

Comment utiliser des logiciels Linux sur un Chromebook ?

Les chromebooks sont des PC aux caractéristiques techniques plutôt modestes. Ils fonctionnent sous Chrome OS, un système d'exploitation léger qui s'appuie sur le navigateur internet Google Chrome.

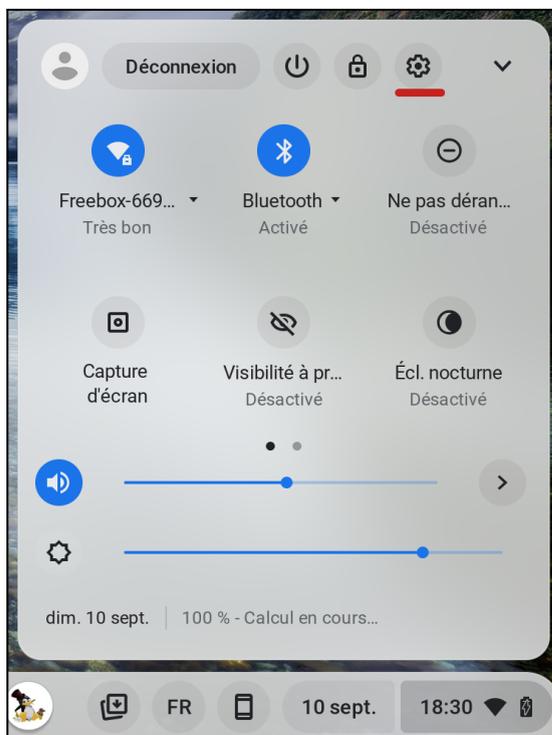
Cet OS possède ses propres applications qui s'installent sous forme d'extensions au navigateur Chrome. Elles sont disponibles sur le Web Store. Chrome OS est également compatible avec les applications Android du Google Play.

Mais même si Google ne le documente pas, on peut également faire fonctionner des logiciels Linux via une machine virtuelle Debian ce qui augmente considérablement le champ d'utilisation du Chromebook.

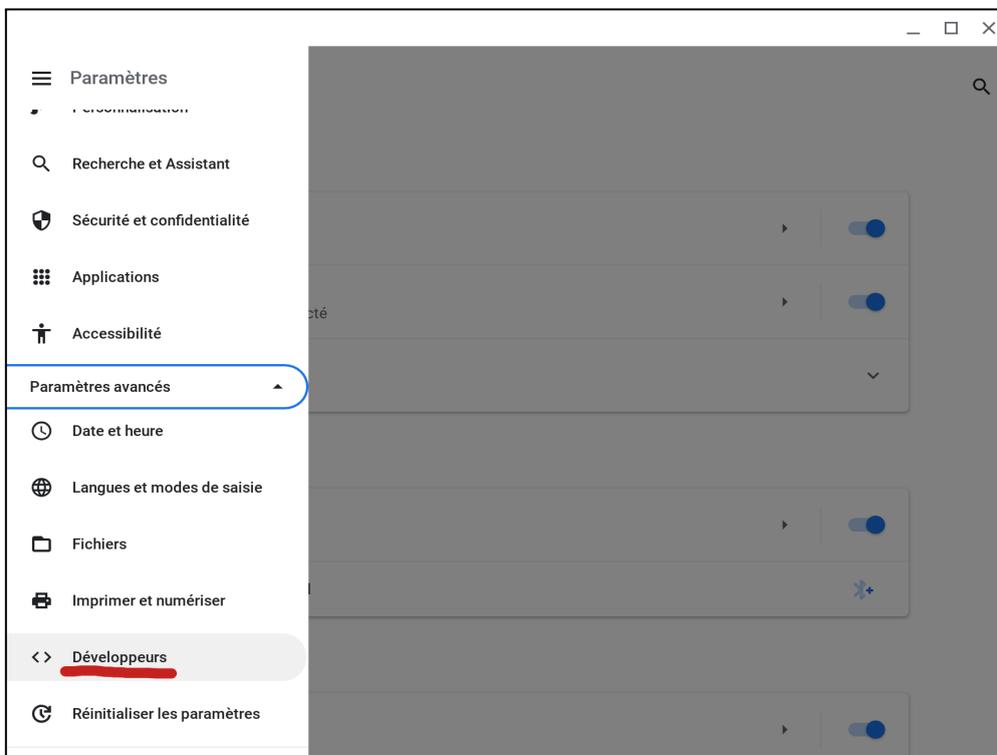
Le but de ce tutoriel est de vous montrer comment installer puis paramétrer la machine virtuelle linux pour qu'elle fonctionne dans la même langue que celle définie pour Chrome OS puis ajouter un outil permettant d'installer des logiciels en quelques clics de souris sans avoir besoin de connaissances pointues des systèmes Linux et enfin d'installer un serveur d'impression de manière à pouvoir imprimer depuis linux.

1 - Installation de linux :

Cliquez sur l'horloge située dans la barre des tâches puis sur la roue dentée en haut à droite pour entrer dans les paramètres de Chrome OS.



Dans la fenêtre qui s'affiche cliquez sur l'icône représentée par trois lignes horizontales situées en haut à gauche de la fenêtre puis sur **Paramètres avancés** enfin, cliquez sur la rubrique **Développeurs**.



Dans la rubrique **Développeurs / Environnement de développement Linux** cliquez sur le bouton **Activer**.

Vous obtenez une fenêtre qui permet de définir le nom d'utilisateur et la taille du disque qui sera dédié à linux. Par défaut c'est le nom de votre compte gmail qui est utilisé et 10 Go d'espace disque qui sont réservés, c'est suffisant dans un premier temps. Vous pourrez modifier cette valeur ultérieurement si le besoin s'en faisait sentir.

Remarque : Aucun mot de passe n'est défini, c'est le mot de passe saisi lors de la connexion à Chrome OS qui sert de mot de passe de connexion à la session Linux, l'utilisateur est donc considéré comme l'administrateur du système. Par conséquent toutes les opérations qui nécessitent habituellement la saisie d'un mot de passe se feront sans qu'aucun mot de passe ne soit demandé.

Configurer l'environnement de développement Linux

Modifier le fonctionnement des applications Linux.

Nom d'utilisateur

demo

Taille du disque

Vous pourrez modifier ce réglage ultérieurement dans les paramètres

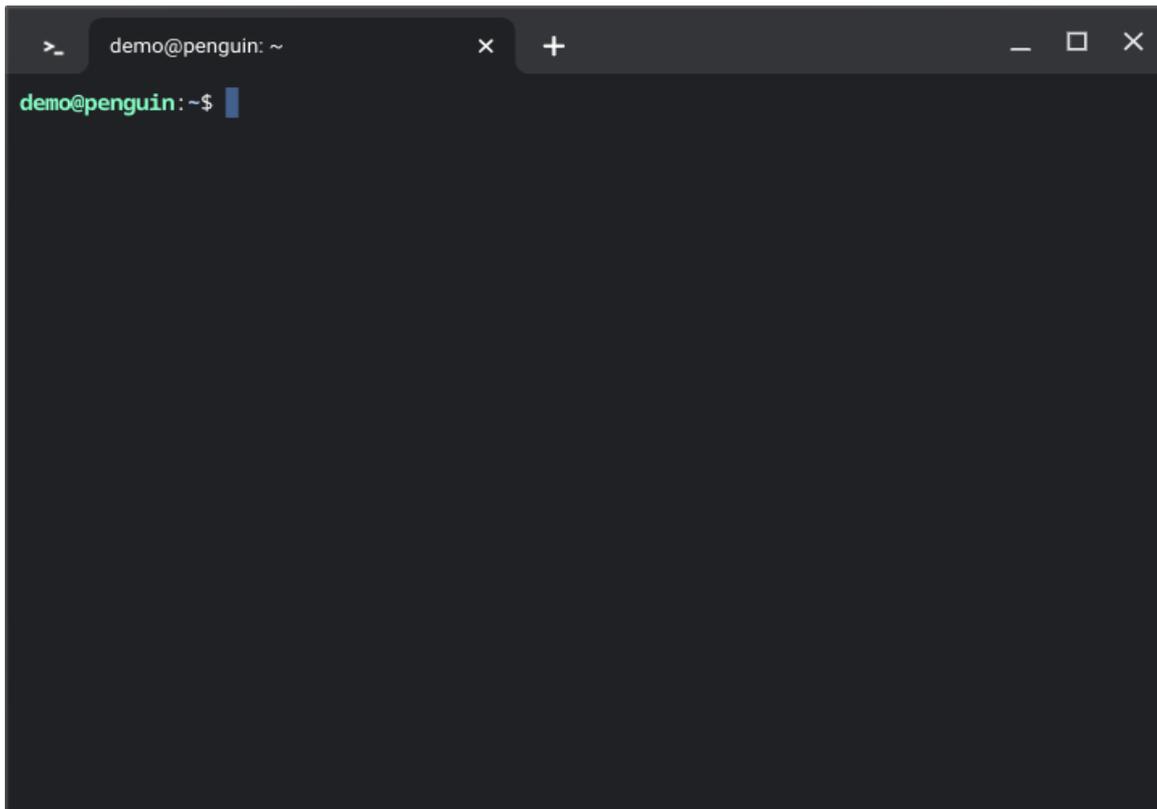
Recommandée (10,0 Go)

Personnalisé

Retour Installer

Le téléchargement et l'installation ne prennent que quelques minutes (seulement 450 Mo à télécharger).

Aussitôt l'installation terminée, la machine virtuelle linux démarre automatiquement et un terminal s'affiche:



Vous pouvez maintenant taper des commandes dans le terminal. C'est l'unique application qui est actuellement installée.

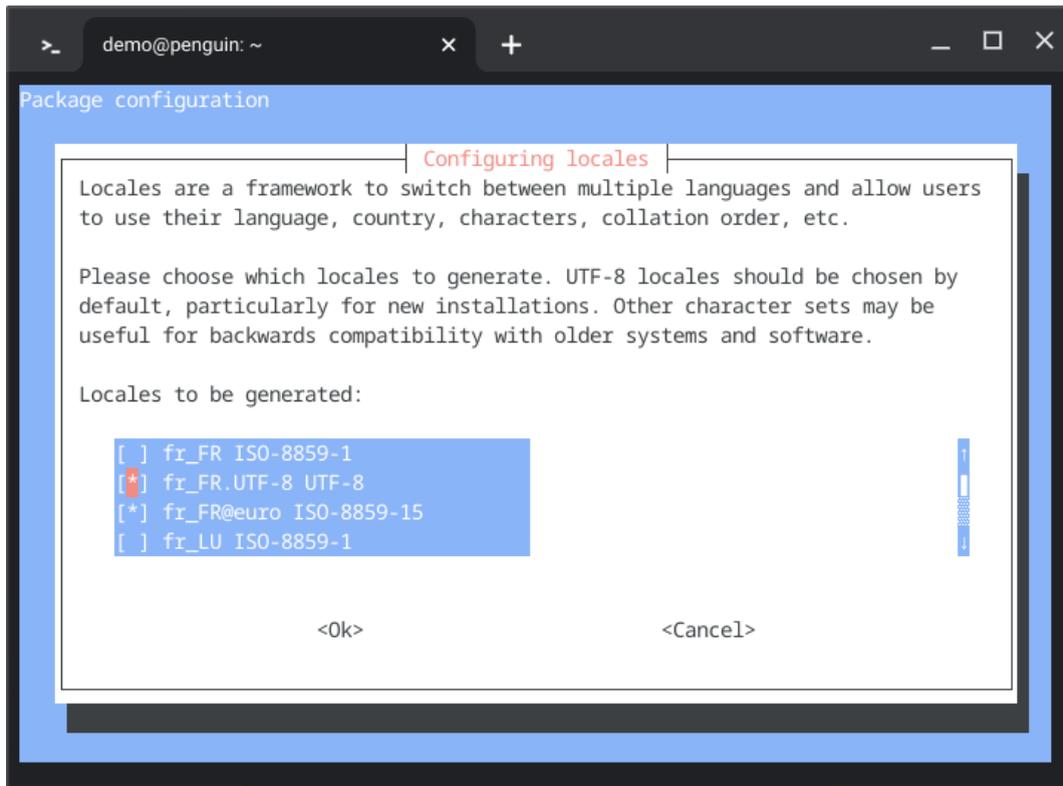
2 - Configuration des locales :

En l'état, la machine virtuelle linux est localisée aux USA, ce qui signifie que même si vous installez un logiciel traduit en français, les menus et les messages de cette application s'afficheront systématiquement en anglais. On va indiquer à linux que l'on se trouve en France et que c'est le français qui doit être utilisé.

Tapez la commande suivante dans le terminal :

sudo dpkg-reconfigure locales

La fenêtre suivante s'affiche en mode texte :

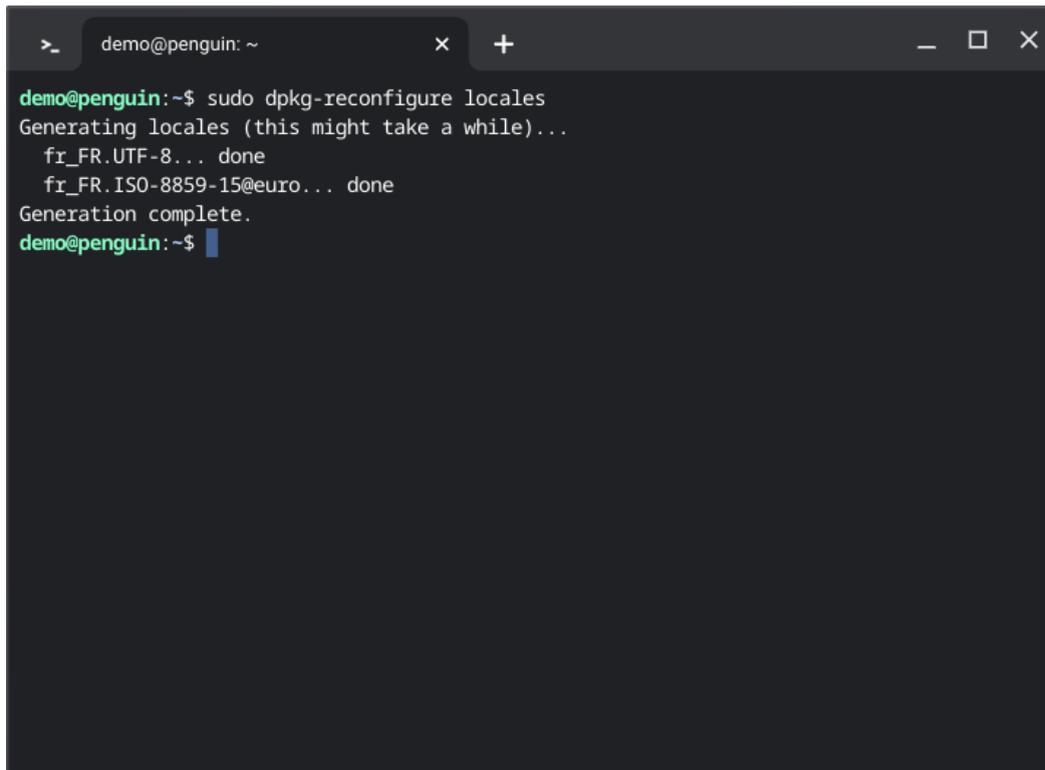


C'est là que vous pouvez sélectionner les locales que vous souhaitez installer sur le système. Toutes les lignes cochées avec une étoile sont sélectionnées. Déplacez vous dans la liste avec les touches fléchées haut et bas, utilisez la touche espace pour sélectionner/désélectionner une locale, les touches fléchées gauche et droite permettent de choisir entre **Ok** et **Cancel**.

Vous devez décocher la ligne **en_US UTF8** puis sélectionner au minimum la ligne **fr_FR UTF8**.

J'ai également coché la ligne **fr_FR@euro-8859-15** pour avoir la compatibilité avec le symbole euro mais je ne suis pas certain que ce soit indispensable.

Il ne reste plus qu'à appuyer sur la touche Entrée pour valider les choix. Une fenêtre récapitulative s'affiche avant la génération des locales qui ne prend que quelques secondes et lorsque c'est terminé, ce message s'affiche :

A screenshot of a terminal window titled 'demo@penguin: ~'. The terminal shows the command 'sudo dpkg-reconfigure locales' being executed. The output indicates that locales are being generated, with 'fr_FR.UTF-8' and 'fr_FR.ISO-8859-15@euro' being processed successfully. The terminal ends with the prompt 'demo@penguin:~\$' and a cursor.

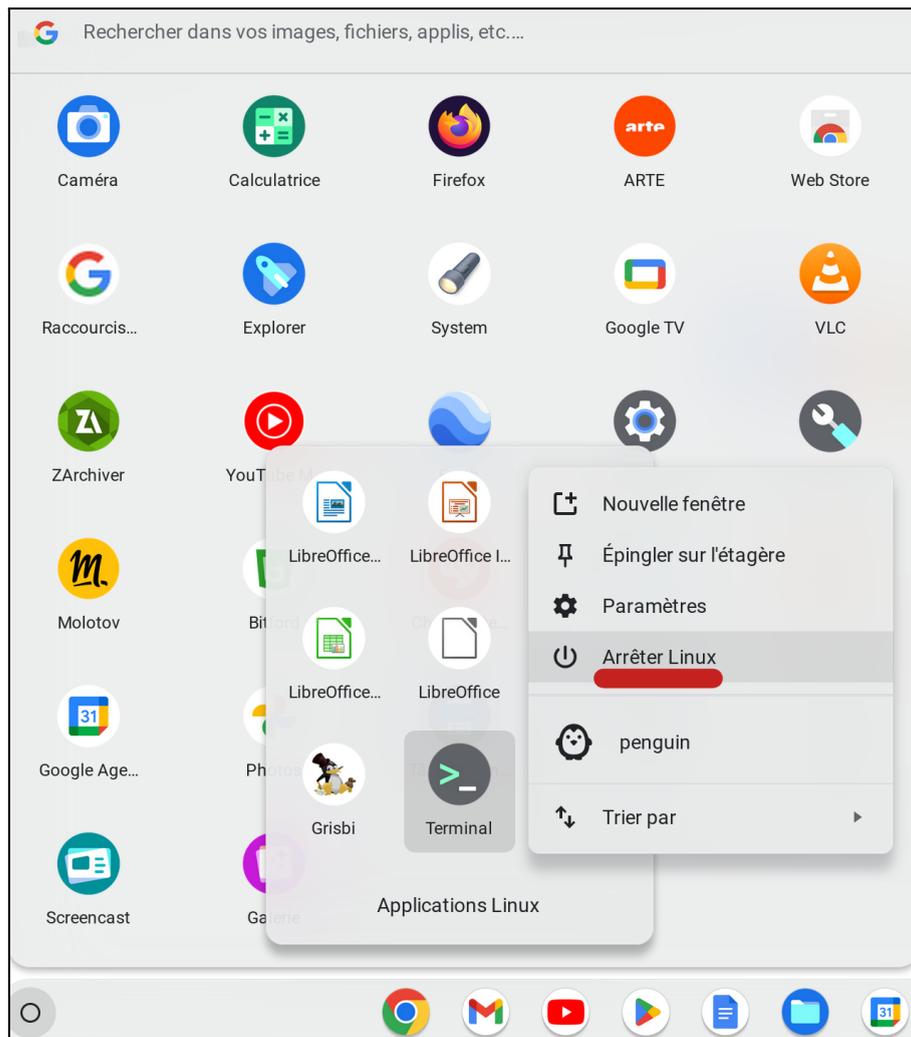
```
demo@penguin:~$ sudo dpkg-reconfigure locales
Generating locales (this might take a while)...
 fr_FR.UTF-8... done
 fr_FR.ISO-8859-15@euro... done
Generation complete.
demo@penguin:~$
```

Pour que la nouvelle localisation soit prise en compte par le système, il faut redémarrer Linux.

Vous avez 2 solutions pour ça :

- 1 - Arrêter le chromebook et le redémarrer. (utilisation du bouton d'arrêt du chromebook, c'est l'option la plus simple et la plus radicale)
- 2 - Arrêter seulement la machine virtuelle linux puis la relancer. Pour ça, cliquez sur l'icône marquée d'un cercle à gauche de la barre des tâches pour afficher le menu des applications. L'icône du terminal se trouve tout en bas de la liste dans un groupe nommé **Applications linux**.

Les icônes de toutes les applications que nous installerons par la suite se retrouveront dans ce groupe. Sur la copie d'écran ci-dessous on voit que libreoffice et grisbi sont déjà installés.

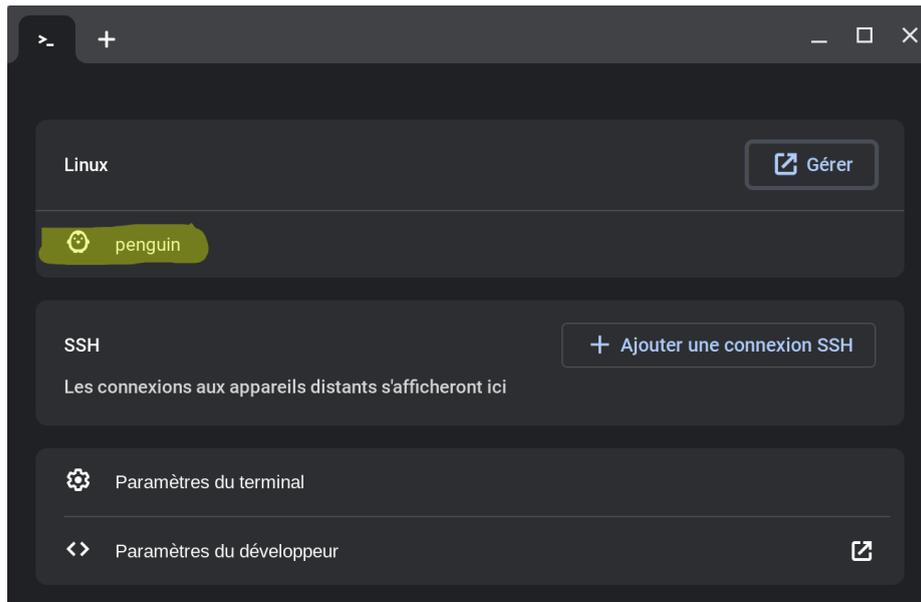


*Remarque : Chez moi la commande **sudo reboot** habituellement utilisée pour redémarrer un système Linux n'a pas fonctionné.*

Faites un clic droit sur l'icône du Terminal ou une tape à 2 doigts sur le pavé tactile si vous n'avez pas de souris afin d'afficher son menu contextuel et cliquez sur **Arrêter Linux**

Pour démarrer à nouveau linux, cliquez sur l'icône du Terminal.

Cette fenêtre s'affiche d'abord :



Le bouton **Gérer** renvoie dans les paramètres Linux de Chrome OS.
La ligne **penguin** permet de lancer le terminal que nous avons quitté précédemment.
On peut ajouter une connexion SSH en cliquant sur le bouton dédié.
L'option **Paramètres du terminal** permet de changer le comportement et les paramètres d'affichage du terminal.
Paramètres du développeur renvoie également dans les paramètres linux de Chrome OS.

C'est sur la ligne **penguin** qu'il faut cliquer pour ouvrir le Terminal. La progression du processus de démarrage de la machine virtuelle s'affiche dans la fenêtre du terminal. Il faut compter une dizaine de secondes dépendant de la puissance du processeur. Si par la suite vous êtes amenés à démarrer d'autres applications Linux, le chargement sera rapide puisque la machine virtuelle sera déjà lancée.

3 - Installation de Gnome Logiciels :

Gnome Logiciels est une logithèque qui permet de choisir, installer, désinstaller des logiciels et faire les mises à jour très facilement.

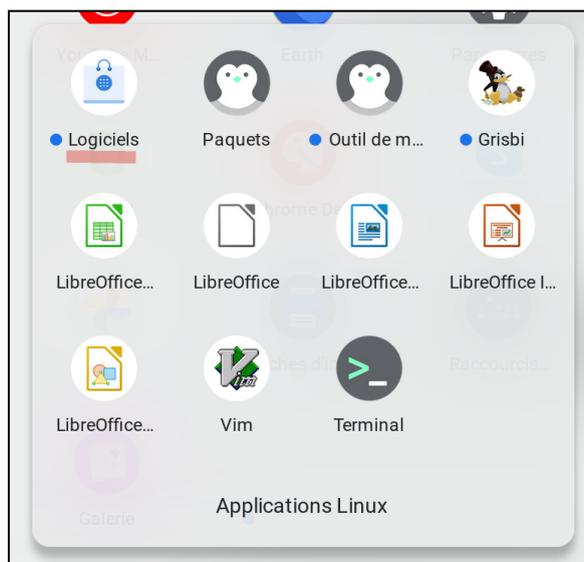
Dans le terminal tapez la commande suivante:
sudo apt install gnome-software
puis appuyez sur la touche entrée

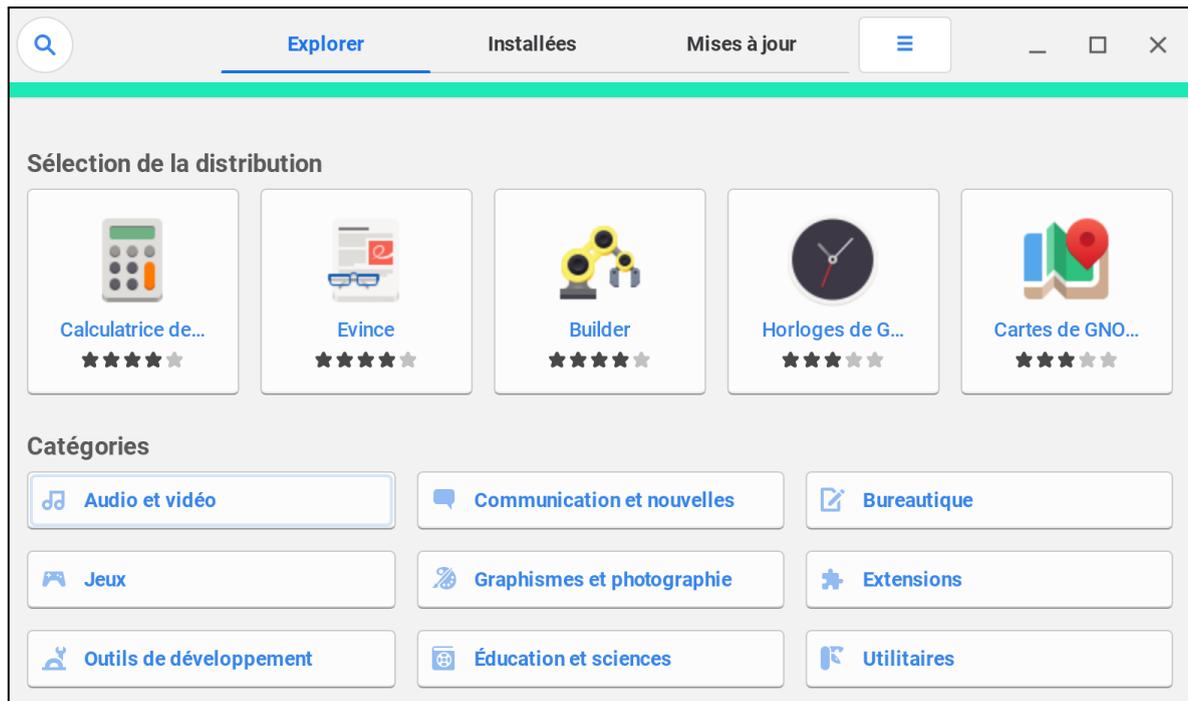
```
>_ Terminal x + - □ ×
@penguin:~$ sudo apt install gnome-software
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
 bolt enchant-2 fwupd fwupd-amd64-signed gnome-software-common
 libappstream-glib8 libarchive13 libefiboot1 libefivar1 libenchanted-2-2
 libflashrom1 libftdi1-2 libfwupd2 libfwupdplugin1 libgcab-1.0-0 libgspell-1-2
 libgspell-1-common libjcat1 libmalcontent-0-0 libsmbios-c2
 libtss2-esys-3.0.2-0 libtss2-mu0 libtss2-sys1 libtss2-tcti-cmd0
 libtss2-tcti-device0 libtss2-tcti-mssim0 libtss2-tcti-swtpm0 libxmlb1 tpm-udev
Paquets suggérés :
 gir1.2-fwupd-2.0 apt-config-icons-hidpi gnome-software-plugin-flatpak
 gnome-software-plugin-snap lrzip libenchanted-2-voikko libsmbios-doc
Paquets recommandés :
 secureboot-db
Les NOUVEAUX paquets suivants seront installés :
 bolt enchant-2 fwupd fwupd-amd64-signed gnome-software gnome-software-common
 libappstream-glib8 libarchive13 libefiboot1 libefivar1 libenchanted-2-2
 libflashrom1 libftdi1-2 libfwupd2 libfwupdplugin1 libgcab-1.0-0 libgspell-1-2
 libgspell-1-common libjcat1 libmalcontent-0-0 libsmbios-c2
 libtss2-esys-3.0.2-0 libtss2-mu0 libtss2-sys1 libtss2-tcti-cmd0
 libtss2-tcti-device0 libtss2-tcti-mssim0 libtss2-tcti-swtpm0 libxmlb1 tpm-udev
 0 mis à jour, 30 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 11,1 Mo dans les archives.
Après cette opération, 30,3 Mo d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [0/n]
```

Tapez “O” et attendez que le terminal vous rende la main en affichant l’invite de commande. Vous trouverez l’icône nommée **Logiciels** dans le menu principal en cliquant sur l’icône **Applications Linux**.

Cliquez sur l’icône pour lancer le programme.

Remarque : Si vous faites un clic droit sur cette icône pour afficher son menu contextuel vous verrez qu’il y a une option pour ajouter l’icône dans la zone de lancement rapide de la barre des tâches.





Diverses options sont disponibles :

Explorer permet de naviguer dans les différentes catégories de logiciels.

Installées permet de gérer toutes les applications installées.

Mises à jour affiche la liste des mises à jour disponibles.

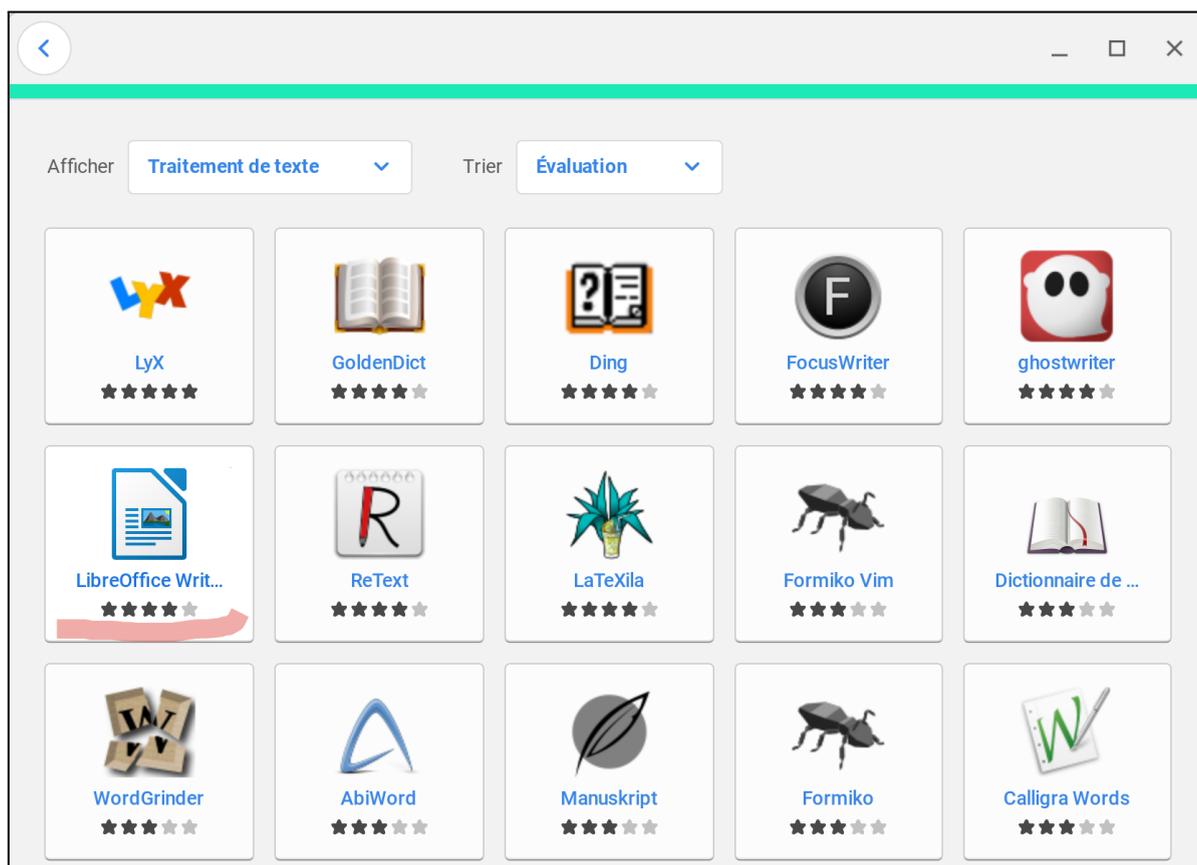
La loupe permet de rechercher un logiciel.

Remarque : seules les applications compatibles Wayland s'affichent ici.

Exemple d'installation d'un logiciel : LibreOffice Writer

Nous allons maintenant installer LibreOffice Writer :

Cliquez sur la rubrique Bureautique puis dans **Afficher** sélectionnez **Traitement de texte**.



En cliquant sur l'icône LibreOffice Writer vous obtenez des informations sur l'application. Cliquez simplement sur le bouton **Installer** et patientez jusqu'à la fin de l'installation. Lorsque c'est terminé, les boutons **Ouvrir** et **Supprimer** s'affichent. Vous pouvez maintenant fermer Logiciels.

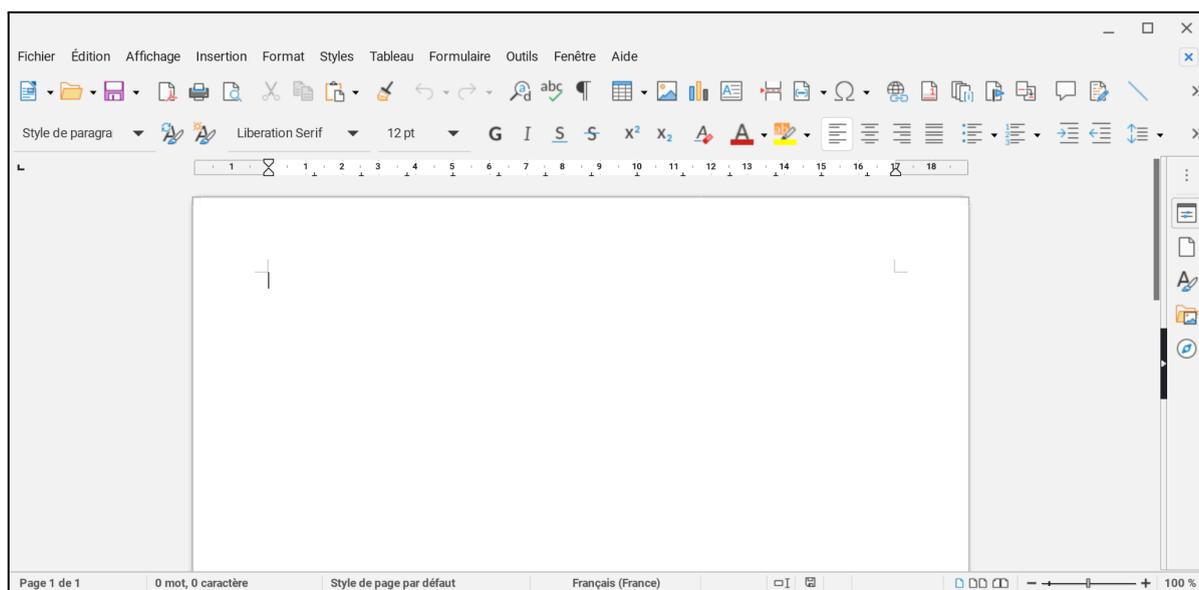
Remarque importante : si vous lancez Writer maintenant, les menus s'afficheront en anglais car les traductions se trouvent dans un paquet séparé qui n'a pas été installé automatiquement par la logithèque. De plus, l'affichage de la barre de menu est plutôt moche. Nous allons arranger ça en installant 2 paquets supplémentaires avec une ligne de commande.

Ouvrez le Terminal et tapez la ligne suivante :

sudo apt install libreoffice-l10n-fr libreoffice-gtk3

Vous pouvez utiliser la fonction copié/collé en sélectionnant le texte dans cette page puis clic droit/Copier ensuite un clic droit dans la fenêtre du terminal collera le texte copié.

Au prochain démarrage LibreOffice Writer ressemblera à ceci :



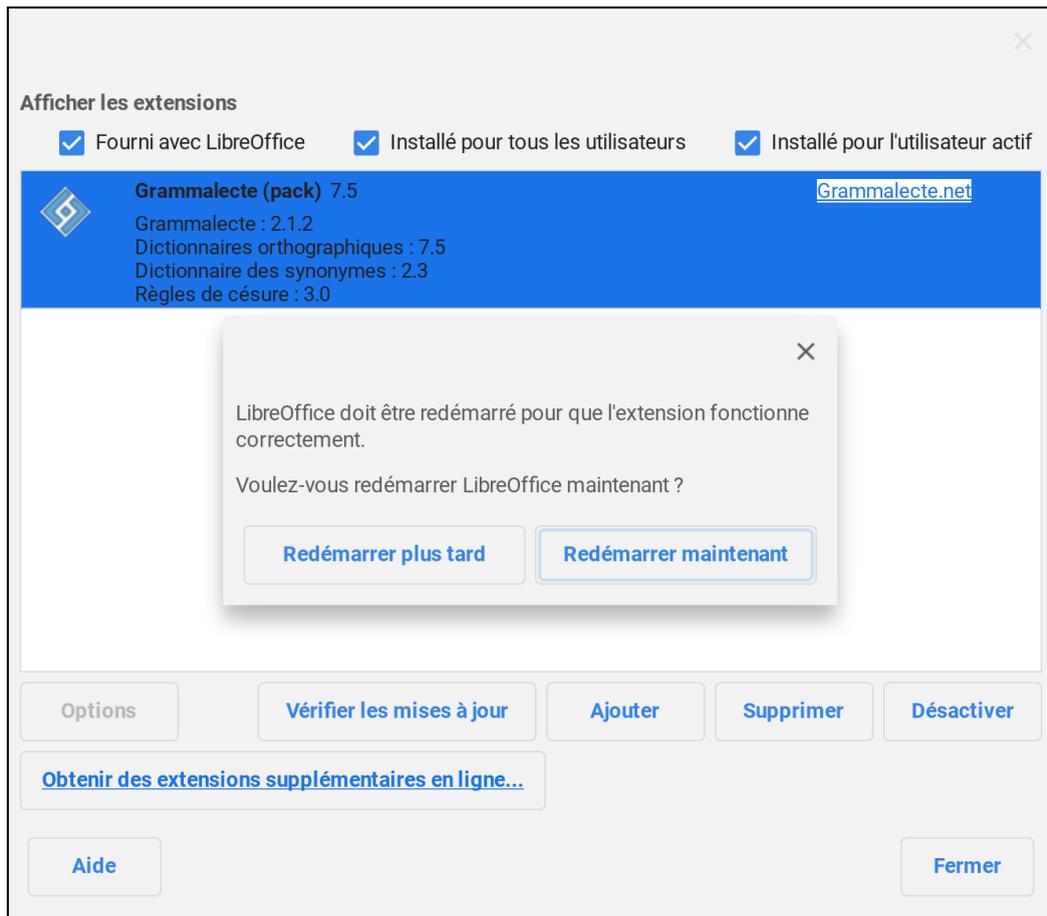
Vous pouvez compléter l'installation en ajoutant l'extension de correction orthographique et grammaticale **Grammalecte** qui installera les dictionnaires automatiquement.

Cette extension est disponible ici :

<https://grammalecte.net/>

Cliquez sur Télécharger puis sélectionnez Grammalecte pour LibreOffice 5.3+

Le fichier **Grammalecte-fr-v2.1.2.oxt** sera téléchargé dans le dossier Téléchargements, il suffira d'un double clic sur le nom du fichier pour installer l'extension dans LibreOffice.



4 - Installation de cups

Vous allez probablement vouloir imprimer un document depuis Writer ou tout autre logiciel linux.

Si vous ouvrez maintenant la boîte de dialogue d'impression, vous constaterez que votre imprimante bien que fonctionnelle sous Chrome OS et actuellement allumée n'apparaît pas dans la liste des imprimantes disponibles. Il y a seulement **Generic Printer** et **Imprimer dans un fichier**. C'est tout à fait normal car le serveur d'impression **cups** n'est pas installé (cups signifie **C**ommon **U**nix **P**rinting **S**ystem).

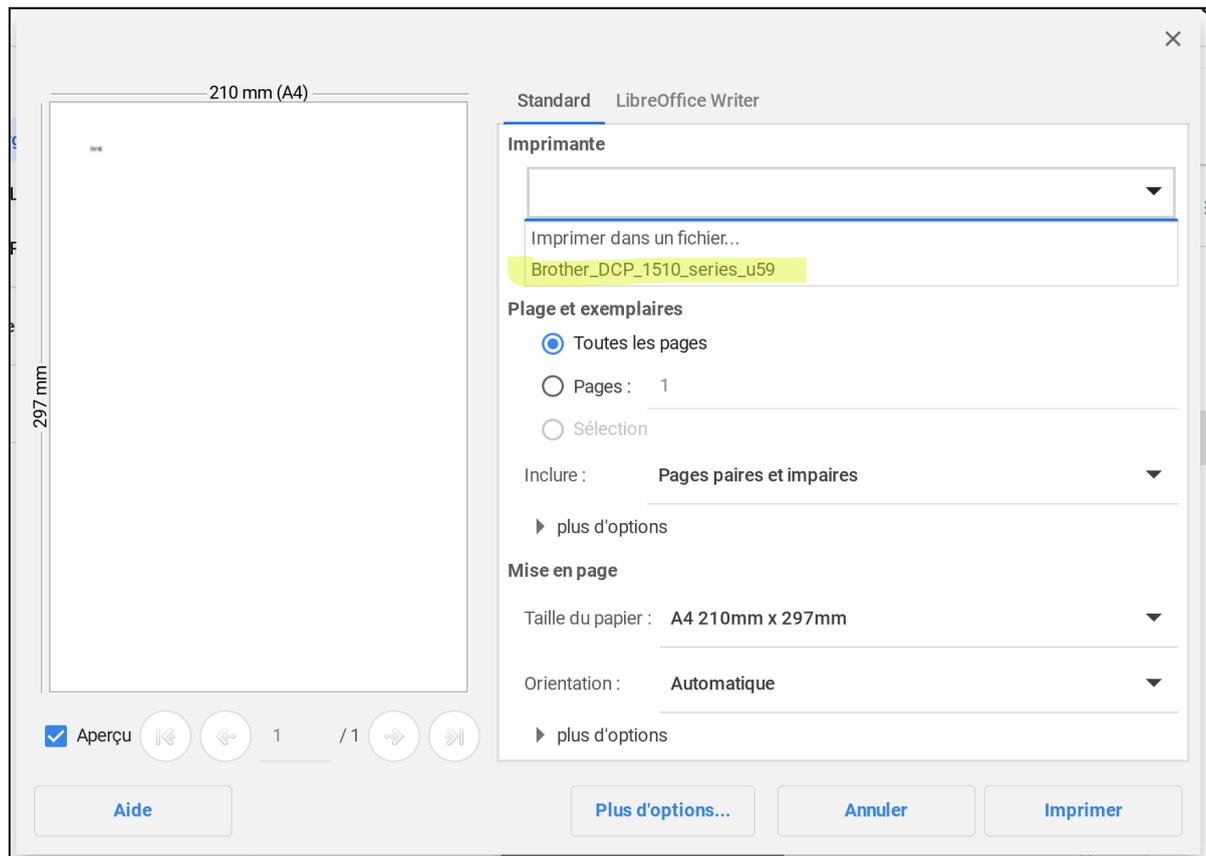
Cups ne peut pas être installé depuis la logithèque car il n'apparaît pas dans la liste, nous allons devoir l'installer via une ligne de commande.

Ouvrez le terminal linux et tapez la commande suivante

sudo apt install cups

et attendez que l'installation arrive à son terme pour redémarrer linux afin que le service cups soit lancé automatiquement.

Ouvrez à nouveau la boîte de dialogue d'impression et dans le cadre **Imprimante** cliquez sur la petite flèche vers le bas pour dérouler la liste et vous verrez votre imprimante apparaître, sélectionnez-la :

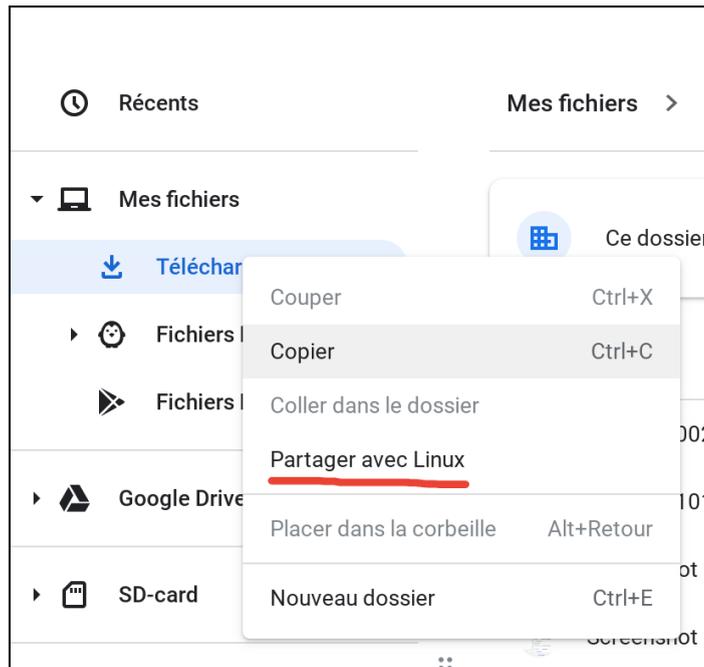


Vous pouvez imprimer maintenant.

5 - Partager des fichiers entre Chrome OS et Linux :

Les fichiers linux sont sauvegardés par défaut dans le dossier personnel de l'utilisateur. Les fichiers téléchargés sous Chrome OS sont placés systématiquement dans le dossier **Téléchargements**. Ces 2 dossiers sont placés à des endroits différents.

Chrome OS peut voir vos fichiers linux qui s'affichent dans le dossier **Fichiers linux** mais par défaut, linux lui, ne peut pas voir vos fichiers du dossier Téléchargements de Chrome OS. Pour permettre à linux d'accéder à ce dossier, il faut déclarer un partage de fichiers. Pour ça, ouvrez le navigateur de fichiers et faites un clic droit sur le nom du dossier que vous souhaitez partager afin d'afficher le menu contextuel et cliquez sur **Partager avec Linux**.



Le contenu du dossier Téléchargements de Chrome OS est accessible depuis linux en **/mnt/chromeos/MyFiles/Downloads** ! pas très pratique tout ça. On va simplifier en créant un lien symbolique qu'on nommera Téléchargements dans votre répertoire personnel. Ce lien pointerait vers le dossier Downloads. Pour ça ouvrez le terminal et tapez la commande suivante :

sudo ln -s /mnt/chromeos/MyFiles/Downloads Téléchargements

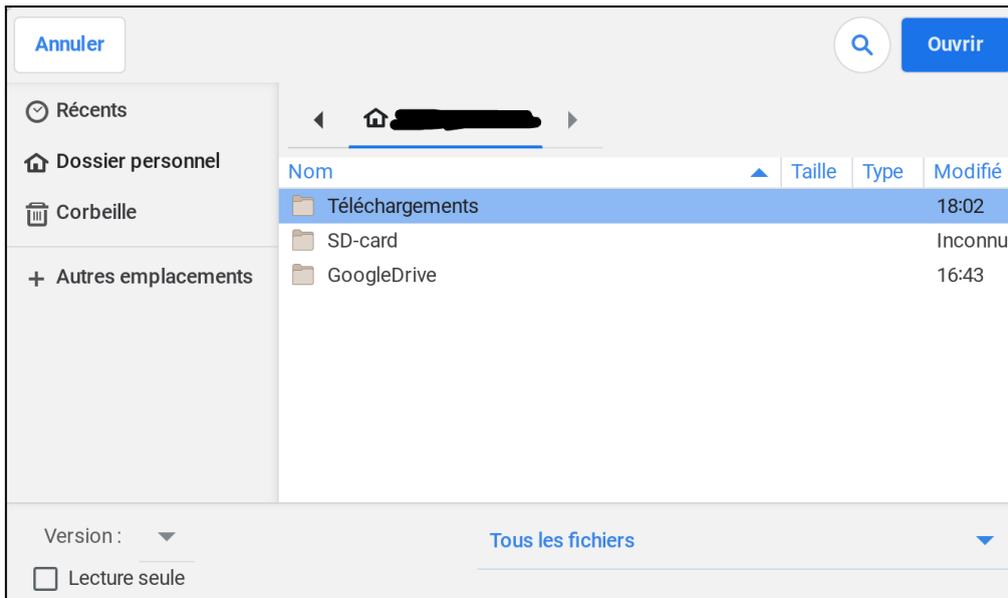
Maintenant depuis linux, vous avez un accès direct au dossier Téléchargements.

De même, pour créer un lien symbolique vers la carte SD, déclarez d'abord le partage dans le navigateur de fichiers puis, pour obtenir l'accès à SD-card, tapez :

sudo ln -s /mnt/chromeos/removable/SD-card SD-card

On peut même faire la même chose avec le Drive : clic droit sur Google Drive / Partager puis **sudo ln -s /mnt/chromeos/GoogleDrive/MyDrive GoogleDrive**

Lorsque vous cliquez sur **Ouvrir** dans Writer vous obtenez maintenant :



C'est tout de même plus pratique !

Maintenant que nous avons apporté beaucoup de modifications au système initial, nous allons le nettoyer afin qu'il utilise le moins d'espace de stockage possible.

Dans le terminal tapez successivement les 2 commandes suivantes :

sudo apt autoremove

Cette commande supprime les paquets devenus inutiles suite à des mises à jour, des installations ou des suppressions de logiciels.

sudo apt clean

Cette commande efface le cache de téléchargement de APT.

6 - Quel est l'impact de l'installation de linux sur l'utilisation de la ram et de l'espace disque ?

La commande **free** permet de connaître l'utilisation de la mémoire sur un système linux.

Voici le résultat de **free -m** tapée dans le terminal linux aussitôt le démarrage de la machine virtuelle linux : (les valeurs sont exprimées en Mb)

| | total | utilisé | libre | partagé | tamp/cache | disponible |
|----------------------|-------|---------|-------|---------|------------|------------|
| Mem: | 2763 | 108 | 2184 | 2 | 469 | 2654 |
| Partition d'échange: | | 0 | 0 | 0 | | |

On voit que linux a le droit d'utiliser au maximum 2763 Mb de ram sur les 4 Go disponibles sur mon Chromebook.

108 Mb seulement sont effectivement utilisés. Linux utilise donc très peu de mémoire pour fonctionner, ça s'explique par le fait qu'il n'y a aucune interface graphique installée dans la machine virtuelle. C'est Chrome OS qui gère la partie affichage graphique de Linux.

469 Mb sont mis en cache, cet espace contient des données susceptibles de servir mais qui peuvent être supprimées à tout moment en cas de besoin.

Voyons maintenant l'utilisation générale de la mémoire du chromebook (Chrome OS + Linux tournant simultanément)

J'ai lancé plusieurs fois la commande **free -m** dans **crosh** (c'est le nom du terminal de Chrome OS) :

```
crosh> free -m
      total        used        free      shared  buff/cache   available
Mem:   3789         862         811        1033        2116        1463
Swap:  7579           0        7579
crosh> free -m
      total        used        free      shared  buff/cache   available
Mem:   3789         806         301        1968        2681         595
Swap:  7579         792        6786
crosh> free -m
      total        used        free      shared  buff/cache   available
Mem:   3789         828         209        2298        2751         243
Swap:  7579         883        6695
crosh> free -m
      total        used        free      shared  buff/cache   available
Mem:   3789         922         266        2212        2601         240
Swap:  7579        1128        6451
crosh>
```

La première fois Chrome OS vient d'être lancé, le système est au repos avec seulement le terminal Crosh affiché.

On voit que 862 Mb de ram sont utilisés. Le swap n'est pas utilisé. Mais déjà plus de 2 Go mis en cache !

La seconde fois, j'ai lancé en plus la machine virtuelle linux.

On voit qu'il y a peu d'impact sur la quantité de ram utilisée, une bonne partie des éléments en mémoire ont été soit transférés sur le swap soit mis en cache.

La troisième fois, j'ai lancé LibreOffice Writer :

Toujours peu d'impact sur la ram utilisée, légère augmentation de la taille du swap. On voit que LibreOffice n'est pas vraiment gourmand en mémoire.

Et enfin la quatrième fois, j'ai lancé en plus Google Chrome :

Légère augmentation de l'usage de la ram, il reste toujours plus de 6 Go d'espace disponible dans le swap et plus de 2 Go de données au total ont été mises en cache. Il y a donc encore pas mal de marge avant de saturer la mémoire du Chromebook au point de le bloquer !

En cours de fonctionnement, la taille du cache augmente rapidement jusqu'à utiliser la presque totalité de la ram disponible. Chrome OS a recours à une fonctionnalité spécifique

au noyau linux appelée **zRam** qui lui permet de compresser des données en mémoire afin d'accélérer son fonctionnement en diminuant le nombre d'accès physiques au swap.

Lors de l'écriture de ce tutoriel, j'ai eu à certains moments une bonne dizaine de fenêtres appartenant à divers logiciels Chrome OS et Linux ouvertes simultanément. À aucun moment je n'ai constaté de ralentissement. La seule latence constatée est lors du démarrage d'une première application linux depuis le menu principal. Le lancement de la machine virtuelle prend une dizaine de secondes avant de commencer à charger l'application et rien ne s'affiche à l'écran pour montrer qu'elle est en cours de démarrage, on peut être tenté de cliquer à nouveau sur l'icône de l'application en pensant qu'elle n'a pas démarré. Par contre si on démarre cette même application depuis une icône placée dans la barre de lancement rapide, l'icône s'anime montrant le démarrage en cours.

En ce qui concerne l'utilisation de l'espace de stockage maintenant : il me reste 30 Go d'espace libre sur un total initial de 64 Go mais j'ai pas mal d'applications installées.

J'estime que 64 Go d'espace de stockage est le minimum requis pour installer et utiliser des applications Linux dans de bonnes conditions sur un Chromebook et l'idéal est d'avoir 8 Go de ram installée. J'ai tenté l'installation sur un vieux chromebook avec 4 Go de ram et seulement 32 Go d'espace disque, l'installation a réussi mais j'ai relevé des dysfonctionnements en cours d'utilisation.

Matériel utilisé : Samsung Galaxy Chromebook2 360, processeur Celeron N4500, 4 Go de ram et 64 Go d'espace de stockage.

Chrome OS version 116.0.5845.168 (Build officiel) (64 bits).