Installation et premières configurations du système GNU GuixSD

- Objet : Installer et commencer à configurer GuixSD
- Niveau requis :
  avisé
- Commentaires :
  https://www.gnu.org/software/guix/manual/html_node/GNU-Free-Documentation-License.html
- Débutant, à savoir : Utiliser GNU/Linux en ligne de commande, tout commence là !.

Introduction

Comment j'ai installé GuixSD 0.14 sur un HP ProBook 6460b (nécessitant un adaptateur USB sans fil pour GNU/Linux-libre)

Edit : Pour l'installation de la version **0.16 en Dual-boot avec Debian**, c'est par ici : **https://debian-facile.org/utilisateurs:gonzoleri:tutos:guixsd-0.16-en-dual-boot-avec-debian-testing-sid**

Prérequis :

1. Il est très hautement recommandé de télécharger et graver **Superdisk grub2** **https://www.supergrubdisk.org/category/download/supergrub2diskdownload**. Il est possible que vous en ayez besoin lors du redémarrage. C'est ce qui s'est produit chez moi. Heureusement, je l'avais déjà; il m'a permis de finaliser avec succès l'ensemble de la procédure.
2. Pour le **Wi-Fi** avoir un adaptateur USB sans fil pour GNU/Linux-libre : **https://tehnoetic.com/adapters** ou bien du matériel Wi-Fi ne nécessitant pas de pilote propriétaire. (Il m'a fallu utiliser la connexion filaire pour l'installation).
3. Savoir que pour les partitionnements **EFI** et **GPT**, vous devrez suivre les liens indiqués plus bas à l'étape du partitionnement (Ces options n'étant pas adaptées à mon cas de figure).

Installation

J'ai ensuite téléchargé **GuixSD**, la distribution de logiciels GNU Guix :

https://www.gnu.org/software/guix/download/

Démarrage à partir d'un DVD gravé avec '**guixsd-install-0.14.0.x86_64-linux.iso**' Sources du manuel :

https://www.gnu.org/software/guix/manual/html_node/Preparing-for-Installation.html#Preparing-for-Installation

L'image d'installation utilise la disposition du clavier qwerty. Si je veux le changer, je peux utiliser la commande **loadkeys**. Par exemple, la commande suivante sélectionne la disposition du clavier **fr**,

donc allons-y :

```
# loadkeys fr
```

Pour avoir le Wi-Fi, voir ci-dessous :

https://www.gnu.org/software/guix/manual/html_node/Preparing-for-Installation.html#Networking

Ma carte wifi nécessite l'utilisation d'un firmware non libre (Broadcom Limited BCM4313 802.11bgn Wireless Network Adapter).

Je décide de commander une carte wifi **USB Adaptateur USB sans fil Mini Technoethical N150 pour GNU/Linux-libre** à l'adresse :

https://tehnoetic.com/adapters/tehnoetic-wireless-adapter-gnu-linux-libre-tet-n150

En attendant, par le biais de mon joli câble rj45, j'exécute la commande suivante pour voir comment mes interfaces réseau sont nommées :

```
# ifconfig -a
```

Les interfaces filaires ont un nom commençant par '**e**'; Par exemple, l'interface correspondant au premier contrôleur Ethernet intégré est appelée «**eno1**».

```
# ifconfig enp0s25 up
```

À ce stade, je dois faire l'acquisition d'une adresse IP. Les adresses IP étant automatiquement attribuées via DHCP, je peux exécuter:

```
# dhclient -v enp0s25
```

Essayons d'envoyer une requête ping à un serveur pour voir si la mise en réseau est opérationnelle : (sachant que j'habite dans une zone non dégroupée, vitesse 2.0 Mbs :/)

```
# ping -c 3 gnu.org
PING gnu.org (208.118.235.148): 56 data bytes
64 bytes from 208.118.235.148: icmp_seq=0 ttl=51 time=138.917 ms
64 bytes from 208.118.235.148: icmp_seq=1 ttl=51 time=139.388 ms
64 bytes from 208.118.235.148: icmp_seq=2 ttl=51 time=138.998 ms
--- gnu.org ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 138.917/139.101/139.388/0.206 ms
```

Partitionnement GPT ou **EFI** : Les tests que j'ai effectués dans les machines virtuelles n'ayant pas été très concluants, je n'en sais pas trop sur le sujet. Si vous souhaitez utiliser GPT ou EFI, veuillez vous reporter à la documentation :

https://www.gnu.org/software/guix/manual/html_node/Preparing-for-Installation.html#Disk-Partitioning

Dans mon cas, sachant que je n'aurai pas à m'inquiéter de cette question pour cause d'ordinateur datant de 2011, c'est parti :

```
# cfdisk

Device     Boot     Start     End         Sectors    Size   Id Type
/dev/sda1    *       2048    479995903   479993856  228.9G  83 Linux
/dev/sda2       479995904    488397167     8401264      4G  82 Linux swap
```

De préférence, attribuez une étiquette aux systèmes de fichiers afin de pouvoir les consulter facilement et de manière fiable dans les déclarations de système de fichiers (voir Systèmes de fichiers). Cela se fait généralement à l'aide de l'option **-L** de **mkfs.ext4** et des commandes associées. En supposant que la partition racine cible se trouve dans **/dev/sda1**, vous pouvez créer un système de fichiers portant le libellé **my-root** avec la commande suivante, personnellement, j'ai fait :

```
# mkfs.ext4 -L my-root /dev/sda1
```

Monter le système de fichiers cible sous **/mnt** avec cette commande (encore une fois, en supposant que **my-root** est l'étiquette du système de fichiers racine) :

```
# mount LABEL=my-root /mnt
```

Ma partition de swap étant **/dev/sda2**, j'ai exécuté :

```
# mkswap /dev/sda2
# swapon /dev/sda2
```

Procédons à l'installation

Avec les partitions cibles prêtes et la racine cible montée sur **/mnt**, on est bon. Premièrement :

```
# herd start cow-store /mnt
```

Cela crée une copie en écriture de **/gnu/store**, de telle sorte que les packages qui lui sont ajoutés au cours de la phase d'installation soient écrits sur le disque cible sur **/mnt** plutôt que conservés en mémoire. Cela est nécessaire car la première phase de la commande **guix system init** implique le téléchargement ou la compilation dans **/gnu/store**, qui est initialement un système de fichiers en mémoire.

Création du répertoire cible de configuration du système :

```
# mkdir /mnt/etc
```

Création du prochain fichier de configuration (vide pour l'instant) :

```
# touch /etc/configuration/perso.scm
```

Remarque : vous pouvez également éditer un fichier existant déjà dans **/etc/configuration** (par exemple, "**desktop.scm**" ou "**bare-bones.scm**" et le modifier à votre guise).

Pour adapter la configuration de l'ordinateur à mes besoins, j'ai utilisé **vi** , ne connaissant que celui-ci. Nano et Zile sont toutefois disponibles.

```
# vi /etc/configuration/perso.scm
```

perso.scm

```
;; This is an operating system configuration template
;; for a "desktop" setup, without full-blown desktop
;; environments. "host-name" is my own laptop.

(use-modules (gnu) (gnu system nss))
(use-service-modules desktop)
(use-package-modules bootloaders certs ratpoison suckless wm)

(operating-system
  (host-name "lxx-xxx-xx-xx-xxx-xxx-xx6")
  (timezone "Europe/Paris")
  (locale "en_US.utf8")

  ;; Assuming /dev/sda is the target hard disk, and "my-root" is
  ;; the label of the target root file system.
  (bootloader (grub-configuration (target "/dev/sda")))
  (file-systems (cons (file-system
                        (device "my-root")
                        (title 'label)
                        (mount-point "/")
                        (type "ext4"))
                      %base-file-systems))

  (swap-devices '("/dev/sda2"))

  (users (cons (user-account
                (name "hubert")
                (comment "Cathy's brother")
                (group "users")
                (supplementary-groups '("wheel" "netdev"
                                        "audio" "video"))
                (home-directory "/home/hubert"))
               %base-user-accounts))

  ;; Add a bunch of window managers; we can choose one at
  ;; the log-in screen with F1.
  (packages (cons* ratpoison i3-wm i3status dmenu ;window managers
                   nss-certs                      ;for HTTPS access
                   %base-packages))

  ;; Use the "desktop" services, which include the X11
  ;; log-in service, networking with Wicd, and more.
```

```
  (services %desktop-services)

  ;; Allow resolution of '.local' host names with mDNS.
  (name-service-switch %mdns-host-lookup-nss))
```

Faisons maintenant une copie de ce fichier renommé '**config.scm**' sur '**/mnt/etc**'

```
# cp /etc/configuration/perso.scm /mnt/etc/config.scm
```

Une fois la préparation du fichier de configuration terminée, le nouveau système doit être initialisé (rappelez-vous que le système de fichiers racine cible est monté sur **/mnt**) :

```
# guix system init /mnt/etc/config.scm /mnt
```

Ceci copie tous les fichiers nécessaires et installe **GRUB** sur **/dev/sda**, à moins que vous ne passiez l'option **–no-bootloader** . Cette commande peut déclencher des téléchargements ou des versions des packages manquants, ce qui peut prendre un certain temps.

Copie sur /mnt

```
[##########################]
```

L'installation est terminée, je peux redémarrer :

```
# reboot
```

Supergrub2disk \o/

Après le redémarrage, un message comme celui-ci: "**No Operating System installed, please install on hard disk**"

J'ai réussi à démarrer grâce au **superdisk grub2** que j'avais gravé sur un CD **\o/**

https://www.supergrubdisk.org/category/download/supergrub2diskdownload/

Après avoir inséré le CD **superdisk grub2**, j'ai fini par trouver l'existence de cette ligne :

Grub

```
gnu linux-libre
```

J'ai validé et GuixSD a démarré **\(^ o ^)/**

On dirait que **superdisk grub2** a restauré **grub** ...? (J'ai également fait basculer l'indicateur d' amorçage avec **cfdisk** , un détail passé inaperçu lors de l'étape de partitionnement). Ça ressemble à ça maintenant :

```
# cfdisk
```

```
Device     Boot     Start     End         Sectors    Size   Id Type
/dev/sda1    *       2048    479995903   479993856  228.9G  83 Linux
/dev/sda2       479995904    488397167     8401264      4G  82 Linux swap
```

Plus besoin du CD superdisk grub2, le démarrage est maintenant autonome. Je ne sais pas s'il faut remercier **Superdisk grub2** ou **cfdisk** pour la restauration du démarrage, mais l'installation est un succès.

Ctrl-Alt-F2 pour passer en pleine console. l'accès à root est vide (autorisé) Il suffit d'appuyer sur [Entrée]. Après m'être créé un mot de passe utilisateur (moi, l'utilisateur final…)

```
# passwd hubert
New password:
Retype new password:
passwd: password updated successfully
```

…l'utilisateur que je suis peut maintenant se connecter avec mon compte utilisateur.

Choix '**i3**': un bureau minimal. La souris est ok.

Choix '**Ratpoison**' : “Can't logging in …” Pas de souris, rien …

Depuis la console, **Alt-F7** est utilisé pour passer à **X11**.

Ctrl-Alt-F2 et partez à l'exploration!

Un champ de possibilités va enfin être dévoilé ;)

Pour voir la suite sur **xfce**, **GNOME**, **ssh** et plus encore :

Utilisation

Mes tentatives pour apprivoiser GuixSD

Les packages sont installés ou mis à niveau vers la dernière version disponible dans la distribution actuellement disponible sur votre ordinateur local. Pour mettre à jour cette distribution, avec les outils Guix, **vous devez exécuter guix pull** : la commande télécharge le dernier code source Guix et les dernières descriptions de paquetages et le déploie. Le code source est téléchargé depuis un référentiel Git.

Une fois l'opération terminée, **guix package** utilisera les packages et les versions de package de cette copie de Guix qui vient d'être récupérée. De plus, toutes les commandes Guix et tous les modules Scheme seront également repris de cette dernière version. De nouvelles sous-commandes guix ajoutées par la mise à jour sont également disponibles.

Tout utilisateur peut mettre à jour sa copie Guix à l'aide de **guix pull**. L'effet est limité à l'utilisateur qui exécute guix pull. Par exemple, lorsque l'utilisateur root exécute guix pull, cela n'a aucun effet sur la version de Guix visualisée par l'utilisateur, et inversement.

Pour les mises à jour, j'utilise les lignes de commande suivantes (à noter, la commande **guix package -u** vient après celle de **guix pull** :

```
$ guix pull
```

```
$ guix package -u
```

```
$ sudo guix system reconfigure /etc/config.scm
```

Maintenant, je veux modifier 'config.scm' pour affiner la configuration. **Vi(m)** n'est pas encore installé.

Un regard sur la configuration du système me montre comment ajouter **gnome** et **xfce** :

https://www.gnu.org/software/guix/manual/html_node/Using-the-Configuration-System.html#Using-the-Configuration-System

```
# vim /etc/config.scm
```

/etc/config.scm

```
;; This is an operating system configuration template
;; for a "desktop" setup with GNOME and Xfce and
;; where "host-name" is my own laptop.

(use-modules (gnu) (gnu system nss))
(use-service-modules desktop)
(use-package-modules certs gnome)

(operating-system
  (host-name "lxx-xxx-xx-xx-xxx-xxx-xx6")
  (timezone "Europe/Paris")
  (locale "en_US.utf8")

  ;; Assuming /dev/sda is the target hard disk, and "my-root" is
  ;; the label of the target root file system.
  (bootloader (grub-configuration (target "/dev/sda")))
  (file-systems (cons (file-system
                        (device "my-root")
                        (title 'label)
                        (mount-point "/")
                        (type "ext4"))
                      %base-file-systems))

  (swap-devices '("/dev/sda2"))

  (users (cons (user-account
                (name "hubert")
                (comment "Cathy's brother")
                (group "users")
                (supplementary-groups '("wheel" "netdev"
```

```
                                          "audio" "video"))
                (home-directory "/home/hubert"))
              %base-user-accounts))

  ;; This is where we specify system-wide packages.
  (packages (cons* nss-certs         ;for HTTPS access
                   gvfs              ;for user mounts
                   %base-packages))

  ; Add GNOME and/or Xfce---we can choose at the log-in
  ;; screen with F1.  Use the "desktop" services, which
  ;; include the X11 log-in service, networking with Wicd,
  ;; and more.
  (services (cons* (gnome-desktop-service)
                   (xfce-desktop-service)
                   %desktop-services))

  ;; Allow resolution of '.local' host names with mDNS.
  (name-service-switch %mdns-host-lookup-nss))
```

Invocation de guix system

https://www.gnu.org/software/guix/manual/html_node/Invoking-guix-system.html#Invoking-guix-system

Maintenant que j'ai modifié la configuration du système, demandons à **guix system** de le reconfigurer :

```
$ sudo guix system reconfigure /etc/config.scm
```

```
# reboot
```

La nouvelle version du noyau est proposée.

Dans ma session utilisateur, je choisis le bureau par défaut **xfce**

Après quelques instants, le bureau **xfce** apparaît. Cela semble bien fonctionner, le navigateur **Web** de GNOME fonctionne parfaitement et il est fluide.

Je vais de surprises en surprises, même **GNOME** s'ouvre sans problèmes et le look est très sympa.

Dans le navigateur **Web**, je peux voir des vidéos et la carte son fonctionne très bien.

À ce stade, je remarque que les sessions de bureau prennent quelques instants pour démarrer.

J'essaie de continuer, car je dois appliquer les paramètres régionaux **fr** pour le clavier, **ssh** en mode utilisateur et essayer d'apprendre les règles de l'administrateur à la gestion des utilisateurs, de leurs droits, etc.

"Tout utilisateur peut mettre à jour sa copie Guix en utilisant '**guix pull**', et l'effet est limité à

l'utilisateur qui exécute guix pull. Par exemple, lorsque l'utilisateur root exécute 'guix pull', cela n'a aucun effet sur la version de Guix que cet utilisateur utilise. Alice voit, et vice versa. ”

```
$ guix pull
.....................................................................
...

 GNU Guix mis à jour et déployé avec succès sous
'/home/hubert/.config/guix/latest'
```

Pour une mise à jour complète:

```
$ guix package -u
```

```
$ sudo guix system reconfigure /etc/config.scm
```

Vivant dans une zone numérique fracturée, je peux parfois observer un temps très long lors de la mise à jour de la session utilisateur par cette dernière commande. Une douzaine d'heures est parfois nécessaire. Une fois il m'avait fallu 22 heures et une autre fois 24 heures, ce furent les 2 seules fois. Je pense que c'est très différent dans les villes, peut-être 3 ou 4 heures ?

En mode utilisateur, je peux ajouter mon hébergement Web distant dans un **signet Nautilus** et **ssh** fonctionne parfaitement ;)

C'est la première fois de ma vie que je suis confronté à un système aussi impressionnant de stabilité et de cohérence. Mais je me demande parfois comment trouver mes marques et commencer à apprivoiser le système.

Le canal irc est une mine d'infos et l'accueil vraiment sympa :
https://www.gnu.org/software/guix/contact/irc/

Et voilà, c'est un retour de ce que je réussis à faire en tant qu'utilisateur final de Logiciel Libre plus ou moins expérimenté 😁

From:
http://debian-facile.org/ - **Documentation - Wiki**

Permanent link:
http://debian-facile.org/utilisateurs:gonzoleri:tutos:installation-configuration-de-guixsd

Last update: **01/10/2023 11:41**