

Petit guide d'installation d'ArchLinux avec Gnome 3.14.x / KDE SC / Plasma 5.1.1 / Xfce / Cinnamon / Mate Desktop.

Dans ce petit guide, je vais détailler l'installation d'ArchLinux avec Gnome 3.14 et suivante, KDE SC, Plasma 5.1.1, Xfce, Cinnamon et Mate Desktop. L'installation terminée proposera un environnement suffisamment étoffé pour être utilisable. C'est une adaptation des précédents guides.

NOTE : Les spécificités concernant l'UEFI sont clairement indiquées par un **Spécial UEFI** en face du texte en question.

Cependant, certains points plus techniques et/ou plus particuliers ont été volontairement oubliés : ce sont les manipulations liées à l'installation du pare-feu, ou encore du matériel comme des imprimantes avec scanners.

Je vous renvoie aux wikis anglophone <https://wiki.archlinux.org/> et francophone <http://wiki.archlinux.fr/Accueil> pour ce genre de manipulations.

La variété du dernier point ne rentre pas en effet dans le cadre de ce petit guide.

NOTE : Ce tutoriel est **volontairement simplifié**. Il va vous permettre de voir comment installer une ArchLinux. Ensuite, s'il y a des spécificités liées à votre matériel, c'est au cas par cas qu'il faut regarder et compiler frénétiquement les Wikis ci-dessus.

Pour cette version du guide, je me suis basé sur la dernière ISO officielle, celle qui utilise les scripts d'installation. En décembre 2014, c'est la 2014.12.01.

Enfin, j'ai utilisé une machine virtuelle pour la simple et bonne raison que cela permet d'avoir un environnement standardisé. **Si vous suivez ce guide pour installer sur une machine réelle, il faudra penser à prendre des notes sur le matériel, spécialement votre carte vidéo.** Suis-je clair ?

Ce document est proposé sous licence CC-BY-SA. Merci à Ewolnux pour les correctifs apportés lors d'une précédente version, intégrés depuis.

<http://creativecommons.org/licenses/by-sa/3.0/>

Il a été créé avec LibreOffice 4.3.4, avec une distribution ArchLinux, Gnome 3.14.2, GDM et les outils proposés. Des addenda sont présents pour KDE SC, Plasma 5.x, Xfce, Cinnamon et Mate Desktop.

I. Installons notre base

Installer une ArchLinux, c'est comme construire une maison. On commence par les fondations, et on rajoute les murs et le reste par la suite. L'image ISO utilisée est la archlinux-2014.12.01-dual.iso, mise en ligne début décembre 2014.

La machine virtuelle est une session Qemu à laquelle j'ai rajouté un disque virtuel de 128 Go. L'installation est comparable dans une machine VirtualBox, sauf précision complémentaire.

Voici donc le premier écran qui nous accueille.

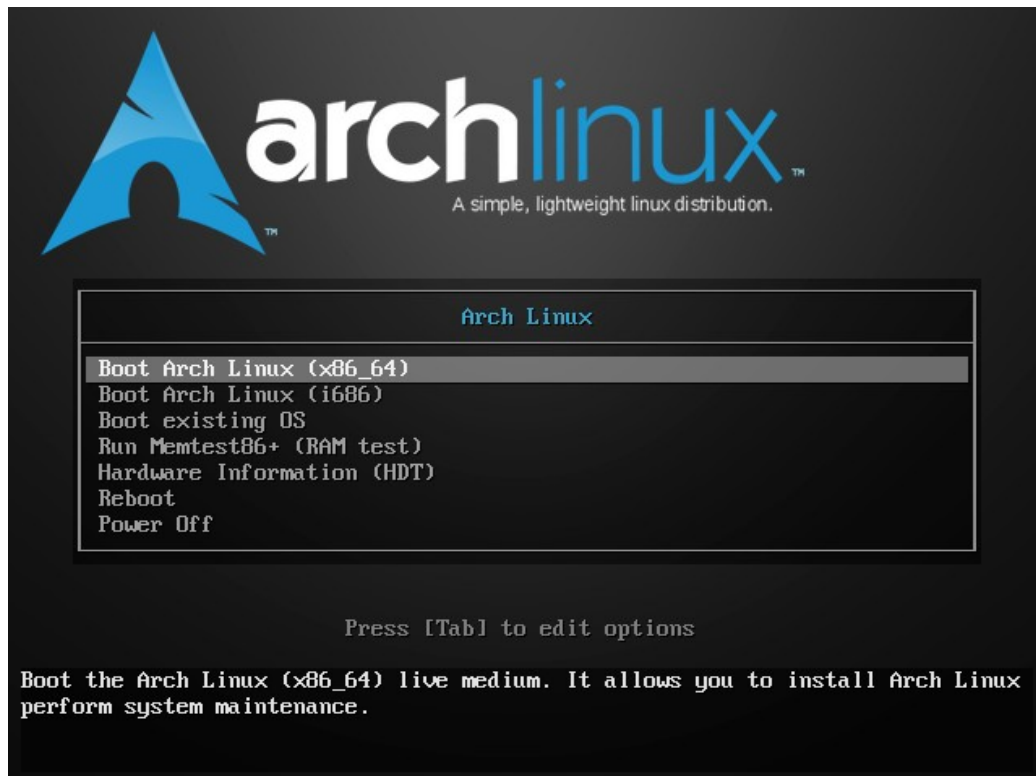


Illustration 1: Ecran de démarrage

On nous demande simplement si on veut faire une installation 32 bits, une 64 bits, avec un noyau « normal ». Si vous voulez utiliser un noyau LTS (long terme), il y a quelques légères modifications à effectuer.

Pour les besoins du guide, j'ai pris une version 64 bits. Sachant que mis à part des outils spécifiques comme Skype, le 64 bits n'a plus à rougir devant une version 32 bits côté équipement logiciel.

Donc si votre processeur supporte le 64 bits, n'hésitez pas ! Il est toujours possible d'utiliser des logiciels 32 bits sur un système 64 bits, nous verrons cela à la fin de l'étape d'installation.

La première chose à faire, c'est d'avoir le clavier français :

```
loadkeys fr
```

Nous passons ensuite à la première étape délicate, à savoir le partitionnement. Utilisant une machine virtuelle, je n'ai pas eu à me soucier outre mesure de ce point précis. Si vous utilisez une machine ayant déjà MS-Windows, il vous faudra avoir créé une partition auparavant... Ou utiliser un disque dédié. Il faudra bien sûr adapter le nom du disque en fonction de la situation.

Dans le cas de notre tutoriel, je vais utiliser le partitionnement suivant :

- partition /boot en /dev/sda1, 512 Mo en ext2
- une partition de swap d'au moins la taille de la mémoire vive en /dev/sda2
- une partition / en /dev/sda3 de 20 Go en ext4
- le reste sera la partition /home, en ext4

On peut utiliser l'outil de son choix, je suis resté classique en utilisant ce bon vieux cfdisk. Au premier démarrage de cfdisk 2.25.2, on peut désormais choisir le type de partitionnement à appliquer :

- gpt (même si je préfère utiliser cgdisk) pour les machines avec circuit UEFI
- dos (l'ancienne option par défaut)

Et deux autres dont j'ignore l'utilité.



Illustration 2: Nouvel écran de démarrage de cfdisk

Spécial UEFI : le partitionnement à appliquer sera le suivant, en utilisant l'outil cgdisk ou gparted pour les plus prudents.

- partition / en /dev/sda1 d'une taille de 20 Go en ext4
- partition EFI en /dev/sda2, d'une taille de 512 Mo, qui sera formaté en Fat32, et référencée en EF00
- partition swap d'au moins la taille de la mémoire vive en /dev/sda3
- le reste sera la partition /home, en ext4

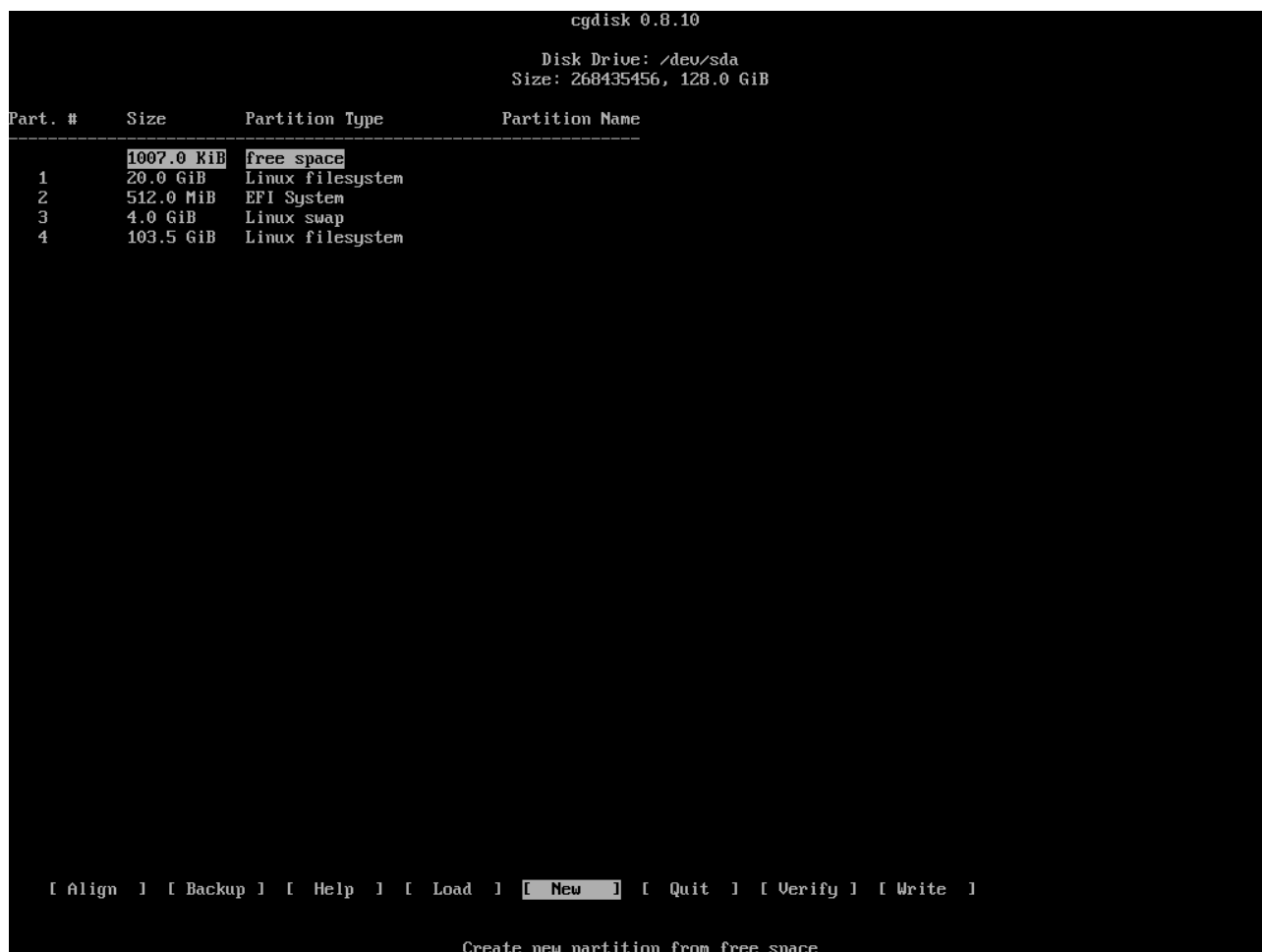


Illustration 3: Partitionnement pour une machine avec l'UEFI

On peut très bien utiliser le liveCD de Gparted si on ne veut pas faire de bêtises. Live CD téléchargeable à l'adresse suivante : <http://www.gparted.org/>

```

Disk: /dev/sda
Size: 128 GiB, 137438953472 bytes, 268435456 sectors
Label: dos, identifier: 0x2d822742

Device      Boot      Start        End    Sectors    Size    Id Type
/dev/sda1          *          2048      1050623     1048576    512M   83 Linux
/dev/sda2          1050624      9439231     8388608     4G    82 Linux swap / Solaris
/dev/sda3          9439232     51382271    41943040    20G    83 Linux
>> /dev/sda4       51382272     268435455    217053184   103.5G   83 Linux

[ Bootable ] [ Delete ] [ Quit ] [ Type ] [ Help ] [ Write ]
```

Illustration 4: Partitionnement avec cfdisk

On va formater les partitions dans le cas d'une machine avec un BIOS :

```
mkfs.ext2 /dev/sda1
mkswap /dev/sda2
swapon /dev/sda2
mkfs.ext4 /dev/sda3
mkfs.ext4 /dev/sda4
```

On va ensuite créer les points de montages et y associer les partitions qui correspondent.

```
mount /dev/sda3 /mnt
mkdir /mnt/{boot,home}
mount /dev/sda1 /mnt/boot
mount /dev/sda4 /mnt/home
```

Spécial UEFI :

```
mkfs.ext4 /dev/sda1
mkfs.fat -F32 /dev/sda2
mkswap /dev/sda3
swapon /dev/sda3
mkfs.ext4 /dev/sda4
```

Et pour les points de montages :

```
mount /dev/sda1 /mnt
mkdir /mnt/{boot,home}
mount /dev/sda2 /mnt/boot
mount /dev/sda4 /mnt/home
```

La connexion est normalement fonctionnelle par défaut. Pour choisir le miroir le plus rapide, il suffit de jeter un œil dans le fichier `/etc/pacman.d/mirrorlist`

Celui d'`archlinux.polymorf.fr`, car c'est le plus rapide pour moi, et comme il est listé dès le départ par ordre de vitesse...

Ensuite, c'est à vous de voir celui qui vous est le plus proche géographiquement parlant ! J'ai donc commenté les autres en rajoutant un `#` devant la ligne `Server`.

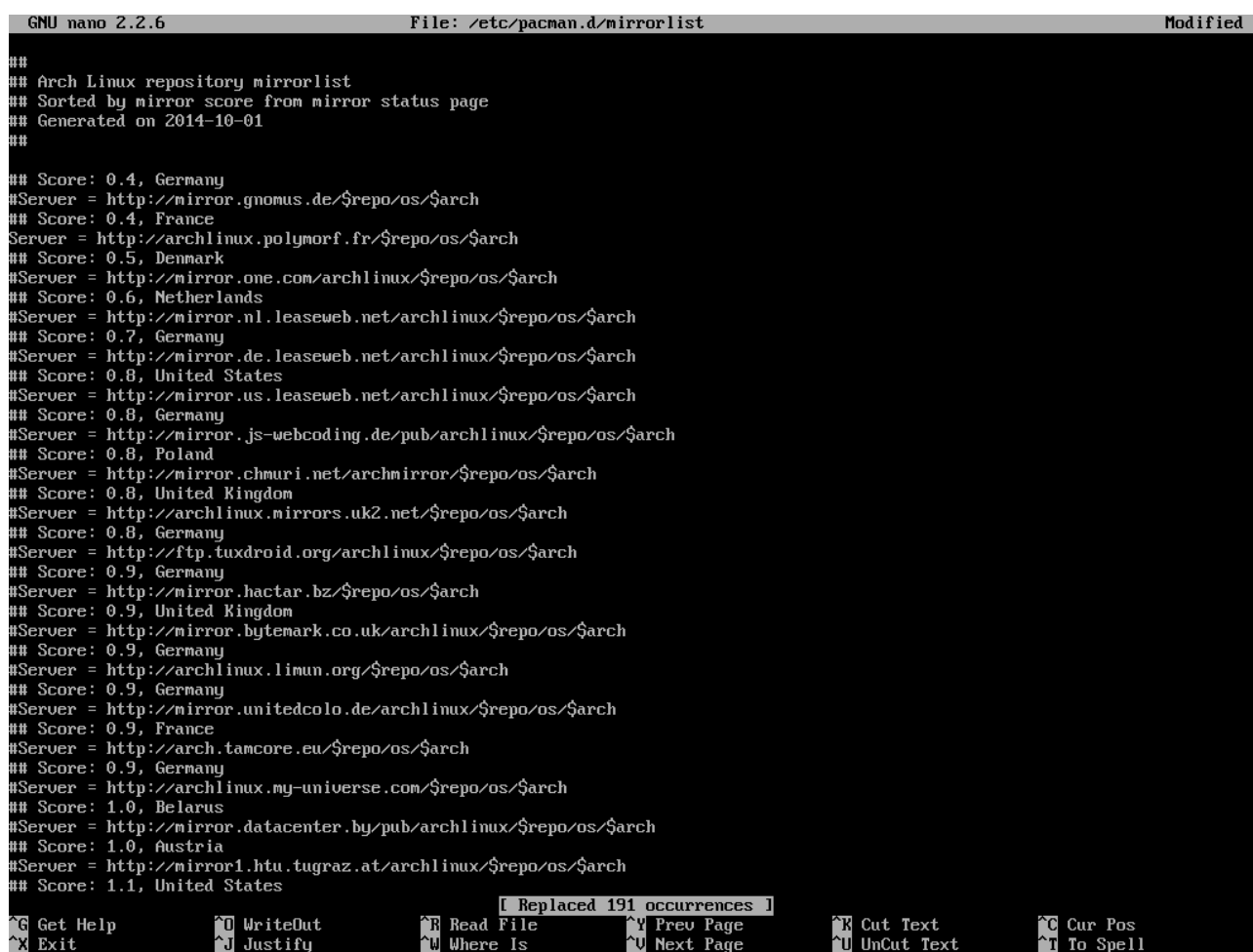


Illustration 5: Sélection du serveur

On passe à l'installation de la base. La deuxième ligne rajoute certains outils bien pratique à avoir dès le départ. Si on veut utiliser un noyau linux long terme, il faut rajouter à la deuxième ligne `pacstrap` le paquet `linux-lts`.

```
pacstrap /mnt base base-devel
pacstrap /mnt zip unzip p7zip vim-minimal mc alsa-utils syslog-ng mtools
dosfstools (sur une seule ligne !)
```

Maintenant que les outils de base sont installés, on va passer aux choses sérieuses. Mais avant tout, il faut générer le fichier `/etc/fstab`.

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

Spécial UEFI : si vous rencontrez des lenteurs au premier démarrage, il faudra commenter la ligne concernant la partition SWAP dans `/etc/fstab`, une fois que vous serez dans le chroot (cf la commande `arch-chroot` un peu plus loin dans le document).

On installe le chargeur de démarrage. Pour mon tutoriel, j'utilise Grub2 qui depuis quelques versions reconnaît automatiquement la version à installer (bios ou UEFI). Le paquet `os-prober` est indispensable si on installe Archlinux en double boot avec MS-Windows ou une autre distribution GNU/Linux.

Spécial UEFI : Le paquet `efibootmgr` n'est indispensable qu'avec l'UEFI.

```
pacstrap /mnt grub os-prober efibootmgr
```

On va passer aux réglages de l'OS qu'on vient d'installer. Il faut donc y entrer.

```
arch-chroot /mnt
```

Étant donné que depuis le 13 octobre 2012, le système installé utilise `systemd`, on va lui créer les fichiers nécessaires à son bon fonctionnement, soit avec `nano` soit avec `vim`.

Pour avoir le bon clavier en mode texte, modifiez le fichier **`/etc/vconsole.conf`**. Pour une personne vivant en France et voulant le clavier azerty, ça donnera cela. La deuxième ligne est la police d'affichage en terminal.

```
KEYMAP=fr-latin9  
FONT=lat9w-16
```

Pour avoir la localisation française, le fichier **`/etc/locale.conf`** doit contenir :

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

Il faut vérifier que la ligne **`fr_FR.UTF-8 UTF-8`** dans le fichier **`/etc/locale.gen`** n'a pas de `#` devant elle. Ainsi que la ligne **`en_US.UTF-8 UTF-8`**. Si vous habitez ailleurs, à adapter en fonction du lieu concerné.

On va maintenant générer les traductions :

```
locale-gen
```

On peut spécifier la locale pour la session courante :

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

Le nom de la machine ? Il est contenu dans le fichier **`/etc/hostname`**. Il suffit de taper le nom sur la première ligne. Par exemple : `fredo-archlinux-gnome`. À remplacer par le nom de la machine bien entendu !

Le fuseau horaire ? Une seule étape. Prenons le cas d'une installation avec le fuseau horaire de Paris. On commence par créer un lien symbolique :

```
ln -sf /usr/share/zoneinfo/Europe/Paris /etc/localtime
```

Ensuite, deux cas se présentent. Soit on a une machine en mono-démarrage sur ArchLinux, et on peut demander à ce que l'heure appliquée soit UTC, soit un double démarrage avec MS-Windows. Prenons le premier cas.

On va demander à ce que l'heure soit par défaut réglée sur UTC :

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

Sinon, on ne touche à rien. Car MS-Windows est un goujat dans ce domaine.

Étape suivante ? Générer le bon fichier grub, installer grub – regardez bien la ligne qui vous convient par la suite et générer le noyau. Si vous voulez un noyau lts, il faudra utiliser **mkinitcpio -p linux-lts** en lieu et place du **mkinitcpio -p linux**

```
mkinitcpio -p linux  
grub-mkconfig -o /boot/grub/grub.cfg
```

Installation en mode BIOS :

```
grub-install --no-floppy --recheck /dev/sda
```

Spécial UEFI :

```
mount -t efivarfs efivarfs /sys/firmware/efi/efivars  
grub-install --target=x86_64-efi --efi-directory=/boot --bootloader-  
id=arch_grub --recheck → sur une seule ligne !
```

De plus, pour éviter tout problème de démarrage (en cas d'ordinateur capricieux, j'y ai été confronté lors de la mise à jour du tutoriel), il est conseillé de rajouter les commandes suivantes :

```
mkdir /boot/EFI/boot  
cp /boot/EFI/arch_grub/grubx64.efi /boot/EFI/boot/bootx64.efi
```



```

[ Wrote 1 line ]
sh-4.3# ln -sf /usr/share/zoneinfo/Europe/Paris /etc/localtime
sh-4.3# hwclock --systohc --utc
sh-4.3# mkinitcpio -p linux
==> Building image from preset: /etc/mkinitcpio.d/linux.preset: 'default'
-> -k /boot/vmlinuz-linux -c /etc/mkinitcpio.conf -g /boot/initramfs-linux.img
==> Starting build: 3.17.4-1-ARCH
-> Running build hook: [base]
-> Running build hook: [udev]
-> Running build hook: [autodetect]
-> Running build hook: [modconf]
-> Running build hook: [block]
-> Running build hook: [filesystems]
-> Running build hook: [keyboard]
-> Running build hook: [fsck]
==> Generating module dependencies
==> Creating gzip-compressed initcpio image: /boot/initramfs-linux.img
==> Image generation successful
==> Building image from preset: /etc/mkinitcpio.d/linux.preset: 'fallback'
-> -k /boot/vmlinuz-linux -c /etc/mkinitcpio.conf -g /boot/initramfs-linux-fallback.img -S autodetect
==> Starting build: 3.17.4-1-ARCH
-> Running build hook: [base]
-> Running build hook: [udev]
-> Running build hook: [modconf]
-> Running build hook: [block]
==> WARNING: Possibly missing firmware for module: aic94xx
==> WARNING: Possibly missing firmware for module: smsdmu
-> Running build hook: [filesystems]
-> Running build hook: [keyboard]
-> Running build hook: [fsck]
==> Generating module dependencies
==> Creating gzip-compressed initcpio image: /boot/initramfs-linux-fallback.img
==> Image generation successful
sh-4.3#

```

Illustration 6: Génération du noyau d'Archlinux, linux 3.17.4 début décembre 2014

Bien entendu, aucune erreur ne doit apparaître.

On donne un mot de passe au compte root :

```
passwd root
```

Pour le réseau, étant donné qu'on va utiliser Gnome par la suite, installer et activer NetworkManager est une bonne idée. Sinon ArchLinux propose un outil qui gère directement les connexions. Quoiqu'il arrive, on aura une connexion fonctionnelle.

```
pacman -S networkmanager
systemctl enable NetworkManager.service
```

Dernier réglage, **optionnel si on veut avoir accès à l'outil yaourt**, il faut ajouter ceci au fichier **/etc/pacman.conf** à la fin. Une fois yaourt installé, on peut enlever les lignes en question.

```
[archlinuxfr]
SigLevel = Optional TrustAll
Server = http://repo.archlinux.fr/\$arch
```

Si vous voulez utiliser un outil comme Skype (qui est uniquement en 32 bits) et que vous installez un système 64 bits, il faut décommenter (enlever les #) des lignes suivantes :

```
#[multilib]
#include = /etc/pacman.d/mirrorlist
```

On peut maintenant quitter tout, démonter proprement les partitions et redémarrer. C'est un peu plus délicat qu'auparavant. Mais au moins, on voit les étapes à suivre.

```
exit
umount -R /mnt
reboot
```

Voilà, on peut redémarrer. On va passer à la suite, largement moins ennuyeuse !

II. Installons maintenant l'environnement graphique !

Nous attaquons donc la partie la plus intéressante, l'installation de Gnome. C'est plutôt simple. Pour KDE SC, Xfce, Cinnamon ou Mate, il faudra se référer aux addenda en fin de document, dès la section « Gnome ».

Note : Il y a une liste de services à activer un peu plus bas dans le document. Ils sont communs à **tous** les environnements du document.

Une fois le système démarré, on se connecte en root. Étant donné que j'ai installé NetworkManager à l'étape précédente, le réseau fonctionne directement. On tape les lignes de commande suivante pour mettre à jour les dépôts et installer yaourt un outil qui va nous simplifier grandement la vie. Ainsi que ntp et cronie (pour les tâches d'administration à automatiser). L'installation de yaourt dépend de l'ajout du dépôt archlinuxfr, bien entendu.

```
pacman -Syy  
pacman -S yaourt ntp cronie
```

Note 2 : Si on veut avoir les logs en clair en cas de problème, il faut modifier le fichier /etc/systemd/journald.conf en remplaçant la ligne

```
#ForwardToSyslog=no
```

par :

```
ForwardToSyslog=yes
```

Les outils en place, on lance alsamixer, pour configurer le niveau sonore :



Illustration 7: Alsamixer en action

Une fois l'ensemble configuré, pour le conserver tel quel, il suffit d'entrer :

```
alsactl store
```

Nous sommes dans le multimédia ? Restons-y.

On va installer l'ensemble des greffons gstreamer qui nous donneront accès aux fichiers multimédias une fois Gnome lancé. Si vous n'avez pas installé yaourt, il faudra le remplacer par **pacman -S** ou **sudo pacman -S** quand vous utiliserez votre compte utilisateur « normal ». Pour l'exécution de la ligne suivante, il est demandé de choisir un support pour opengl. Pour le moment, on choisit MesaGL. La modification correspondant à votre matériel sera faite lors de l'installation de Xorg.

```
pacman -S gst-plugins-base gst-plugins-good gst-plugins-bad gst-plugins-ugly
gst-libav
```

gst-libav ? Il prend en charge tout ce qui est x264 et apparenté.

Passons à l'installation de Xorg. Il suffit de rentrer :

```
pacman -S xorg-server xorg-xinit xorg-xmessage xorg-utils xf86-input-mouse
xf86-video-modesetting xorg-server-utils xorg-apps
```

Il faut ensuite choisir le pilote pour le circuit vidéo. Sur ce point précis, je ne peux

que vous conseiller de lire le wiki anglophone qui donne la liste des principaux pilotes.

https://wiki.archlinux.org/index.php/Xorg#Driver_installation

Dans le cas d'une machine virtuelle, j'ai utilisé le paquet `xf86-video-vesa`. On passe ensuite à l'installation des polices. Voici la ligne de commande pour les principales, y compris les polices de Microsoft (Times, Arial, Courier). Pour les polices Microsoft, le paquet `ttf-ms-fonts`, elles sont sur le dépôt AUR, donc il faut utiliser `yaourt` pour les récupérer et les installer.

```
yaourt -S ttf-bitstream-vera ttf-liberation ttf-freefont ttf-dejavu ttf-ms-fonts
```

Si vous faites une installation dans VirtualBox, deux étapes supplémentaires. D'abord installer le paquet **virtualbox-guest-utils** ou **virtualbox-guest-utils-lts** (si vous utilisez un noyau LTS).

Ensuite, il faut créer un fichier `/etc/modules-load.d/vbox.conf` qui contient les lignes suivantes :

```
vboxguest  
vboxvideo  
vboxsfnet
```

Ensuite, on va rajouter des outils, histoire de ne pas voir un gnome vide au premier démarrage. On commence par tout ce qui est graphique : `gimp`, `cups` et `hplip` (si vous avez une imprimante scanner Hewlett Packard) :

```
yaourt -S cups gimp hplip
```

Depuis la fin juillet 2014, il y a deux versions qui sont supportées par Archlinux pour LibreOffice, en conformité avec ce que propose la Document Foundation. La version « fresh » et la version « stable » (dite still).

La version « fresh » est plus récente, et est par défaut conseillée aux utilisateurs conscients des problèmes qui pourrait se produire. Si on veut la version stable, on utilise la ligne de commande :

```
yaourt -S libreoffice-still libreoffice-still-fr
```

Pour les utilisateurs plus aventureux :

```
yaourt -S libreoffice-fresh libreoffice-fresh-fr
```

On rajoute ensuite Mozilla Firefox en français :

```
yaourt -S firefox-i18n-fr
```

Vous préférez Chromium ?

```
yaourt -S chromium
```

On passe enfin au morceau de choix : l'installation de Gnome, les extensions étant indispensables pour avoir le mode « Gnome Classique ». Le paquet `telepathy-idle` est

nécessité pour faire fonctionner l'ajout de compte en ligne. Si vous voulez un autre environnement, jetez un œil aux addenda, puis revenez pour la gestion des utilisateurs.

```
yaourt -S gnome gnome-extra gnome-tweak-tool system-config-printer xdg-user-dirs telepathy
```

L'installation de Gnome terminée, on crée un utilisateur avec la commande suivante :

```
useradd -m -g users -G wheel -c 'Nom complet de l'utilisateur' -s /bin/bash  
nom-de-l'utilisateur → sur une seule ligne !  
passwd nom-de-l'utilisateur
```

Avant de finir, on va configurer sudo en utilisant visudo. En effet, il nous suffit de modifier une ligne pour que l'on puisse accéder en tant qu'utilisateur classique aux droits complets sur la machine de manière temporaire.

Il faut aller jusqu'à la ligne :

```
##Uncomment to allow members of group wheel to execute any command
```

Et enlever le # sur la ligne qui suit.

Étant donné que systemd est utilisé, voici la liste des services à activer, qui sera la même pour chaque environnement dans les « addenda » :

```
systemctl enable syslog-ng.service  
systemctl enable cronie.service  
systemctl enable avahi-daemon.service  
systemctl enable avahi-dnssconfd.service  
systemctl enable org.cups.cupsd.service  
systemctl enable bluetooth.service  
systemctl enable ntpd.service
```

Note : Dans un premier temps, on ne va pas activer gdm.service. Juste le lancer avec **systemctl start gdm.service**. Si GDM se lance, et que Gnome se charge, on pourra l'activer sans danger par la suite.

Au démarrage suivant, GDM nous accueille. Et nous pouvons nous connecter.

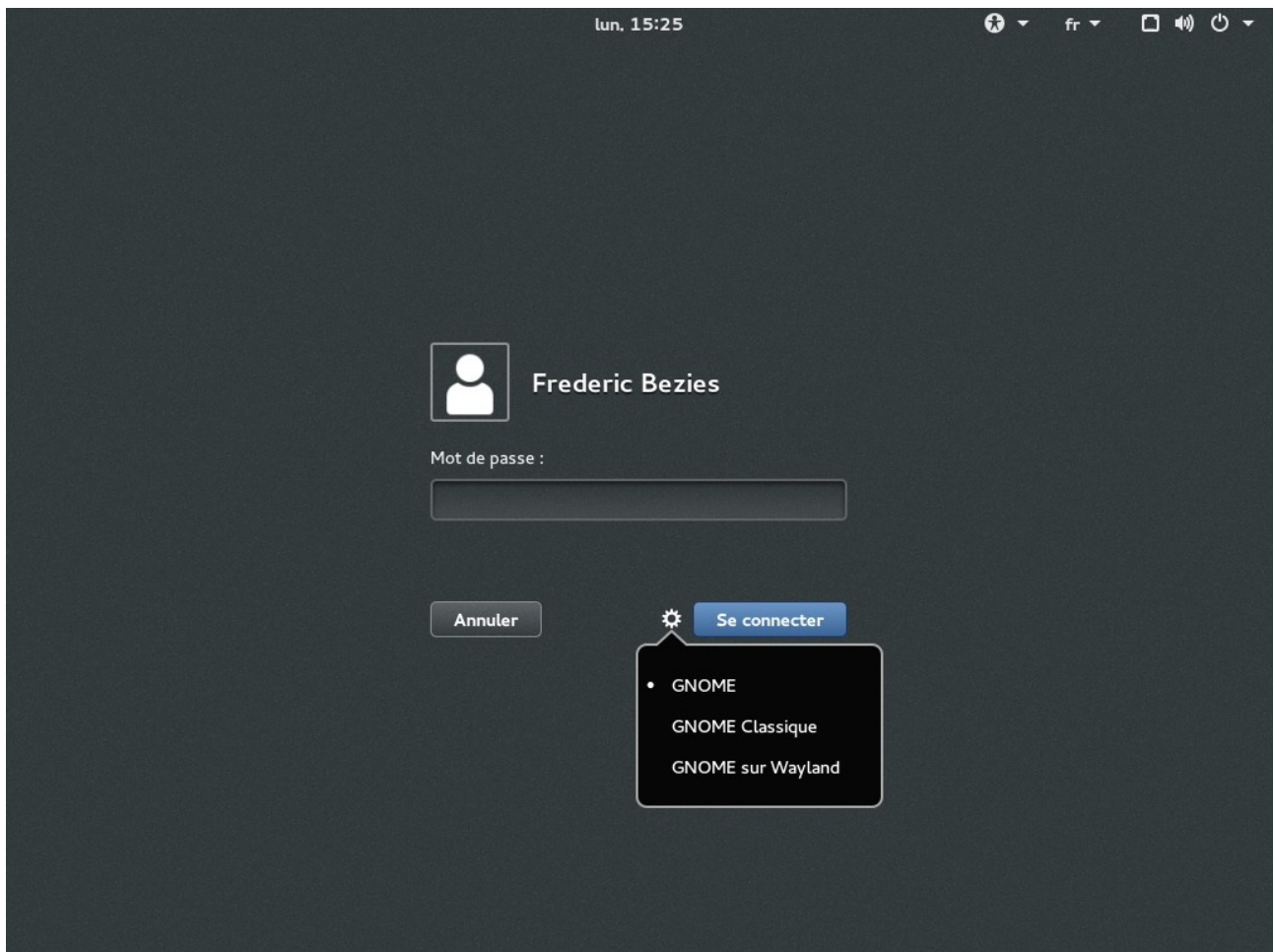


Illustration 8: GDM 3.14.2 avec l'expérimentale session Gnome sur Wayland

III. Finalisons notre installation

Quelques outils à rajouter : xsane (pour le scanner), mais aussi unoconv (pour l'aperçu des fichiers dans Gnome Documents) et Flash...

```
yaourt -S xsane unoconv flashplugin
```

Il faut noter que les répertoires « Documents », « Images », « Téléchargements » sont présents dès le départ. Un bon point !

Il faut penser à configurer le clavier pour qu'il soit en français. Ce qui se fait dans menu système unifié, options de configuration.

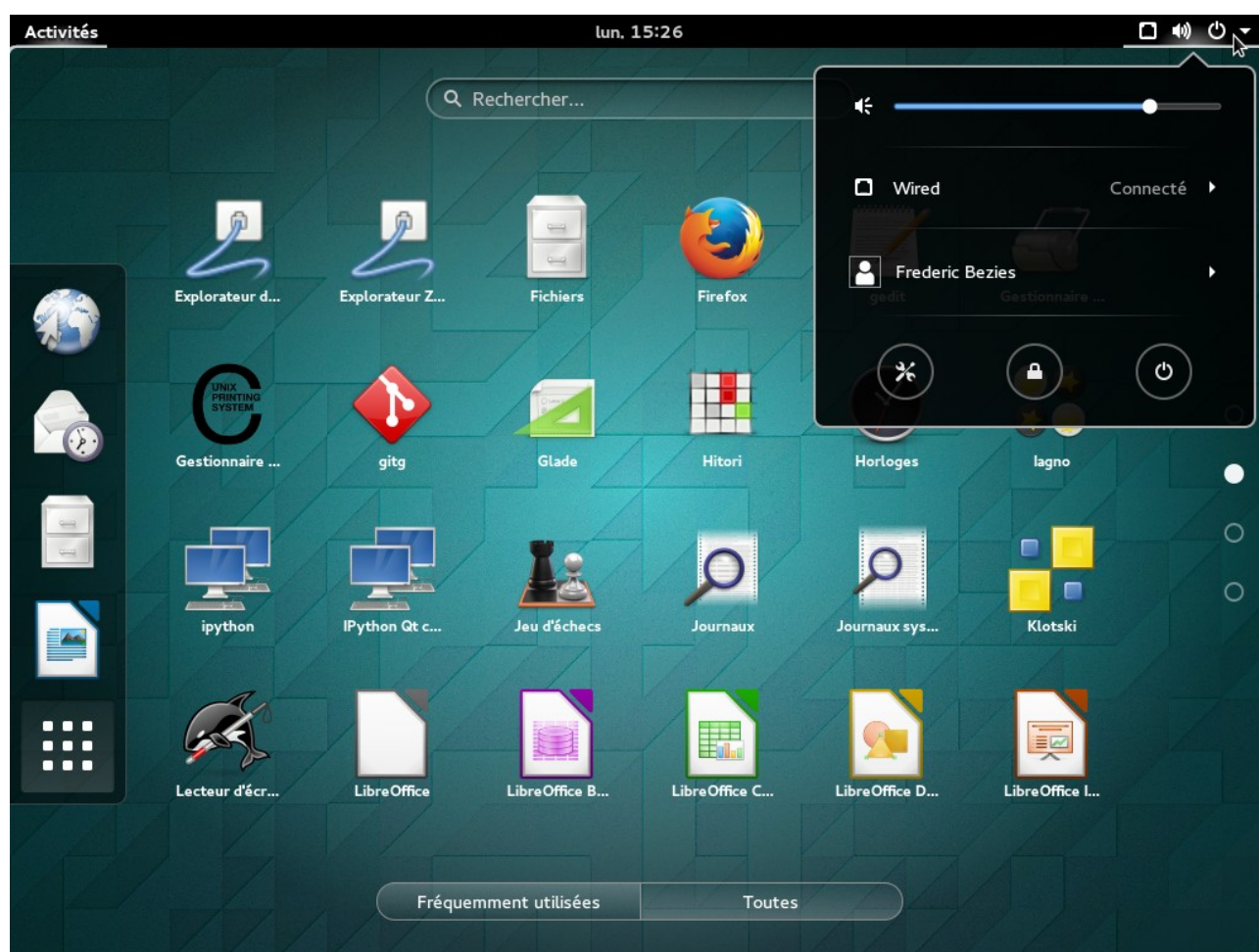


Illustration 9 : Gnome 3.14.2 en vue activités

Pour utiliser le mode Gnome Classique dans sa totalité, il faut activer l'option « Icon On desktop » dans la catégorie Bureau de l'outil Gnome Tweak Tool.

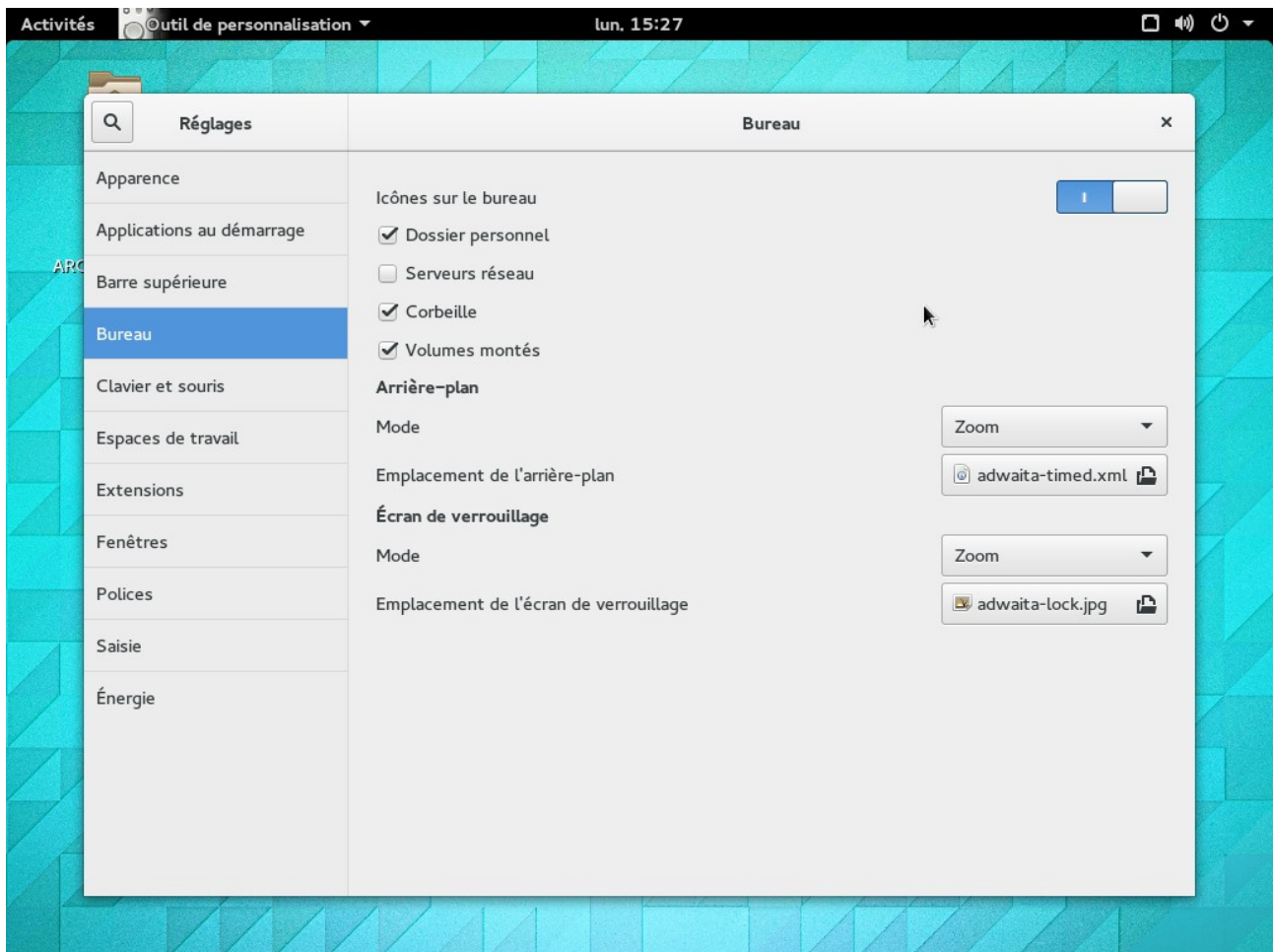


Illustration 10 : Gnome Tweak Tool

Et des captures d'écran du mode « Gnome Shell » et « Gnome Classique ».

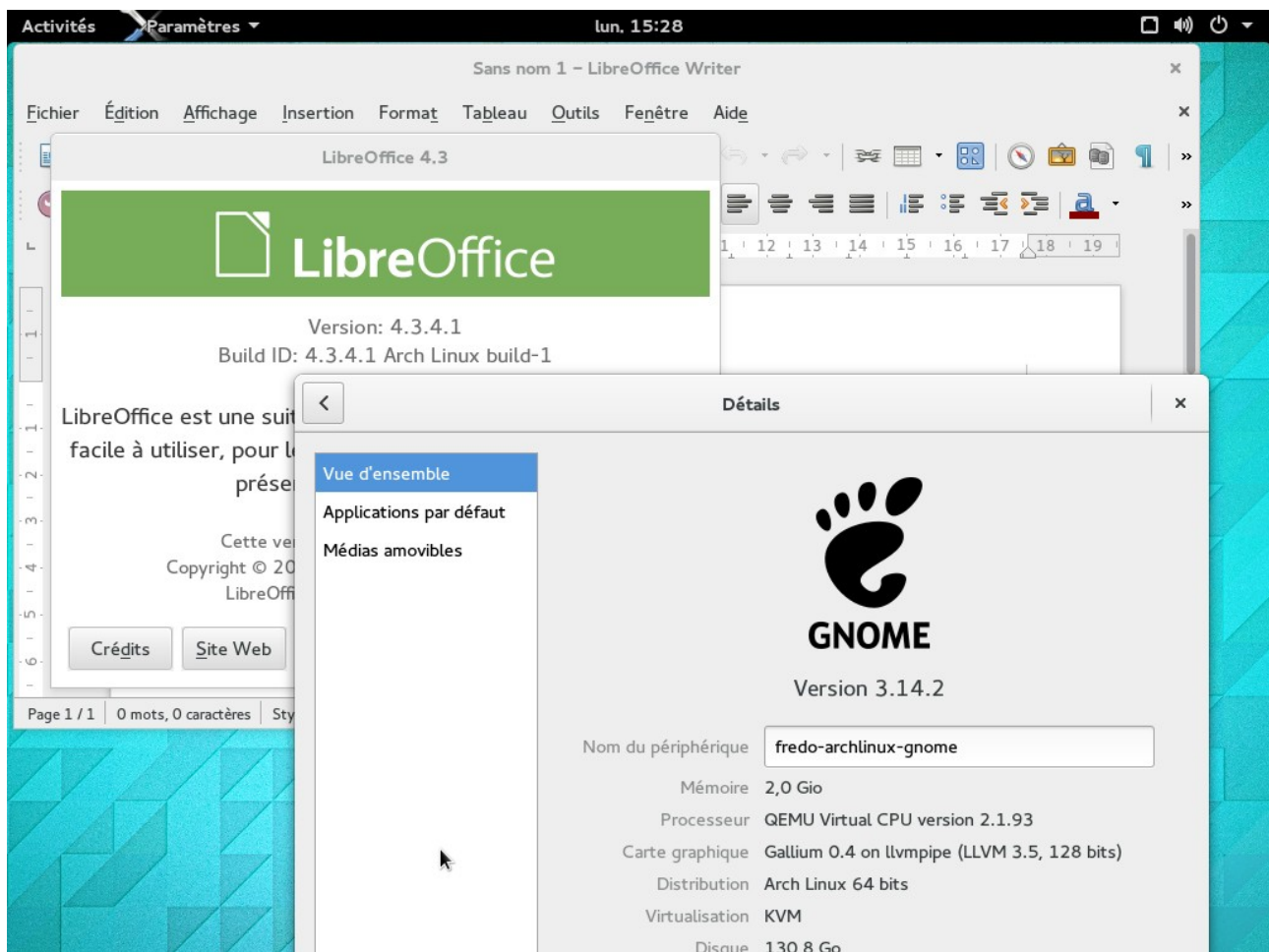


Illustration 11 : Gnome Shell 3.14.2 et « LibreOffice-fresh »

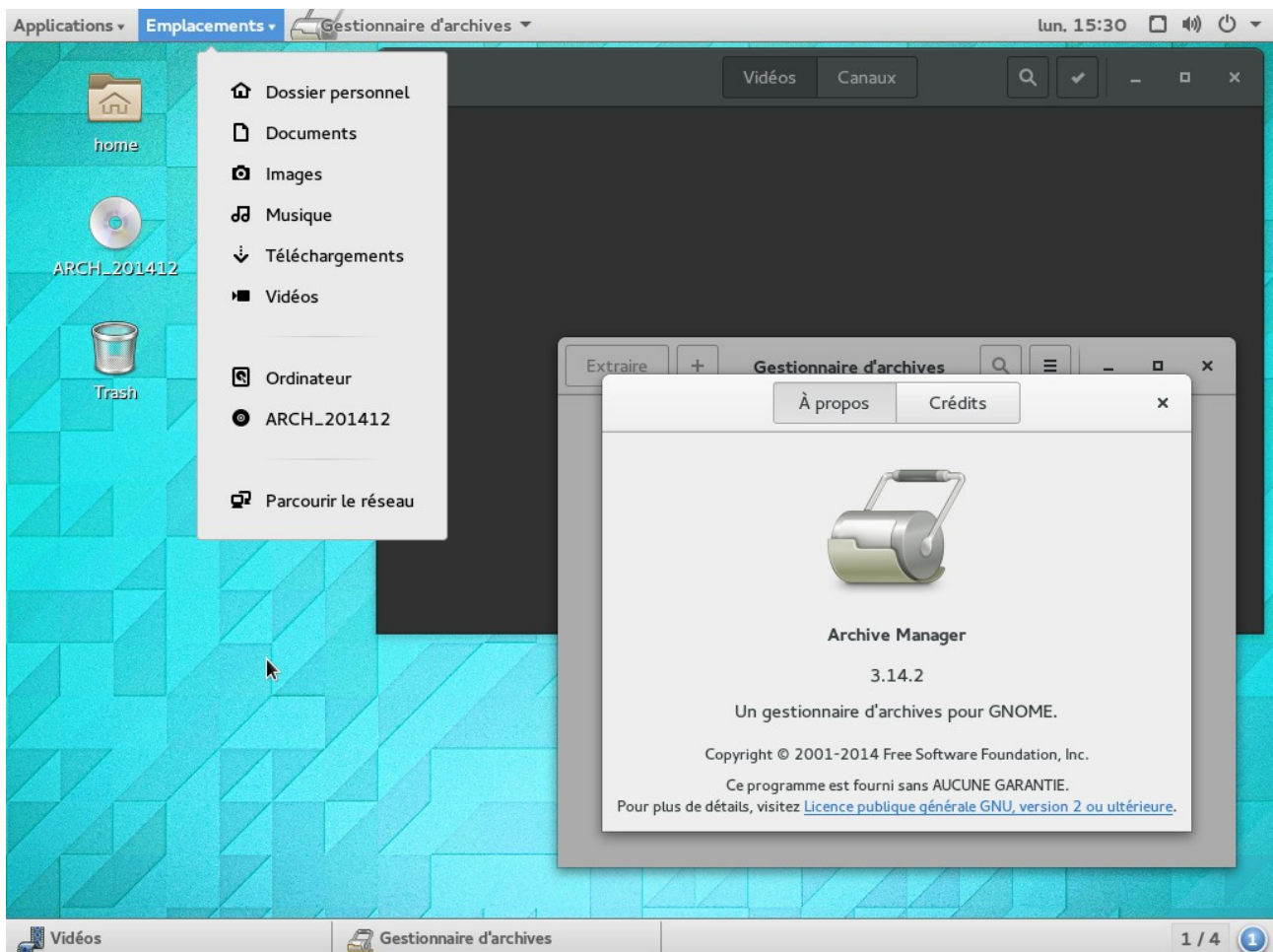


Illustration 12 : Gnome Classique en action

Voilà, le guide est maintenant fini. Cependant, je n'ai pas abordé l'installation d'un pare-feu. C'est quelque chose de plus technique.

J'ai surtout voulu faire un **guide rapide**, histoire de montrer qu'en une petite heure on pouvait avoir un environnement installé et assez complet pour le figurer par la suite.

Bonne découverte !

Addenda 1 : Installer KDE SC / Plasma 5.x :

Pour installer KDE SC, il faut entrer, en demandant l'utilisation pour kactivities de la version kactivities4, l'autre étant prévu pour Plasma 5.1.1 :

```
yaourt -S kde kde-l10n-fr amarok digikam kwebkitpart k3b
```

Pour lancer KDE SC, il faut entrer dans un premier temps :

```
systemctl start kdm.service
```

Et si tout se passe bien, on peut utiliser :

```
systemctl enable kdm.service
```

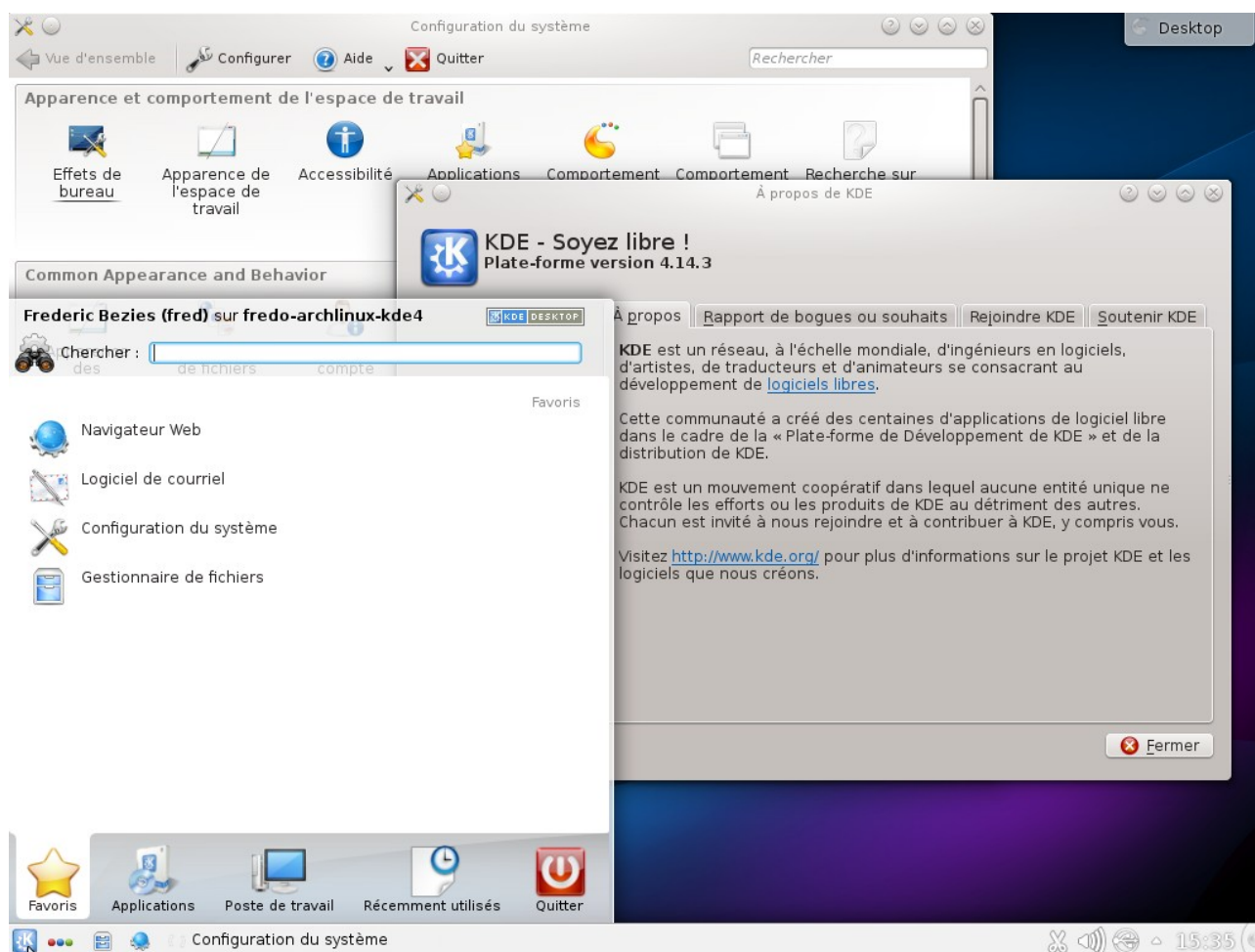


Illustration 13 : KDE SC 4.14.3

Si vous voulez Plasma (alias « KDE 5 »), voici l'installation à faire :

```
yaourt -S kde kde-l10n-fr amarok digikam kwebkitpart k3b (en choisissant d'installer kactivities-frameworks et phonon-qt5-gstreamer) quand c'est demandé.
```

Ensuite, il faut enlever kdebase-workspace tout seul :

```
yaourt -Rdd kdebase-workspace
```

L'installation de Plasma :

```
yaourt -S kf5-aids plasma-next
```

Pour le gestionnaire de connexion, sddm n'étant pas trop configurable pour le moment (le clavier étant en agencement QWERTY par défaut), mais étant plus léger et moins intrusif que lightdm. Il faut espérer que KDM soit porté pour Plasma 5 par la suite.

```
yaourt -S sddm
```

Et pour lancer Plasma 5.1.1, dans un premier temps :

```
systemctl start sddm.service
```

Et si tout se passe bien, on peut utiliser :

```
systemctl enable sddm.service
```

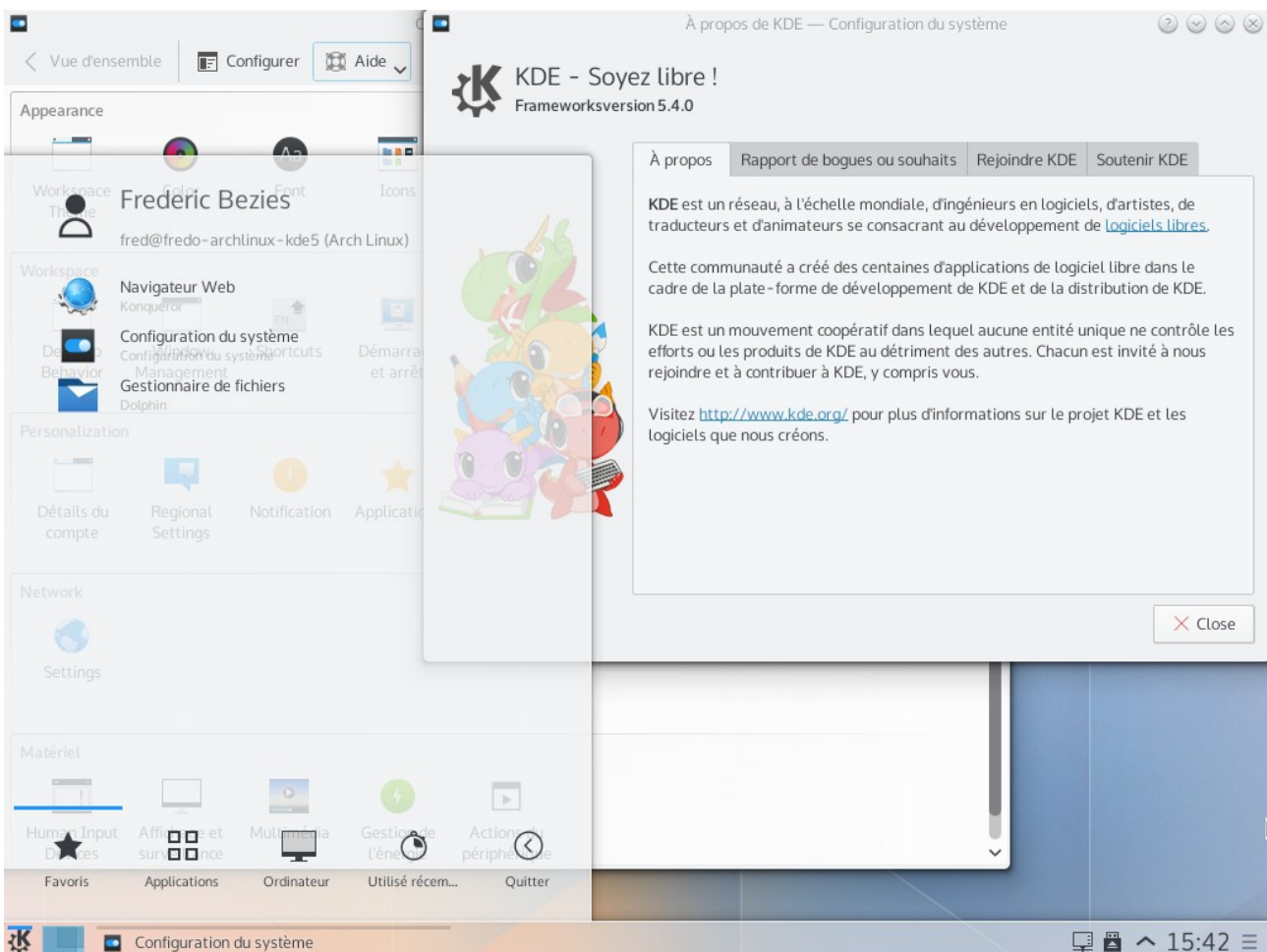


Illustration 13b : Plasma 5.1.1 – KDE Frameworks 5.4 en action.

Addenda 2 : Installer Xfce

Pour installer Xfce, il faut entrer :


```
yaourt -S xfce4 xfce4-goodies gvfs vlc quodlibet-plugins lightdm-gtk2-greeter gnome-icon-theme xdg-user-dirs midori xarchiver claws-mail
```

Vlc et Quodlibet ? Pour la vidéo et l'audio. Midori ? Pour la navigation internet. Pour les périphériques amovibles, gvfs est obligatoire. Claws-Mail ou Mozilla Thunderbird (avec le paquet `thunderbird-i18n-fr`) pour le courrier. Lightdm étant pris, car plus rapide à installer. Si vous utilisez NetworkManager, vous pouvez rajouter l'applet pour gérer et surveiller votre réseau avec le paquet « `network-manager-applet` ».

Pour lancer Xfce, il faut entrer dans un premier temps :

```
systemctl start lightdm.service
```

Et si tout se passe bien, on peut utiliser :

```
systemctl enable lightdm.service
```

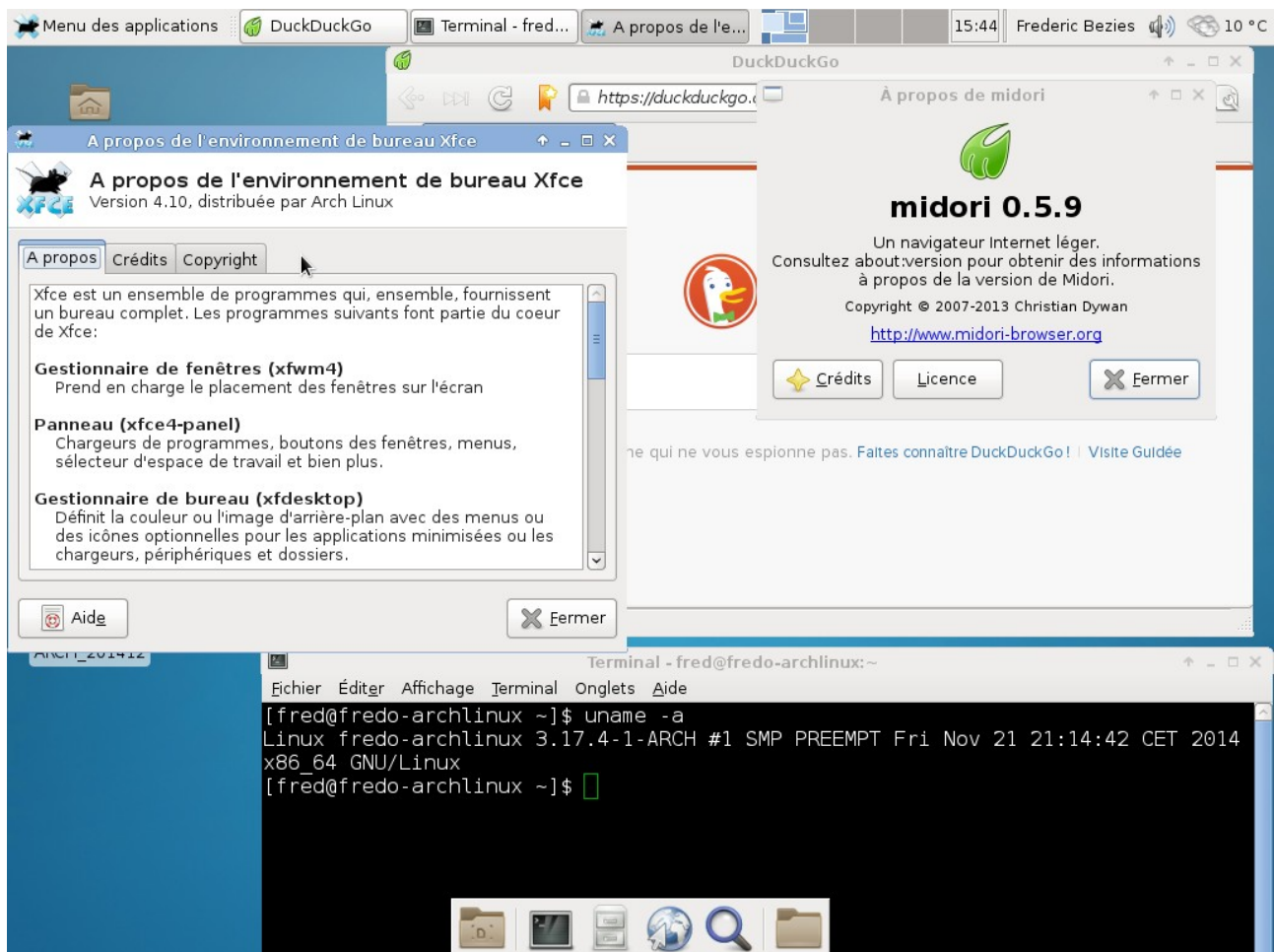


Illustration 14 : Xfce 4.10.1 sous Archlinux.

Addenda 3 : Installer Cinnamon

Pour installer Cinnamon, il faut entrer :

```
yaourt -S cinnamon gnome-extra gnome-terminal lightdm-gtk3-greeter xdg-user-dirs gnome-icon-theme
```

Gnome-extra et gnome-terminal ? Pour avoir les outils gnome pour compléter l'offre de Cinnamon. Lightdm étant utilisé, car plus rapide à installer. Pour lancer Cinnamon, il faut entrer dans un premier temps :

```
systemctl start lightdm.service
```

Et si tout se passe bien, on peut utiliser :

```
systemctl enable lightdm.service
```

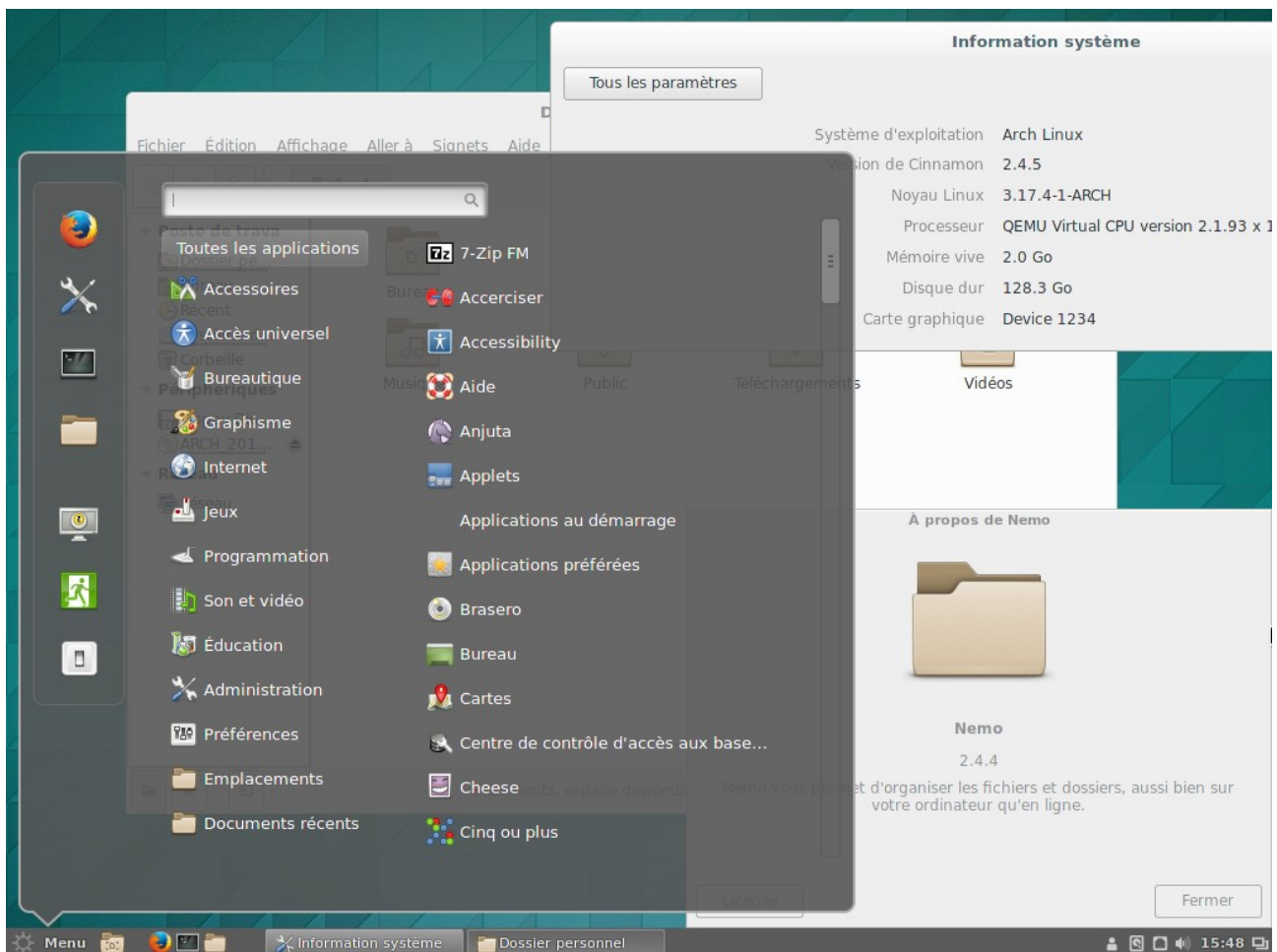


Illustration 15 : Cinnamon 2.4.5 sous Archlinux.

Addenda 4 : Installer Mate Desktop

L'installation ? Elle ressemble à celle de Xfce.

```
yaourt -S mate mate-extra lightdm-gtk2-greeter gnome-icon-theme xdg-user-dirs vlc quodlibet-plugins accountsservice claws-mail
```

Vlc et QuodLibet ? Pour la vidéo et l'audio. Lightdm étant pris, car plus rapide à installer. Claws-Mail ou Mozilla Thunderbird (avec le paquet thunderbird-i18n-fr)

pour le courrier. Si vous utilisez NetworkManager, vous pouvez rajouter l'applet pour gérer et surveiller votre réseau avec le paquet « network-manager-applet ».

Pour lancer Mate Desktop, il faut entrer dans un premier temps :

```
systemctl start accounts-daemon.service  
systemctl start lightdm.service
```

Et si tout se passe bien, on peut utiliser :

```
systemctl enable accounts-daemon  
systemctl enable lightdm.service
```

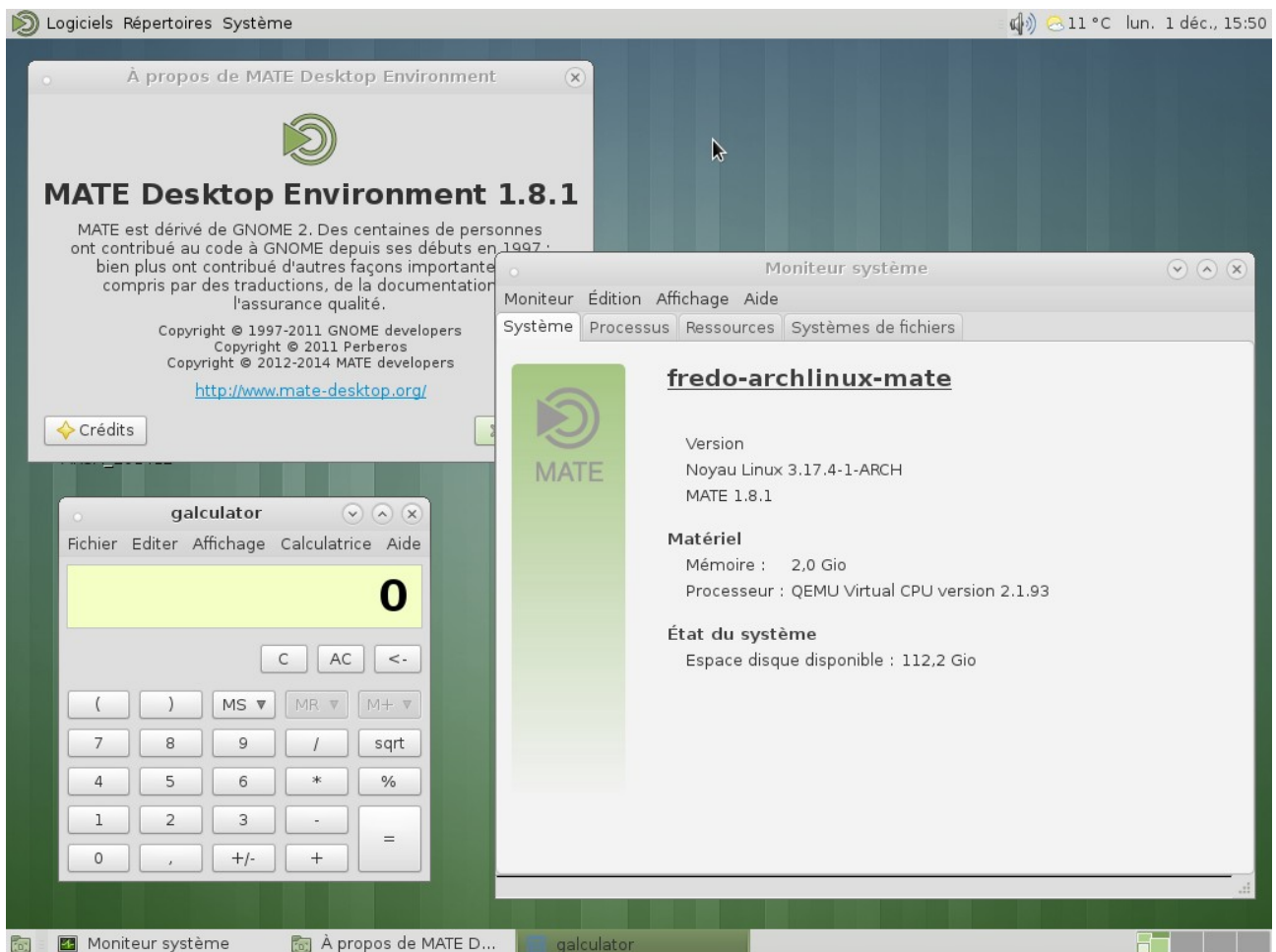


Illustration 16 : Mate 1.8.1 sous Archlinux.