur de démarrage de l'ordinateur sur lequel il est installé. Une fois que l'hyperviseur est chargé, il démarre le dom0 (abréviation de "domaine 0", parfois appelé hôte ou domaine privilégié) qui, dans notre cas, exécute Arch Linux. Une fois que le dom0 a démarré, un ou plusieurs domU (en abrégé, pour les domaines utilisateur, parfois appelés VM ou invités) peuvent être démarrés et contrôlés à partir du dom0. Xen prend en charge les domU à la fois paravirtualisés (PV) et virtualisés sur le matériel (HVM). Voir Xen.org pour un aperçu complet.

Les packages de noyau linux et linux-lts Arch intègrent une prise en charge au niveau du noyau de l'hyperviseur Xen.

Archlinux

Booter sur l'iso d'ArchLinux

Choisir le système : 64 ou 32 bits.

Sur le shell root ... en anglais (c'est du qwerty) il faut donc de suite, pour plus de facilité, changer le clavier en FR en tapant la commande suivante :

```
$ loadkeys fr-pc
```



le « q » est sur le « a » et le tiret (du 6) est le signe - (moins) du pavé numérique.

Pré-requis

Réseau

Vérifier de suite le réseau par :

```
$ ping -c 3 www.google.com
```

Théoriquement il ne devrait pas y avoir de problème car le démon dhcp est lancé au démarrage. Je vous recommande vraiment de faire l'installation en étant connecté par câble au réseau. Ça sera beaucoup plus simple, dans un premier temps que par wifi par exemple. Si au boot, vous n'avez pas de réseau, il va falloir le configurer manuellement.

Matériel

Pour exécuter HVM domU , le matériel physique doit prendre en charge la virtualisation Intel VT-x ou AMD-V (SVM). Pour le vérifier, exécutez la commande suivante lorsque l'hyperviseur Xen n'est pas en cours d'exécution:

```
$ grep -E "(vmx | svm)" --color = toujours / proc / cpuinfo
```

Si la commande ci-dessus ne génère pas de sortie, la prise en charge de la virtualisation matérielle n'est pas disponible et votre matériel ne peut pas exécuter HVM domU (ou on utilise déjà l'hyperviseur Xen). Pours'assurer que le processeur prend en charge l'une de ces fonctionnalités, il faut accéder au menu de configuration du BIOS du système hôte pendant le processus de démarrage et vérifier si les options liées à la prise en charge de la virtualisation ont été désactivées. Si une telle option existe et est désactivée, l'activer, démarrer le système et répéter la commande ci-dessus. L'hyperviseur Xen prend également en charge le transfert PCI, où les périphériques PCI peuvent être passés directement à domU, même en l'absence de support dom0 pour le périphérique. Pour pouvoir

utiliser la liaison PCI, la CPU doit prendre en charge IOMMU/VT-d.

Partitionnement

Pour le partitionnement on dispose de différents utilitaires disponibles : `fdisk`, `parted`, `cfdisk`, etc.

Une fois le partitionnement effectué, il va falloir formater ces partitions :

```
/dev/sda1 = /  
/dev/sda2 = swap
```

Formater les partitions :

```
$ mkfs.btrfs /dev/sda1  
$ mkfs.btrfs /dev/sda5
```

Le système de fichier btrfs permet de prendre des snapshots (photos) mais pour cela il faut installer les système sur des subvolumes btrfs.

Pour créer les subvolumes btrfs :

Monter le disque nouvellement créer ...

```
$ mount /dev/sda1 /mnt
```

... créer les subvolumes ...

```
$ btrfs subvolume create /mnt/arch_root  
$ btrfs subvolume create /mnt/snap_root  
$ btrfs subvolume create /mnt/arch_home  
$ btrfs subvolume create /mnt/snap_home
```

... démonter le volume physique

```
$ umount /mnt
```

Créer également la swap :

```
$ mkswap /dev/sda2
```

Montage des partitions

Puis, il faut monter les volumes logiques :

```
$ mount -t btrfs -o subvol=arch_root /dev/sda1 /mnt  
$ mount -t btrfs -o subvol=arch_home /dev/sda1 /mnt/home
```

... et on oublie surtout pas d'activer la swap :

```
$ swapon /dev/sda5
```

Pacman

Modifier le fichier de configuration des miroirs de pacman pour utiliser un miroir français, ce qui facilitera la réception des paquets en raison d'une bande passante certainement bien meilleure qu'un miroir situé à l'autre bout de la planète :

```
$ nano /etc/pacman.d/mirrorlist
```

Aller à la ligne du premier miroir FR et de faire Alt+6 afin de copier la ligne. Puis rendez-vous en début de fichier et copier la ligne avec Ctrl+u (voir image). Enregistrez.

Installation du système de base

Autre étape très importante, installer le système de base par la commande suivante :

```
$ pacstrap /mnt base base-devel
```

Là, il y a pas mal de chose à télécharger donc selon la bande passante, le PC, etc. le temps peut varier.

Configuration

Il faut créer ou éditer quelques fichiers de configuration.

fstab

Créer directement /etc/fstab par la commande suivante :

```
$ genfstab -U -p /mnt >> /mnt/etc/fstab
```

chroot

Il faut maintenant chrooter le nouveau système avec la commande suivante :

```
arch-chroot /mnt
```

locales

Editer /etc/locale.gen et dé-commenter la ligne fr_FR.UTF-8 qui correspond au français pour configurer l'utilisation en français puis lancer la commande suivante :

```
$ locale-gen
```

... ce qui va générer les locales.

hostname

Renseigner le nom de la machine dans le fichier /etc/hostname.

```
echo nomhote > /etc/hostname
```

vconsole.conf

Éditer/créer le fichier /etc/vconsole.conf afin d'y spécifier la disposition du clavier à utiliser et ajouter la ligne suivante :

```
KEYMAP=fr-pc
```

locale.conf

Créer un fichier /etc/locale.conf avec la ligne suivante :

```
LANG=fr_FR.UTF-8
```

time-zone

Configurer le time-zone en entrant la commande suivante :

```
$ ln -s /usr/share/zoneinfo/Europe/Paris /etc/localtime
```

Puis configurer l'horloge :

```
$ hwclock --systohc --utc
```

pacman.conf

Pour utiliser un système 64 bits, on doit activer le dépôt [multilib] qui vous donne une compatibilité 32-64 bits :

```
$ nano /etc/pacman.conf
```

Puis dé-commenter les 3 lignes :

```
[multilib]
SigLevel = PackageRequired
Include = /etc/pacman.d/mirrorlist
```

Puis mettre à jour les dépôts avec la commande suivante :

```
$ pacman -Sy
```

Réseau

On a vu tout à l'heure que vous devriez déjà disposer d'une connexion au Net par l'intermédiaire du dhcp. Vous devez maintenant activer le service Réseau sur votre nouvelle configuration. Pour le configurer en arrière-plan (service), vous devez utiliser systemctl en faisant :

```
$ systemctl enable dhcpd.service
```

Si vous êtes sur un portable et que prévoyez d'utiliser le wifi, vous devriez installer les paquets suivants (je n'ai pas testé car je ne suis pas connecté en wifi) :

```
$ pacman -S net-tools wireless_tools wpa_supplicant wpa_actiond dialog
```

... En même temps, on découvre « pacman » qui avec l'option -S permet d'installer les paquets (-Ss pour rechercher des paquets).

Pour le wifi, taper la commande suivante :

```
$ wifi-menu
```

Cela aura pour effet de créer votre profile dans /etc/network.d, puis vous devez activer le service wireless avec la commande suivante :

```
$ systemctl enable net-auto-wireless.service
```

Utilisateurs

Créer le mot de passe root avec la commande :

```
$ passwd
```

Puis créer un nouvel utilisateur puis son mot de passe avec les commandes suivantes :

```
$ useradd -m -g users -G wheel,storage,power -s /bin/bash machinbidule $  
passwd machinbidule
```

Installer sudo (bien pratique). Pour se faire, utiliser la commande suivante :

```
$ pacman -S sudo
```

Une fois sudo installé, on va autoriser le nouvel utilisateur à “utiliser” cette commande en éditant le fichier de configuration avec la commande suivante :

```
$ visudo
```

Si vous n’êtes pas familiarisé avec l’éditeur vi, vous pouvez aussi utiliser visudo avec nano en faisant comme suit :

```
$ EDITOR=nano visudo
```

Une fois le fichier ouvert, aller vers le bas du fichier et dé-commenter la ligne suivante :

```
%wheel ALL=(ALL) ALL
```

Enregistrer. Ceci va permettre à tous les membres du groupe wheel (auquel l'utilisateur appartient puisque on a précisé cette option lors de la création de l'utilisateur) d'utiliser sudo.

bootloader : grub

Nous allons maintenant installer le bootloader qui va permettre de démarrer ArchLinux mais aussi d'éventuels autres système présents sur votre PC (comme un Windows par exemple...).

Pour installer grub, taper les commandes suivantes (en remplaçant éventuellement /dev/sda par le véritable emplacement de votre disque) :


```
$ pacman -S grub-bios
$ grub-install --target=i386-pc --recheck /dev/sda
$ cp /usr/share/locale/fr/LC_MESSAGES/grub.mo /boot/grub/locale/fr.mo
```

Si on est en dual-booting (avec un Windows), installer os-prober :

```
$ pacman -S os-prober
```

Puis taper la commande suivante :

```
$ grub-mkconfig -o /boot/grub/grub.cfg
```

Cela va créer le fichier de configuration pour votre boot-loader.

Démontage et reboot

Enfin, on quitte le chroot, on démonte les partitions et on reboot :

```
$ exit
$ umount /mnt/home
$ umount /mnt
$ reboot
```

Produits complémentaires

```
$ sudo pacman -S wget git dos2unix
```

L'installation du système est terminée.

Environnement graphique : X

Installation du serveur X

Avant de pouvoir installer un environnement graphique il faut d'abord installer le serveur X:

```
$ sudo pacman -S xorg-server xorg-server-common xorg-server-utils xorg-
server-devel
xorg-xinit xorg-utils xf86-video-vesa((préferez un driver video adapté))
xf86-video-fbdev((xf86-video_fbdev ne doit pas être installé si on doit
```

```
utiliser le driver spécifique à Displaylink))
```

Drivers graphiques ATI standards

Il ne faut surtout pas oublier d'installer aussi le driver pour la carte graphique. Si vous ne savez pas quelle carte graphique possède votre PC, vous pouvez le déterminer en tapant:

```
$ lspci | grep -i vga
```

Installer le driver graphique adéquat

```
$ sudo pacman -S xf86-video-ati
```

En cas de Ralentissement des performances avec les pilotes open-source on peut :

Forcer les EDID

Dans le cas où votre moniteur/téléviseur n'envoie pas des données EDID appropriées ou de problèmes similaires, la résolution native n'est pas automatiquement configurée ou pas d'affichage du tout. Vous pouvez vérifier si il y a un problème de chargement d'EDID^[1] lors de la connexion du DisplayPort en exécutant:

```
$ dmesg | egrep "drm|radeon"
```

Si cette commande montre quelque chose comme ça :

```
[ 137.375556] radeon 0000:02:00.0: HDMI Type A-1: EDID block 0 invalid.  
[ 137.375560] [drm:radeon_dvi_detect] *ERROR* HDMI Type A-1: probed a  
monitor but no invalid EDID
```

Le noyau linux contient une disposition pour charger les données binaires EDID, et fournit des données pour quatre des résolutions les plus typiques.

Si vous avez le fichier EDID de votre moniteur le processus est facile. Si vous n'en disposez pas, vous pouvez soit utiliser l'un des EDID binaires haute-résolution (ou en générer un lors de la compilation du noyau) ou construire votre propre EDID ainsi :

Télécharger le kernel linux correspondant à celui installé par la Archlinux par exemple le kernel 3.8 :

```
$ wget  
http://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/snapshot/linu  
x-3.8.tar.gz  
tar xzf linux-3.8.tar.gz  
$ cd linux-3.8/Documentation/EDID
```

Construire les binaires par un simple make

```
$ make
```



Le paquet dos2unix doit être installé pour pouvoir compiler les binaires

```
$ sudo -S dos2unix
```

Pour utiliser les binaires haute résolution , créer un répertoire edid dans /lib/firmware:

```
$ sudo mkdir /lib/firmware/edid
```

...puis copier le fichier binaire dans le répertoire /lib/firmware/edid.

```
$ sudo cp *.bin /lib/firmware/edid/
```

Pour le charger au démarrage, créer le fichier drmkmshelper.conf dans le répertoire /etc/modprobe.d :

```
$ sudo nano /etc/modprobe.d/drm_kms_helper.conf
```

... et y ajouter ce qui suit

```
options drm_kms_helper edid_firmware=edid/your_edid.bin
```

.. ou spécifier uniquement pour une connexion:

```
options drm_kms_helper edid_firmware=VGA-1:edid/your_edid.bin
```

Pour les quatre niveaux de résolution, voir dans le tableau ci-dessous le nom à spécifier:

Resolution	Name to specify
1024×768	edid/1024×768.bin
1280×1024	edid/1280×1024.bin
1680×1050	edid/1680×1050.bin
1920×1080	edid/1920×1080.bin

Si on charge KMS plus en amont du boot, il faut inclure le fichier personnalisé EDID dans l'initramfs ou on aura des problèmes.



Un mode peut être forcé sur la ligne de commande du noyau. Malheureusement, l'option

vidéo en ligne de commande est mal documentée dans le cas DRM. Une aide sur la façon de l'utiliser peut être trouvée dans :



- * <http://cgит.freedesktop.org/nouveau/linux-2.6/tree/Documentation/fb/modedb.txt>
- * http://cgит.freedesktop.org/nouveau/linux-2.6/tree/drivers/gpu/drm/drm_fb_helper.c

Le format est le suivant:

```
video=<conn>:<xres>x<yres>[M][R][-<bpp>][@<refresh>][i][m][eDd]
```

<conn>: Connecteur, par exemple DVI-I-1, consultez votre journal du noyau.

<xres>x<yres>: résolution

M: calculer un mode CVT?

R: blanking réduit?

-<bpp>: Profondeur de couleur

@<refresh>: Le taux de rafraîchissement

i: entrelacé (non CVT Mode)

m: les marges?

e: sortie sur forcés de

d: sortie forcée d'arrêt

D: sortie numérique forcée pour le (par exemple connecteur DVI-I)

On peut remplacer les modes de sorties à l'aide de plusieurs «video» à plusieurs reprises, par exemple, pour forcer DVI à 1024×768 à 85 Hz et TV-out off:

```
video=DVI-I-1:1024x768@85 video=TV-1:d
```

Désactiver KMS

Vous pouvez désactiver KMS pour diverses raisons, telles qu'un écran vide ou une erreur "no signal" sur l'écran, lorsque vous utilisez le driver Catalyst, etc Pour désactiver KMS, ajouter nomodeset comme un paramètre du noyau. Voir Paramètres du noyau pour plus d'informations.



Certains pilotes Xorg ne fonctionnent pas lorsque KMS est désactivé. Voir la page wiki sur votre pilote spécifique pour plus de détails.

Certaines cartes installées par défaut peuvent essayer d'utiliser KMS. Vous pouvez vérifier si tel est votre cas en exécutant:

```
dmesg | egrep "drm|radeon"
```

Cette commande peut montrer quelque chose comme ça, c'est à dire qu'il essaie d'employer KMS par défaut:

```
[drm] radeon default to kernel modesetting.  
...  
[drm:radeon_driver_load_kms] *ERROR* Failed to initialize radeon, disabling  
IOCTL
```

Si votre carte ne supporte pas KMS (avant les cartes e100), alors vous pouvez désactiver KMS. Cela devrait résoudre le problème.

Il convient de désactiver le kernel mode-setting au cas de "mode panic" du noyau provoqué par framebuffer, de déformation au démarrage, d'absence de signal GPU, de refus de Xorg de démarrer, de relance le logiciel Mesa de tramage suite à un «POWER OFF» de Xorg (pas d'accélération 3D) ou de problème d'arrêt (noyau 2.6.33 -2).

- Ajouter radeon.modeset=0 (ou nomodeset, si cela ne fonctionne pas) à la ligne kernel options dans le fichier de configuration du bootloader. Cela devrait fonctionner.



L'ajout de nomodeset à la ligne de démarrage du noyau pourrait empêcher gnome-shell de GNOME 3 ou les effets de bureau KDE de fonctionner.

Si vous voulez retirer le support KMS de l'initramfs, suivez les deux étapes suivantes.

- Si radeon a été ajouté au tableau MODULES dans mkinitcpio.conf pour permettre le démarrage rapide, retirez-le.
- Reconstruire les-initramfs avec

```
mkinitcpio-p linux
```

Alternativement, les options de module peuvent être spécifiées dans un fichier dans le répertoire /etc/modprobe.d. Si vous utilisez le module radeon (lsmod | grep radeon) désactiver KMS en créant un fichier /etc/modprobe.d/radeon.conf contenant le code ci-dessus:

```
options radeon modeset=0
```

Drivers propriétaires (catalyst) pour les cartes ATI

Le driver catalyst est disponible dans le dépôt [community], mais il n'est pas compatible avec la dernière version du serveur Xorg.

Avant d'installer xorg-server il convient :

- de désinstaller xorg-server installé depuis le dépôts extra.
- d'ajouter les serveurs catalyst adaptés dans le fichier /etc/pacman.conf

Désinstaller xorg-server

```
sudo pacman -R xorg-server xf86-video-evdev
```

Ajouter les dépôts catalyst :

```
[xorg###]  
Server = http://catalyst.wirephire.com/repo/xorg###/x86_64
```

...où ### est le numéro de la version du serveur à installer. Par exemple :

```
[xorg113]  
Server = http://catalyst.wirephire.com/repo/xorg113/x86_64
```

... commenter le dépôt [extra] dans /etc/pacman.conf ...

```
#[extra]  
#SigLevel = PackageOptional  
#Include = /etc/pacman.d/mirrorlist
```

... ajouter la clé de signature du serveur ...

```
$ sudo pacman-key --keyserver pgp.mit.edu --recv-keys [clé du serveur]((la  
clé du serveur peut être obtenue en tentant une première synchronisation de  
pacman qui échouera forcément))  
$ sudo pacman-key --lsign-key [clé du serveur]((la clé du serveur peut être  
obtenue en tentant une première synchronisation de pacman qui échouera  
forcément))
```

... synchroniser les dépôts ...

```
$ sudo pacman -Syy
```

... réinstaller xorg-server ...

```
$ sudo pacman -S xorg-server xorg-server-common xorg-server-utils xorg-  
server-devel xf86-video-evdev netkit-bsd-finger(**netkit-bsd-finger:**  
affiche des informations sur les utilisateurs du système (nom de  
l'utilisateur, son vrai nom, nom du terminal))
```

Installer les drivers catalyst

- Depuis [community] :

```
$ sudo pacman -S catalyst-dkms catalyst-utils
```

- Ou depuis le serveur catalyst :

pour cela ajouter le serveur [catalyst] dans /etc/pacman.conf

```
[catalyst]
Server = http://catalyst.wirephire.com/repo/catalyst/x86_64
```

... synchroniser les dépôts ...

```
$ sudo pacman -Syy
```

... puis installer les drivers catalyst.

```
$ sudo pacman -S catalyst catalyst-utils
```



Ne pas oublier de réactiver le dépôt **[extra]** dans le fichier /etc/pacman.conf puis de refaire une synchronisation de pacman pas `sudo pacman -Syy`

Charger le pilote fglrx

```
modprobe fglrx
```



Rajouter à la ligne MODULES de /etc/rc.conf pour une prise en charge automatique au démarrage.

Configuration du serveur

Ajouter au fichier de configuration /etc/X11/xorg.conf.d/10-evdev.conf le type de clavier utilisé et accessoirement rétablir l'usage des touches ctrl+alt+backspace pour tuer le serveur X:

```
nano /etc/X11/xorg.conf.d/10-evdev.conf
```

Ensuite modifiez la section InputClass correspondant à evdev keyboard catchall de façon à ce qu'elle ressemble à ceci:

```
Section "InputClass"
    Identifier "evdev keyboard catchall"
```

```
MatchIsKeyboard "on"
MatchDevicePath "/dev/input/event*"
Driver "evdev"
Option "XkbLayout" "fr"
Option "XkbOption" "terminate:ctrl_alt_bksp"
EndSection
```

Si on utilise le driver catalyst, ce pilote n'est pas détecté automatiquement, par conséquent il faut rajouter au minimum le fichier de configuration suivant dans `/etc/X11/xorg.conf.d/`, par exemple nommé `20-catalyst.conf`:



```
Section "Device"
    Identifier "Card0"
    Driver      "fglrx"
    VendorName  "ATI Technologies Inc"
    BusID       "PCI:1:0:0"
EndSection
```

Le BusID correspond à la série de nombres sur la ligne de la carte graphique obtenu par `lspci` (attention, pour X, il faut remplacer le point par deux points).

Si on utilise auparavant le pilote libre, il faut veiller à bien le désinstaller et vérifier que le module `radeon` n'est plus utilisé au moyen de `lsmod`. S'il est encore présent, le désactiver avant de tester X:



```
rmmod radeon
```

Dans ce cas, penser à rajouter un `radeon_blacklist.conf` dans `/etc/modprobe.d/` avec pour contenu:

```
blacklist radeon
```



Bien que l'usage d'un `xorg.conf` soit à proscrire (car déprécié), si la manipulation précédente n'a pas donné satisfaction (X ne parvenant pas à configurer automatiquement tout le reste de votre installation), ATI fournit `aticonfig` pour modifier `/etc/X11/xorg.conf`. Pour créer la majeure partie du fichier :

```
aticonfig --initial --input=/etc/X11/xorg.conf
```

Vérifier le fichier `Xorg.conf`. Notamment la présence dans la section "Screen" du paramètre "DefaultDepth 24", ainsi que dans la section "DRI" du paramètre "Mode 666". Sans on aura un écran noir. Voici un exemple de fichier :


```
Section "ServerLayout"
    Identifier      "X.org Configured"
    Screen 0       "Screen0" 0 0
EndSection

Section "Monitor"
    Identifier      "Monitor0"
    VendorName      "Monitor Vendor"
    ModelName       "Monitor Model"
EndSection

Section "Device"
    Identifier      "Card0"
    Driver          "fglrx"
    VendorName      "ATI Technologies Inc"
    BoardName       "Radeon Mobility X1400"
    BusID           "PCI:1:0:0"
EndSection

Section "Screen"
    Identifier      "Screen0"
    Device          "Card0"
    Monitor         "Monitor0"
    DefaultDepth    24
    SubSection "Display"
        Viewport    0 0
        Depth       24
    EndSubSection
EndSection

Section "DRI"
    Mode 0666
EndSection
```

Répartir ensuite les différentes sections dont X a besoin (aidez-vous des informations de /var/log/Xorg.0.log) dans autant de fichiers dans /etc/X11/xorg.conf.d, par exemple: 05-server.conf, 10-monitor.conf, 20-catalyst.conf 25-screen.conf 30-options.conf.

Configuration double Head avec des écrans indépendants

Un double head Indépendant peut être configuré de la manière habituelle. Cependant, vous voudrez peut-être que le pilote radeon ait une option "ZaphodHeads" qui permette de lier un périphérique spécifique à une sortie de votre choix, par exemple en utilisant:

```
Section "Device"
    Identifier      "Device0"
    Driver          "radeon"
    Option "ZaphodHeads" "VGA-0"
    VendorName      "ATI"
```

```
BusID "PCI: 1:0:0"
Screen 0
EndSection
```

Cela peut être d'un grand secours, car certaines cartes qui ont plus de deux sorties (par exemple une sortie HDMI, une DVI, une VGA), ne choisissent pas et utilisent les sorties HDMI + DVI pour la configuration à double head, à moins de spécifier explicitement "ZaphodHeads" "VGA-0".

En outre, cette option vous permet de sélectionner l'écran que vous souhaitez marquer comme primaire.

Installation d'OpenBox

OpenBox est un gestionnaire de fenêtres très léger, simple et configurable, il est similaire à BlackBox, et FluxBox.

L'installation est simple, il suffit d'installer le paquet openbox et quelques accessoires.

```
$ sudo pacman -S openbox xterm openbox-themes (**openbox-themes :**
collection de themes)
```

Configuration

Pour pouvoir modifier la configuration par défaut:

```
$ mkdir -p ~/.config/openbox
$ cp /etc/xdg/openbox/* ~/.config/openbox
```

Openbox se configure à l'aide de 3 fichiers:

- **rc.xml** : Ce fichier contient l'ensemble des paramètres du gestionnaire de fenêtres, on y retrouve les raccourcis clavier, les règles de placement des fenêtres, la gestion des bureaux...
- **menu.xml** : Ce fichier contient la structure du menu des applications.
- **autostart.sh** : Ce fichier contient, comme son nom l'indique, les programmes à charger au démarrage d'openbox.



Il est possible d'installer les applications obconf²⁾ et obmenu³⁾ pour avoir une interface pour la configuration.

Toute modification peut être prise en compte en lançant:

```
$ openbox --reconfigure
```



Pour permettre le démarrage automatique des applications renseigner le fichier
~/.config/openbox/autostart.sh (par exemple pour tint2)

```
tint2 &
```

Paramétrage du lancement

- **Démarrage manuel**

Mettre dans le ~/.xinitrc et utiliser xinit ou startx:

```
exec dbus-launch openbox-session
```

- **Automatiquement au démarrage**

Installer et configurer un gestionnaire de connexions (par exemple [lightdm](#)).

Personnalisation du bureau

- **Fond d'écran**

```
$ sudo pacman -S feh imagemagick
```

Rajouter dans le fichier ~/.config/openbox/autostart.sh

```
feh --bg-scale /path/to/wallpaper/wallpaper.jpg
```

- **Gestionnaire de fichier**

GAMIN : est un système créé pour surveiller les fichiers (c'est par exemple le dispositif qui empêche de démonter une clef USB tant qu'un processus tient une ressource **GVFS** : est le système de fichiers virtuel permettant d'accéder à des systèmes de fichiers tels que SFTP, WebDAV, SMB, et aussi au contenu des appareils photos numériques

```
$ sudo pacman -S pcmanfm gamin gvfs
```

- **Barre des tâches**

```
$ sudo pacman -S tint2
```

Rajouter dans le fichier ~/.config/openbox/autostart.sh

```
sleep 1 && tint2 &
```

Installation de lightdm

```
$ sudo pacman -S lightdm lightdm-gtk-greeter
```

- **Pour l'activer :**

```
$ systemctl enable lightdm.service
```

- **Pour le démarrer sans rebooter :**

```
$ systemctl start lightdm.service
```

- **Création du script de lancemant**

```
$ sudo nano ~/.xinitrc
```

```
#!/bin/sh  
exec ck-launch-session dbus-launch --exit-with-session openbox-session
```



Bien que consolekit ne soit plus maintenu il est cependant indispensable car il fournit ck-launch-session. Il faut donc l'installer depuis le <https://aur.archlinux.org/>

```
yaourt consolekit
```

- **Paramétrage du greeter**

```
$ sudo nano /etc/lightdm/lightdm.conf
```

```
[SeatDefaults]  
greeter-session=lightdm-gtk-greeter  
user-session=openbox[SeatDefaults]
```

Installation de displaylink

Installer udlfb





Depuis la version 2.6.32 du kernel il est inutile d'installer udlfb car il est intégré au noyau

Installer udlfb depuis le AUR ...

```
$ wget https://aur.archlinux.org/packages/ud/udlfb/udlfb.tar.gz
$ tar xzf udlfb.tar.gz
$ cd udlfb
$ makepkg -si
$ cd ..
```

ou si yaourt est installé

```
$ yaourt udlfb
```

Pour activer automatiquement le module au boot modifier/créer les fichier de conf suivants

```
sudo nano /etc/modprobe.d/udlfb.conf
```

```
options udlfb fb_defio=1
```

```
$ sudo nano /etc/mkinitcpio.conf
```

```
MODULES="... udlfb ..."
FILES="/etc/modprobe.d/udlfb.conf"
```

Puis, reconstruire l'image du Kernel

```
$ sudo mkinitcpio -p linux
```

Installer xf86-video-fbdev-for-displaylink

```
$ wget
https://aur.archlinux.org/packages/xf/xf86-video-fbdev-for-displaylink/xf86-
video-fbdev-for-displaylink.tar.gz
$ tar xzf xf86-video-fbdev-for-displaylink.tar.gz
$ cd xf86-video-fbdev-for-displaylink
$ makepkg -si
$ cd ..
```

ou si yaourt est installé

```
$ yaourt xf86-video-fbdev-for-displaylink
```

Processus de boot en mode graphique

Personnaliser GRUB

Pour personnaliser le menu de boot éditer `/etc/default/grub` pour utiliser une image de fond...

```
GRUB_BACKGROUND="/boot/grub/myimage"
```

... ou un theme graphique ...

```
GRUB_THEME="/usr/share/grub/themes/starfield/theme.txt"
```

Générer le menu de boot :

```
$ sudo grub-mkconfig -o /boot/grub/grub.cfg
```

fb splash

Fbsplash (anciennement gensplash) est une implémentation en espace utilisateur d'un écran de démarrage pour les systèmes Linux. Il fournit un environnement graphique au démarrage du système en utilisant la couche framebuffer de Linux.

Installer fbsplash depuis le <https://aur.archlinux.org/>

```
$ wget https://aur.archlinux.org/packages/fb/fbsplash/fbsplash.tar.gz
$ tar xzf fbsplash.tar.gz
$ cd fbsplash
$ makepkg -sic
$ cd ..
```

ainsi que le theme souhaité :

```
$ wget
https://aur.archlinux.org/packages/fb/fbsplash-theme-arch-black/fbsplash-the
me-arch-black.tar.gz
$ tar xzf fbsplash-theme-arch-black.tar.gz
$ cd fbsplash-theme-arch-black
$ makepkg -sic
$ cd ..
```

ou avec yaourt

```
$ yaourt fbsplash-theme-arch-black
```

Les thèmes suivants sont disponibles dans le <https://aur.archlinux.org>

Nom du thème	Description
fbsplash-theme-arch-black	Un thème Arch Linux fbsplash
fbsplash-theme-arch-black-extras	Résolutions supplémentaires et des outils pour arch-black
fbsplash-theme-arch-black-spin	Un thème fbsplash Arch Linux avec le logo 3D . Basé sur un arch-black 1.2 thème par PirateJonno
fbsplash-theme-arch-elegant	Thème simple et élégante pour fbsplash
fbsplash-theme-arch-minimal	Un thème fbsplash ArchLinux , conçu avec l'esprit KISS.
fbsplash-theme-archax	thème minimal Arch Linux fbsplash avec le logo black
fbsplash-theme-darikol	Darikol est un thème fbsplash pour archlinux
fbsplash-theme-itty-domy	Itty-Domy est un thème fbsplash pour archlinux
fbsplash-theme-natural-arch	Un thème fbsplash avec logo natural-arch
fbsplash-themes-arch-banner	fbsplash thèmes avec logos des bannières Arch Linux

Contrôler dans /etc/conf.d/splash

La présence du thème souhaité :

```
SPLASH_THEMES="
  arch-black
  arch-banner-icons/1280x1024.cfg
  arch-banner-noicons/1280x1024.cfg"
```

Configuration de Grub

Ajouter les options de kernel dans /etc/default/grub:

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet console=tty1 splash=silent,theme:arch-
banner-icons"
...
GRUB_GFXMODE=1280x1024x24
GRUB_GFX_PAYLOAD_LINUX=keep
```

Regénérer grub.cfg

```
$ grub-mkconfig -o /boot/grub/grub.cfg
```

Configuration du noyau

Pour démarrer au plus tôt le boot graphique rajouter dans le fichier /etc/mkinitcpio.conf le path du theme utilisé dans la section FILE

```
FILES=////
```

...ainsi que le HOOK...

```
H00KS="base fbsplash udev autodetect [...] keymap encrypt uresume"
```

Désactiver le HOOK fbsplash dans fallback

Pour se prémunir d'un problème quelconque avec fbsplash qui bloquerait le démarrage, il peut être judicieux de ne pas générer l'image fallback avec ce HOOK.

Rajouter dans (/etc/mkinitcpio.d/linux..preset

```
fallback_options="-S autodetect,fbsplash"
```

Recompiler le noyau

```
$ mkinitcpio -p linux
```

Configuration de multitouch

Pour charger le module au démarrage modifier/créer le fichier de conf suivant

```
$ sudo nano /etc/modules-load.d/hid-multitouch.conf
```

```
#Load 'hid_multitouch.ko' at boot.  
hid-multitouch
```

Trouver le Vendor ID (VID) et le product ID (PID) de votre touchscreen en utilisant lsusb :

```
...  
Bus 004 Device 002: ID 0eef:725e D-WAV Scientific Co., Ltd  
...
```

Récupérer la MTCLASS* dans <http://lxr.free-electrons.com/source/drivers/hid/hid-multitouch.c>

Récupérer la valeur du MTCLS* correspondant

```
define MT_CLS_EGALAX                                0x0103
```

Convertir cette valeur en décimal

Après avoir chargé le module hid-multitouch il faut lui passer les options par


```
$ echo BUS VID PID MT_CLASS_* > /sys/module/hid_multitouch/drivers/hid\:hid-multitouch/new_id
```

exemple

```
$ echo 3 0eef 725e 259 > /sys/module/hid_multitouch/drivers/hid\:hid-multitouch/new_id
```

Installation de modules complémentaires

Network Manager

```
$ sudo pacman -S networkmanager network-manager-applet xfce4-notifyd  
hicolor-icon-theme  
gnome-icon-theme gnome-keyring usb_modeswitch
```

Activer le démon

```
$ nano /etc/rc.conf
```

dans la section DAEMONS (dernière ligne) ajouter et remplacer network par networkmanager dbus

```
DAEMONS=(syslog-ng dbus networkmanager crond .. etc. ..)
```

Chromium

Chromium est un navigateur internet libre, son moteur de rendu est basé sur WebKit tout comme le navigateur Safari de Apple. Chromium sert également de base pour le navigateur Chrome de Google. Il a pour avantage d'être plus rapide à se lancer que les autres navigateurs connus (Firefox entre autres) bien qu'il soit plus gourmand en RAM.

```
$ sudo pacman -S chromium
```

linux-headers

Pour pouvoir compiler certains paquets il faut avoir installé les headers au préalable

```
$ sudo pacman -S linux-headers
```

Yaourt

Yaourt est un programme en ligne de commande qui complète et améliore les fonctions de pacman pour la gestion des paquetages sur Archlinux.

```
$ wget  
https://aur.archlinux.org/packages/pa/package-query/package-query.tar.gz  
$ tar zxvf package-query.tar.gz  
$ cd package-query  
$ makepkg -sic  
$ cd ..
```

```
$ wget https://aur.archlinux.org/packages/ya/yaourt/yaourt.tar.gz  
$ tar zxvf yaourt.tar.gz  
$ cd yaourt  
$ makepkg -sic  
$ cd ..
```

XEN

Pré requis

Evaluation

Paravirtual (PV)

Les guests paravirtualisés ont besoin d'un noyau construit avec le support de Xen. C'est le cas par défaut pour tous les noyaux Linux récents et quelques autres systèmes de type Unix. Les Domu paravirtualisés sont généralement plus performants que les domaines HVM car ils n'ont pas à tourner sur le matériel émulé.

Matériel virtuel (HVM)

Pour les systèmes d'exploitation qui n'ont pas nativement le support de Xen (par exemple, Windows), HVM offre une virtualisation matérielle complète. Pour utiliser Xen HVM dans le matériel du système hôte ce dernier doit inclure soit la technologie Intel VT-x soit le support de virtualisation AMD-V (SVM).

Compatibilité

Afin de vérifier la compatibilité du système, exécutez la commande suivante sur le système hôte:

```
$ grep -E "(vmx|svm)" -color=always /proc/cpuinfo
```

Si la commande ci-dessus ne produit pas de sortie, le support de virtualisation matériel n'est pas disponible et votre matériel n'est pas en mesure d'exécuter Xen HVM invités. Il est également

possible que le processeur hôte prenne en charge l'une de ces fonctionnalités, mais que la fonctionnalité soit désactivée par défaut dans le BIOS du système. Pour vérifier cela, accéder au menu d'accueil du système de configuration du BIOS lors du démarrage et chercher une option liée à la prise en charge de la virtualisation. Si une telle option existe et est désactivé, l'activer, démarrer le système et répétez la commande ci-dessus.

Installation

Obtention de Xen

Xen est disponible à partir de AUR. La version stable recommandée actuelle est Xen 4.2. Les paquets fournissent l'hyperviseur Xen, l'interface courant xl et tous les fichiers de configuration et de soutien, y compris les services systemd.

Xen, contrairement à certains autres systèmes de virtualisation, s'appuie sur une installation complète du système d'exploitation de base. Avant d'essayer d'installer Xen, votre machine hôte doit avoir une installation entièrement opérationnelle et up-to-date de Arch Linux. Si vous construisez un nouvel hôte à partir de zéro, voir le Guide d'installation pour obtenir des instructions sur l'installation de Arch Linux.

Comme tous les paquets AUR, les binaires sont construits à partir de Xen source. Il est possible (mais pas nécessaire) de construire le paquet sur une machine séparée et transférer le paquet xz dessus, en supposant que les machines partagent la même architecture (par exemple x86_64). Pour Xen, une connexion Internet est nécessaire lors de sa compilation, car les fichiers sources supplémentaires sont téléchargés au cours du processus. Xen.org recommande un hôte disposant d'une architecture 64-bit. Cela nécessite l'activation du dépôt 'multilib' dans /etc/pacman.conf.

Pour les architectures AMD :

Installer le paquet amd-unicode

```
$ sudo pacman -S amd-unicode
```

Décommenter et adapter l'option MAKEFLAGS dans /etc/makepkg.conf (en fonction du nombre de coeur)

```
MAKEFLAGS="-j2"
```



Pour connaître le nombre de processeurs installés, utiliser la commande **nproc**

Pour construire le paquet, vous aurez besoin des packages suivants:

```
$ sudo pacman -S zlib lzo2 python2 ncurses openssl libx11 yajl libaio glib2  
bridge-utils  
$ iproute gettext dev86 bin86 markdown iasl git wget
```

Il faut activer le dépôts «extra» pour obtenir bin86. Un outil tel que yaourt ou packer peut aider à télécharger, compiler et installer les dépendances pour les paquets AUR.

... ainsi que des packages (optionnels) suivants:

```
$ sudo pacman -S ocaml ocaml-findlib
```

Pour installer Xen (sans yaourt) téléchargez, compilez et installez les dépendances des paquets AUR.

```
$ wget https://aur.archlinux.org/packages/xen/xen/xen.tar.gz  
$ tar xzf xen.tar.gz  
$ cd xen/  
$ makepkg -si  
$ cd ..
```

Configuration de Xen

Les étapes de configuration suivantes sont requises une fois que le paquet Xen est installé.

L'hôte dom0 nécessite

- une entrée dans le fichier de configuration du bootloader
 - l'ouverture au moment du démarrage du service systemd
 - un point de montage du système de fichiers xenfs
 - une configuration pontée (bridged) de mise en réseau

En plus de ces mesures qui s'imposent, le wiki xen.org comporte une section sur les meilleures pratiques pour l'exécution de Xen. Il comprend des informations sur l'affectation d'une quantité fixe de mémoire et comment dédier (pin) un noyau CPU au dom0.

Configuration Bootloader

Vous devez démarrer un noyau xen spéciale (xen.gz) qui, à son tour, boot sur votre noyau système normal. Une nouvelle entrée du bootloader est nécessaire. Pour amorcer le système Xen, nous avons besoin d'une nouvelle menuentry dans grub.cfg.

Le package générateur de Xen fournit un fichier de grub2: /etc/grub.d/09_xen.

Ce fichier peut être édité pour personnaliser les commandes de démarrage Xen, et ajoutera un menuentry à votre grub.cfg lorsque la commande suivante est exécutée:

```
grub-mkconfig -o /boot/grub/grub.cfg
```



Si le système est installé sur un volume logique btrfs ne pas oublier de “flagger” le subvolume. Pour cela, avant de régénérer le fichier grub.cfg, ajouter l'option rootflag la ligne **GRUB_CMDLINE_LINUX_DEFAULT** dans /etc/default/grub du fichier /etc/default/grub:

```
GRUB_CMDLINE_LINUX_DEFAULT="rootflag=subvol=arch_root quiet  
console=tty1 splash=silent,theme:arch-banner-icons"
```

Polkit

polkit est un ensemble d'outils permettant de gérer des règles pour autoriser la communication entre, d'un côté, des processus privilégiés offrant des services et de l'autre, des processus non privilégiés. Xen utilisant des processus privilégiés il peut être intéressant, voire indispensable (cf. [interface d'administration graphique](#)), de fournir aux utilisateurs non privilégiés la possibilité de lancer de tels processus.

```
$ pacman -S polkit-gnome
```

Pour démarrer polkit automatiquement rajouter dans le fichier ~/.config/openbox/autostart.sh

```
sleep 1 && /usr/libexec/polkit-gnome-authentication-agent-1 &
```

Service systemd

- **Exécuter les commandes suivantes en tant que root pour que les services soient démarrés au boot:**

```
$ sudo systemctl enable xenstored.service  
$ sudo systemctl enable xenconsole.service  
$ sudo systemctl enable xendomains.service
```

- **Définir un point de montage pour Xenfs**

Inclure dans votre fichier /etc/fstab

```
none /proc/xen xenfs defaults 0 0
```

- **Mise en réseau pontée (bridged)**

Les précédentes versions de Xen fournissent une connexion en pont tandis que Xen 4.2 exige que les communications réseau entre le client, l'hôte (et au-delà) soient misent en place séparément.

L'utilisation de DHCP et de l'adressage statique est possible, et le choix doit être déterminé par la topologie de votre réseau. Avec un réseau pontée de base, un commutateur virtuel est créé dans dom0 auquel chaque domu est attaché.



Netcfg simplifie grandement la configuration du réseau et est maintenant inclus en standard dans le package de base. Des fichiers d'exemple de configuration sont fournis dans `/etc/network.d/examples` et Xen 4.2 fournit des scripts pour différentes configurations de réseau dans `/etc/xen/scripts`.

Par défaut, Xen s'attend à trouver un pont nommé `xenbr0`. Pour le mettre en place avec `netcfg`, procéder comme suit:

```
$ cd /etc/network.d
$ sudo cp examples/bridge xenbridge-dhcp
```

apporter les modifications suivantes à `xen-bridge`:

```
INTERFACE = "xenbr0"
BRIDGE_INTERFACE = "eth0" # Utilise le nom de l'interface externe trouvé
avec la commande 'ip link'
DESCRIPTION = "Xen pont de liaison"
```

en supposant que votre connexion existante `eth0` est appelé `eth0-dhcp`, éditez le fichier `/etc/conf.d/netcfg`

```
NETWORK = (eth0-dhcp dhcp-xenbridge)
```

redémarrer le réseau:

```
$ sudo systemctl restart netcfg.service
```

vérifier que tout va bien

```
$ ip addr show
$ brctl show
```

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
3: xenbr0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
    link/ether 00:1a:92:06:c0:c0 brd ff:ff:ff:ff:ff:ff
```

```
inet 192.168.1.3/24 brd 192.168.1.255 scope global xenbr0
inet6 fe80::21a:92ff:fe06:c0c0/64 scope link
    valid_lft forever preferred_lft forever
```

bridge name	bridge id	STP enabled	interfaces
xenbr0	8000.001a9206c0c0	no	eth0

Interface d'administration graphique

Pour cela on choisit virt-manager.

Installation

```
$ pacman -S libvirt virt-manager
```

PolicyKit authorisation

Pour permettre à un utilisateur non root de lancer virt-manager il convient de définir une policy [polkit](#), pour cela ajouter dans le fichier /etc/polkit-1/rules.d/50-org.libvirt.unix.manage.rules

```
polkit.addRule(function(action, subject) {
    if (action.id == "org.libvirt.unix.manage" &&
        subject.user == "<nom de l'utilisateur>") {
        return polkit.Result.YES;
    }
});
```

Étapes finales

Redémarrer l'hôte dom0 et veiller à ce que le noyau Xen démarre correctement et que tous les paramètres survivent à un redémarrage. Un dom0 configuré correctement devrait afficher les informations suivantes lorsque vous exécutez xl liste (en tant que root):

```
# xl list
Name                               ID   Mem VCPUs  State   Time(s)
Domain-0                           0   511    2    r-----
41652.9
```

Bien sûr, les colonnes Mem, vCPU et Time seront différentes selon la configuration de la machine. La chose importante est que dom0 est répertorié.

Extended display identification data (EDID = Données d'identification d'affichage étendues) est une structure de données fournies par un dispositif d'affichage numérique pour décrire ses capacités à une source vidéo (par exemple carte graphique ou set-top-box

²⁾

obconf : gestionnaire de fenêtres Openbox sur GNU/Linux

³⁾

obmenu : application graphique qui permet d'éditer le fichier menu.xml très facilement

From:

<http://www.ouarte.garden/> - **dwndoc**

Permanent link:

<http://www.ouarte.garden/doku.php?id=virtualisation:xen-archlinux>

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

