Le HOWTO du Terminal X

Scot W. Stevenson (scot@catzen.gun.de)
Adaptation française : Pierre Vassellerie (Pierre.Vassellerie@obspm.fr)
Version 1.0f BÉTA (11 Novembre 1996)

Ce document est une brève introduction à ¡¡ Comment connecter un Terminal X à un PC sous Linux ¿¿. Il nécessite une connaissance de base du système X Window, de l'adressage TCP/IP et des cartes Éthernet.

Table des matières

1 Hommage et dédicace

« **A René, le traducteur.** »

2 Introduction

Ceci est la première version du document et devra être considérée comme BÉTA. Ce document est plus une méthode d'installation qu'un document complet sur les interactions entre les terminaux X et Linux. Les discussions sur les mécanismes de contrôle d'accès (c-à-d. xaccess, xhost, MIT-COOKIEs) et l'utilisation de NFS ne sont pas incluses pour l'instant.

La plupart des terminaux X ont maintenant une multitude de caractéristiques avancées qui leur permettent d'être plus que de simples serveurs X. Pour la plus grande partie, ces caractéristiques seront ignorées.

2.1 Changement par rapport aux versions précédentes

(Il n'y a pas de version précédente donc rien n'a changé) (NdT : pareil)

2.2 Responsabilités

L'auteur, les distributeurs (NdT : ni le traducteur) de ce HOWTO ne peuvent en aucun cas être tenus pour responsables des dommages physiques, financiers ou moraux survenus en suivant les suggestions de ce texte.

2.3 Copyright

Si vous avez des questions veuillez contacter Greg Hankins, le coordinateur des HOWTO Linux, à Greg Hankins (`gregh@sunsite.unc.edu`). Vous pouvez utiliser finger avec son adresse pour obtenir son numéro de téléphone et de de plus amples informations sur la manière de le contacter.

2.4 Nouvelles versions et réactions

Les nouvelles versions de ce document peuvent être trouvés sur `sunsite.unc.edu:/pub/Linux/docs/HOWTO/`. Si vous n'avez pas d'accès FTP vous pouvez essayer d'obtenir les fichiers d'aide Linux par Bill Riemers. Envoyez un courrier électronique à `bcr@physics.purdue.edu` avec comme sujet "help" pour plus d'information et un fichier d'index.

Toutes additions, corrections ou commentaires sur ce document seront les bienvenues! Pour cela, veuillez envoyer un courrier électronique à Scot W. stevenson (`scot@catzen.gun.de`) (NdT : pour la version française à `Pierre.Vassellerie@obspm.fr`)

.

J'aimerais particulièrement avoir de vos nouvelles si vous avez déjà l'expérience de la liaison d'un terminal X à une machine Linux même si c'est seulement quelque chose comme "a travaillé sur cette machine depuis ce terminal."

Dans les cartons pour les prochaines versions il y a les mécanismes de contrôle d'accès et l'utilisation des systèmes de fichiers NFS pour le démarrage.

3 Contexte

Cette section fournit de l'information très basique pour les non-familiers avec le système X Window et sa terminal-ologie. Si vous connaissez déjà les rudiments de X ou les terminaux X vous devriez pouvoir sauter cette partie sans effet indésirable.

3.1 Qu'est-ce que X ?

Le système X Window ou juste X (jamais X Windows), est un système de fenêtrage portable et transparent pour le réseau comme indiqué (NdT : en anglais) dans la page du manuel en ligne. Il fournit un environnement graphique qui permet la communication à travers différents systèmes d'exploitation, constructeurs et types de matériels. Quand les gens parlent d'un système de fenêtrage en connexion avec Unix, ils font allusion presque toujours à X.

La plus importante caractéristique de X dans notre cas est la stricte séparation entre les programmes qui contrôlent le matériel local avec lequel l'utilisateur s'interface (écran, clavier, souris, etc.) et les programmes que l'utilisateur veut réellement lancer (éditeur, tableur, jeux). Ceci signifie que l'interface logicielle, qui est appelée le serveur X, peut être sur une machine, tandis que les programmes réels, ou clients X, peuvent être sur une ou même plus d'une machine à des endroits totalement différents. Notez que les termes "serveur" et "client" sont utilisés dans le sens inverse dans lequel ils sont utilisés généralement avec d'autres applications client/serveur.

3.2 Qu'est-ce qu'un terminal X ?

Un terminal X (écrit TX à partir de maintenant) est un ensemble spécialisé de matériel et de logiciel qui se combinent pour former un serveur X, qui est la partie de X qui gère les entrées et sorties vers et en provenance de l'utilisateur. Dans le cas le plus primitif, seul le programme serveur de X et le logiciel de communication tournent sur le TX. Même le gestionnaire de fenêtre tourne sur l'ordinateur hôte auquel le TX est connecté par Ethernet (ou dans de rares cas par des lignes séries ou d'autres types de réseaux), en utilisant TCP/IP.

Le matériel constituant un TX incluera au moins un (grand) écran, un clavier, une souris, de la RAM et des cordons pour la connexion à Ethernet. La plupart des TXs n'ont pas de disque dur, de lecteur de disquette ni d'autres matériels de transfert de données. Ceci signifie que le TX a son système d'exploitation en ROM (rare) ou alors le charge depuis un hôte sur le réseau auquel il est connecté.

Pour récupérer son système d'exploitation d'un ordinateur hôte sous Linux au moment du lancement le TX joue le scénario suivant : il envoie un appel à l'aide à travers le réseau avec son numéro Ethernet comme nom d'étiquette. Un ordinateur sur le réseau recherche ce numéro dans la liste des numéros des TX qu'il est autorisé à aider à démarrer. Si ce numéro est trouvé, il envoie au TX le numéro IP qui lui a été assigné (par le daemon `bootpd`). Ceci permet alors au TX de charger son système d'exploitation et les autres données dont il a besoin depuis le disque dur de l'ordinateur hôte (généralement par `tftp` ou `nfs`). Ceci est la procédure généralement employée, expliquée rapidement.

Un TX est donc en fait un ordinateur complètement équipé avec son propre numéro IP, sa RAM, son programme et son matériel indépendant. C'est en fait un savant idiot. Il est doué pour ce qu'il fait le mieux, c'est à dire gérer les communications et graphiques par le protocole X11.

3.3 Avantages et inconvénients

Idéalement, un TX est silencieux, rapide. D'habitude sans ventilateur, lecteur de disquette ni disque dur, ils ne créent pas de bruit du tout et avec quelques mètres de câble Ethernet vous pouvez placer votre ordinateur bruyant dans une pièce différente et avoir le TX silencieux sur votre bureau. Le TX est construit pour gérer du graphisme X et est donc sensé être plus rapide que, disons, un programme de serveur X sous MS Windows, DOS ou MacOS (NdT : vu la rapidité des processeurs actuels et des réseaux, est-ce encore le cas ?).

Avec le serveur sur une machine et le client sur une autre, le processeur n'a pas à se charger des deux à la fois. Cependant ceci n'est guère perceptible en termes de vitesse (les données devant alors être transportées par Ethernet), et seuls la charge du CPU et l'usage de la mémoire du client sous Linux sont remarquables.

En revanche, vous aurez besoin d'une carte Ethernet ce qui généralement signifie renoncer à un slot et à une IRQ. Suivant le fabricant, le logiciel pour le TX peut prendre environ 20 Moctets de l'espace du disque dur sur votre machine Linux. Vous pouvez toujours effacer quelques fichiers inutilisés une fois que vous êtes arrivé à comprendre ce qui est réellement nécessaire. La plupart des TXs ont besoin que la machine hôte ait les daemons `bootpd` et `tftpd` installés et actifs - chacun pouvant engendrer des trous dans votre sécurité. Vous voudrez probablement avoir un démon supplémentaire, `xdm`, lancé en tâche de fond (NdC : `xdm` est le gestionnaire d'accès X, équivalent à `getty` sur un terminal texte). Et finalement, le grand écran du TX prendra beaucoup de place sur votre bureau, déjà bien encombré.

3.4 De quoi ai-je besoin?

Question pertinente !

Tout d'abord, vous avez besoin d'un TX. Si vous avez beaucoup d'argent, et j'ai bien dit beaucoup, vous pouvez sortir et vous en payer un. Jim Morton (`jim@applix.com`) poste régulièrement une liste de TXs et leur prix dans `comp.windows.x`. La chance peut aussi vous sourire. Certains vieux TXs ne peuvent pas être utilisés avec certains systèmes comme MSDOS, Windows ou OS/2. Certaines entreprises décident donc de s'en débarrasser plutôt que de s'en encombrer.

Du côté de l'ordinateur Linux, vous aurez besoin d'une carte Ethernet. Bien qu'il soit en théorie possible d'utiliser un TX via une ligne série et SLIP, ceci n'est pas recommandé à moins que vous ayez des tendances masochistes. Jettez alors un coup d'oeil sur l'Ethernet-HOWTO, maintenu par Paul Gortmaker, (`Paul.Gortmaker@anu.edu.au`), qui contient des informations sur comment bien acheter et installer les cartes Ethernet. SLIP et CSLIP sont traités dans le même document, au cas où vous n'auriez pas d'autre choix. Dans ce cas, il vous faudra aussi consulter le Serial-HOWTO de Greg Hankins (`gregh@cc.gatech.edu`) afin d'obtenir les meilleures performances.

Vous aurez également besoin d'avoir un noyau supportant TCP/IP ainsi qu'un numéro IP pour votre machine et pour le TX. Le Net-2-HOWTO de Terry Dawson (`terryd@extro.ucc.su.oz.au`) traite de tout ceci.

Enfin vous aurez besoin d'avoir X installé sur votre machine Linux. En théorie vous avez uniquement besoin des clients X et des programmes comme `xdm`, les serveurs étant inutiles. Mais autant faire un dernier effort et installer le serveur X sur votre hôte sous Linux à l'aide du XFree86-HOWTO de Helmut Geyer (`Helmut.Geyer@uni-heidelberg.de`).

4 Câbles, réseaux et daemons

Cette section traite des configurations nécessaires du matériel et du logiciel pour réussir à connecter le TX à la machine Linux. Par convention, le TX est appelé "murmure" (parce qu'il ne fait pas de bruit) et la machine Linux hôte "nunux" (parce qu'elle est sous Linux). Ils font tout deux partie du domaine "grenouille" en France (`.fr`) (NdT : les noms et domaines ont été adaptés pour un usage en France `:-)`). Leurs numéros IP sont :

```
192.168.13.1    pour  nunux.grenouille.fr   (la machine Linux)
192.168.13.41   pour  murmure.grenouille.fr (le TX).
```

Notez que ceux-ci sont des numéros IP pour des systèmes isolés, non connectés à un réseau plus grand comme Internet, et qu'à ma connaissance il n'y a aucun domaine `grenouille` en France (mais ça ne saurait tarder). Nous supposerons qu'il n'y a aucune autre machine sur le réseau et que NFS n'est pas installé.

N.B. Si quelqu'un a utilisé NFS pour la connexion avec son TX, j'aimerais vivement le savoir.

4.1 Connexion physique

Ceci devrait être aussi facile que de brancher les câbles sur les deux machines. Notez que certains TXs ont deux entrées série qui ne peuvent fonctionner qu'à certaines vitesses si les deux sont utilisées en même temps. Vérifiez le manuel de votre TX pour les détails. Vous aurez besoin du numéro Ethernet du TX plus tard. Il est affiché dès l'allumage du TX même si aucune connexion n'est encore établie.

Dès que vous aurez les câbles en place, vous serez en mesure de tester le lien Ethernet. Après avoir allumé le TX, il devrait commencer par se plaindre que ses appels à l'aide à un `bootpd` et/ou un `tftpd` n'ont pas de réponse, et ensuite réalisera le lancement du système d'exploitation (généralement implanté dans la ROM du TX). Celui-ci inclut souvent une commande `ping` qui vous permettra de tester la connexion par Ethernet du TX à la machine Linux. Ne paniquez pas si cela ne fonctionne pas dans l'autre sens (la machine sous Linux tentant un ping sur le TX) car certains (rares) TX nécessitent d'avoir leur système d'exploitation complet pour être en mesure de répondre.

4.2 Configurer l'accès au réseau

La configuration de l'accès au réseau est traitée dans le Net2-HOWTO comme mentionné plus haut. Nous considèrerons ici que vous avez déjà TCP/IP tournant sans aucun problème sur votre machine. Le TX est désormais considéré comme un banal ordinateur connecté sur le même réseau. Vous devez vous assurer que la machine sous Linux comme le TX connaissent chacun le numéro IP de l'autre et que le réseau fonctionne.

4.2.1 Configuration de la machine sous Linux

L'information sur le TX doit être au minimum incluse dans les fichiers :

/etc/hosts

Ajoutez une ligne avec le numéro IP attribué au TX, comme par exemple :

```
# /etc/hosts
#
# adresse et nom de la machine.
# lprhost et loghost sont optionels
#
192.168.13.1    nunux.grenouille.fr nunux lprhost loghost

# nouvelle ligne d'informations sur le TX
192.168.13.41   murmure.grenouille.fr murmure
```

/etc/ethers

fournit une liste de numéros Ethernet avec les noms de machines correspondants. Cela ressemble à :

```
04:03:e8:cc:0d:24     nunux
0f:03:11:31:45:f1     murmure

(eh oui, ces numeros Ethernet sont factices)
```

Vous aurez besoin de modifier d'autres fichiers suivant votre configuration, selon que vous utilisez `named`, `routed`, ou `gated`. Comme ce n'est pas mon cas, je serais reconnaissant si quelqu'un pouvait m'envoyer la liste des fichiers à modifier.

Vous devez ensuite relancer votre machine afin d'être certain que ces modifications soient prises en compte.

4.2.2 Configuration du terminal X

Cette configuration ne dépendant que du type de terminal X utilisé, reportez-vous au manuel de votre terminal X. Dans mon cas, le TX contient un fichier de configuration dans lequel je dois modifier les entrées :

```
ip_host_table           192.168.13.1    nunux
ip_host_table           192.168.13.1    nunux.grenouille.fr
ip_host_table           192.168.13.41   murmure
ip_host_table           192.168.13.41   murmure.grenouille.fr

file_access_1           TFTP
file_host_name_1        nunux.grenouille.fr
file_path_1             /usr/local/xterm/cestici

display_access_table    murmure
display_access_table    nunux
enable_access_control   YES

xdmcp_server            nunux
broadcast_address       192.168.13.255
default_telnet_host     nunux
```

Notez que le TX charge ses fichiers par le réseau, en utilisant le protocole `TFTP`, dans le répertoire `/usr/local/xterm/cestici`, et qu'il comprend le protocole `XDMCP` (qui permet l'utilisation de `xdm`).

Vous devrez aussi modifier d'autres paramètres comme celui donnant la liste des polices. Vous pourrez ainsi utiliser celles déjà installées sous Linux. Dans mon cas, le fichier de configuration relatif aux polices est `font..tbl` et ressemble à :

```
/usr/lib/X11/fonts/75dpi
/usr/lib/X11/fonts/100dpi
...
/usr/local/xterm/misc
/usr/local/xterm/openlook
```

Ensuite, quand le TX démarrera sur la machine Linux, vous verrez la liste des polices qu'il aura réussi à charger.

Une autre chose dont vous aurez besoin est le « backing store ». Ceci signifie que les parties de fenêtres recouvertes par d'autres ne seront pas stockées dans la RAM de la machine sous Linux mais dans celle du TX (ce qui fait gagner énormément en vitesse d'affichage). Consultez le manuel de votre TX pour de plus amples renseignements.

4.3 bootpd

`Bootpd` est le daemon qui reste à l'écoute des appels à l'aide de vos terminaux X, et qui leur répondra en leur disant qui ils sont, et où ils vont pouvoir trouver les logiciels qu'ils doivent télécharger. Pour d'étranges raisons, `bootpd` n'est pas inclus dans certaines distributions comme la Slackware 2.2.0.1. Il vous faudra donc le récupérer sur un serveur FTP ou autre. Il doit être placé dans `/usr/sbin/` (et non dans `/etc`, contrairement à ce qu'indique dans la page de manuel) sous le nom `in.bootpd`. Ajoutez ou décommentez la ligne suivante dans le fichier `/etc/inetd.conf` :

```
bootps dgram udp wait root /usr/sbin/tcpd /usr/sbin/in.bootpd
```

et relancez `inetd` par :

```
kill -HUP `ps -aexu | grep inetd | grep -v grep`
```

Ceci permettra ensuite à `inetd` de lancer `bootpd` si une requête de démarrage est détectée.

Le fichier de configuration de `bootpd` est `/etc/bootpd`. La syntaxe est expliquée dans le manuel en ligne. Dans notre exemple, le fichier `/etc/bootpd` ressemble à (« serveur » est de nouveau utilisé dans le sens classique du terme) :

```
#
# Exemple de fichier /etc/bootpd
#
# Entree utilisee par l'ensemble des terminaux
```

```
        #
        allhost:hd=/usr/local/xterm/cestici:\   # Repertoire pere contenant
le logiciel du TX
              :ds=192.168.13.1:\                # Serveur de noms du domaine
              :sm=255.255.255.0:\               # Masque de sous-reseau
              :gw=192.168.13.1:\                # Passerelles
              :ts=192.168.13.1:\                # Serveurs d'heure
              :lp=192.168.13.1:\                # Serveurs d'impression
              :to=-7200:                        # Decalage d'heure (en secondes)
        #
        # Ensuite, les descriptions pour chaque client.
        #
        murmure:ht=ethernet:\         # Type du lien physique
               :ha=0f03113145f1:\     # Numero Ethernet du terminal X
               :ip=192.168.13.41:\    # numero IP du terminal X (murmure)
               :tc=allhost:\          #
               :bf=xtermOS:           # Nom du fichier contenant l'OS du TX
```

Dans notre exemple, le TX va charger son système d'exploitation à partir du fichier `xtermOS` (entrée `bf`) contenu dans le répertoire `/usr/local/xterm/cestici` (entrée `hd`).

`bootpd` va tracer les informations sur les différents lancements dans les fichiers `/var/adm/syslog` et `/var/adm/messages` (voir configuration du fichier `/etc/syslog.conf`). Un démarrage réussi donnera :

```
        Jul 17 05:19:42 nunux in.bootpd[110]: connect from 0.0.0.0
        Jul 17 05:19:42 nunux bootpd[110]: reading "/etc/bootptab"
        Jul 17 05:19:42 nunux bootpd[110]: read 2 entries from "/etc/bootptab"
        Jul 17 05:19:43 nunux bootpd[110]: request from hardware address
0F03113145F1 Type 1
        Jul 17 05:19:43 nunux bootpd[110]: found 192.168.13.41 murmure
```

Après avoir aidé le TX à démarrer, `bootpd` va continuer à tourner dans l'attente d'autres appels à l'aide durant environ quinze minutes, puis se terminer si aucun travail supplémentaire n'est requis.

4.4 tftpd

Le protocol trivial de transfert de fichier (TFTP) est utilisé par le terminal X pour charger son système d'exploitation depuis le disque dur du serveur de démarrage. Il devrait être inclus dans toutes les distributions et ne possède aucun fichier de configuration. Vous pouvez tester `tftp` en tapant la commande `tftp`.

Comme avec `bootpd`, vous devrez ajouter ou décommenter la ligne suivante dans le fichier `/etc/inetd.conf` :

```
tftp dgram udp wait root /usr/sbin/tcpd /usr/sbin/in.tftpd
```

et relancer inetd par :

```
kill -HUP `ps -aexu | grep inetd | grep -b grep`
```

Remarquez que `tftp` ne peut accéder qu'aux fichiers ayant un droit en lecture pour tout le monde. Il faut aussi remarquer que `tftp` est un trou de sécurité potentiel que vous devez garder à l'esprit (c'est pourquoi il est bon de vérifier quels sont les fichiers ayant des droits en lecture universels sur son système, ou de faire un `chroot` de `tptfd` à l'aide du TCP-wrapper `/usr/sbin/tcpd`). La version de `tftp` incluse dans certains paquetages de Linux ne contient pas les options "-r" ou "-s" permettant une utilisation sécurisée.

`tftp` renvoie aussi des messages système dans le fichier `/var/adm/messages`. Si le lancement est réussi, des lignes de ce genre apparaissent :

```
Jul 17 05:19:43 nunux in.tftpd[111]: connect from murmure
Jul 17 05:19:58 nunux in.tftpd[113]: connect from murmure
Jul 17 05:19:59 nunux in.tftpd[115]: connect from murmure
Jul 17 05:20:00 nunux in.tftpd[117]: connect from murmure
etc...
```

Cela montre que le TX charge les fichiers dont il a besoin dans le répertoire situé sur le serveur Linux. Des messages doivent apparaître sur le TX au fur et à mesure du chargement.

4.5 Test de la connexion

Une fois que vous aurez modifié les fichiers ci-dessus, vous devriez être prêt à démarrer le TX. Suivant le constructeur, des messages plus ou moins explicites doivent apparaître. Faites attention aux messages indiquant une erreur au chargement d'un fichier.

Si tout va bien, on devriez en arriver au stade où le TX lance sa propre version de X. On a alors un fond gris et un curseur en croix. Si vous avez déjà lancé `xdm`, vous devriez même avoir la fenêtre de login `xdm`. Il se peut que des choses bizarres apparaissent si certains champs de la configuration de `xdm` ne sont pas corrects. Préparez-vous à tuer xdm en tant que `root` en ultime recours.

La plupart des terminaux X ont des fonctionalités intégrées, comme un client `telnet`, dans le système d'exploitation chargé au démarrage. Cela vous permet alors de tester la connexion en faisant un `telnet` vers le serveur ou une autre machine.

Vous pouvez désormais lancer des programmes X sur le terminal X en utilisant l'option `display`. Par exemple :

```
xclock -display murmure:0 &
```

doit faire apparaître l'horloge "xclock" sur votre TX.

Vous pouvez même (et c'est d'ailleurs recommandé) lancer un gestionnaire de fenêtres tel que `fvwm` de la même manière.

5 Faire tourner X

Cette section porte sur la configuration de xdm afin qu'une invite de connexion soit disponible sur les terminaux X, et que le retour à celle-ci soit réalisé quand un utilisateur se déloge. Le programme `xdm` est l'équivalent pour TX des programmes de connexion sur consoles texte. Il est normalement inclus dans toutes les distributions de Linux.

5.1 Configuration de xdm

Les fichiers de configuration de `xdm` se trouvent dans `/usr/X11R6/lib/X11/xdm` (`/usr/X11R6` peut être un lien sur `/usr/X11`). Le principal fichier de configuration est `xdm-config`. Vous devez y trouver, parmi d'autres, les lignes :

```
DisplayManager._0.authorize:    true
DisplayManager._0.setup:        /usr/X11R6/lib/X11/xdm/Xsetup_0
DisplayManager._0.startup:      /usr/X11R6/lib/X11/xdm/GiveConsole
DisplayManager._0.reset:        /usr/X11R6/lib/X11/xdm/TakeConsole
```

Ces lignes indiquent les fichiers contrôlant l'écran quand X est lancé sur la machine Linux elle-même. Pour la gestion du TX, nous devons ajouter les lignes :

```
DisplayManager.murmure_0.authorize:     true
DisplayManager.murmure_0.setup:   /usr/X11R6/lib/X11/xdm/Xsetup_murmure
DisplayManager.murmure_0.startup: /usr/X11R6/lib/X11/xdm/Xstartup
DisplayManager.murmure_0.reset:   /usr/X11R6/lib/X11/xdm/Xreset
```

Remarquez que `murmure_0` est la notation `xdm` de `murmure:0`, tout comme `_0` est l'équivalent de `:0`. Remarquez aussi que `GiveConsole` a été remplacé par `Xstartup`, qui dans mon cas est un script ne faisant rien, et que `TakeConsole` a été remplacé par `Xreset`, qui est lui aussi un script ne faisant rien. Ces fichiers contrôlent à la fois la « possession » de la console quand X est utilisé directement sur la machine Linux, et qu'il n'y ait pas de problème d'accès à la console uniquement parce qu'un TX est connecté à la machine.

Ces fichiers de démarrage lancent différents programmes avant que l'invite de connexion soit placée à l'écran. C'est est l'endroit indiqué pour vous dire d'utiliser `xv` ou un programme similaire afin de placer une image en fond d'écran. Dans ce cas, copier le fichier `Xsetup_0` en tant que `Xsetup_murmure` et modifier ce dernier.

Comme cette question réapparait encore et encore : une méthode simple de mettre une image en fond d'écran est de mettre la ligne :

```
nice xv -root -quit -rmode 5 <fichier_image> &
```

ou quelque chose du même style dans le fichier de démarrage. `fichier_image` sera alors affiché en fond d'écran derriére l'invite de connexion de `xdm`. Notez que certains TX renverront un message d'erreur si cette image est trop grande.

Le fichier `Xaccess` permet de contrôler l'accès à la machine. Généralement vous n'aurez pas à le modifier. `Xaccess` permet aussi de donner à l'utilisateur à choisir dans une liste de machine (`chooser`) si plusieurs machines du réseau acceptent l'accès depuis un TX.

Le fichier `Xresources` permet de fixer la taille, la forme et le message de bienvenue de la fenêtre de l'invite de connexion. Ainsi en remplaçant la ligne :

```
DisplayManager*resources:       /usr/X11R6/lib/X11/xdm/Xresources
```

par les lignes

```
DisplayManager._0.resources:        /usr/X11R6/lib/X11/xdm/Xres_0
DisplayManager.murmure_0.resources: /usr/X11R6/lib/X11/xdm/Xres_mu_0
```

où `Xres_mu_0` est le fichier de ressources pour le TX `murmure` et `Xres_0`, celui pour la console de la machine Linux. Vous pourrez donner des valeurs différentes pour le TX et pour la machine sous Linux.

Normalement vous ne devriez pas avoir à modifier le fichier `Xsession`.

La configuration du fichier `Xservers` est aussi presque triviale. Au pire vous aurez à décommenter (cas de la distribution Slackware 2.2.0.1) la ligne :

```
:0 local /usr/X11R6/bin/X
```

ou une ligne ayant le même effet. Cela permet le démarrage automatique du serveur X sur la machine hôte `nunux` lors de l'appel de `xdm`. Si vous commentez cette ligne, X ne sera pas lancé sur la machine `nunux` lors d'un appel à `xdm`. C'est le cas si vous désirez que X ne soit utilisé que sur les TX et non sur la machine hôte. Dans ce cas, vous pourrez démarrer X sur la machine `nunux` par la commande `startx` et pour le temps que vous voudrez, sans que cela ait d'incidence sur le TX.

Si votre TX n'a pas `XDMCP`, vous devrez alors ajouter une ligne telle que :

```
murmure:0 foreign
```

`XDMCP` est un protocole standardisé qui, par exemple, laisse les TXs discuter avec leurs hôtes. Si votre TX supporte `XDMCP` vous ne devez pas ajouter cette ligne. Ceci laisserait pensé à `xdm` qu'un TX ne comprend pas `XDMCP`, alors qu'au même moment celui-ci tenterait d'utiliser ce protocole pour se connecter. Cela peut conduire à de nombreux effets fortement désagréables comme la lutte de deux `xdm` pour la prise de contrôle.

Vous pouvez utiliser les entrées du fichier `xdm-config` même s'il n'y a pas de ligne dans `Xservers` pour le TX. Vous pourrez toujours personnaliser l'invite de connexion, etc., si le TX utilise `XDMCP`.

Afin que `xdm` démarre à chaque lancement de Linux, vous pouvez ajouter la ligne :

```
/usr/bin/X11/xdm
```

dans `/etc/rc.d/rc.local`. Certains démarrent `xdm` par le biais du fichier `/etc/inittab` en remplaçant :

```
# Default runlevel.
id:3:initdefault:
```

(première entrée du fichier) par

```
# Default runlevel.
id:5:initdefault:
```

Dans tous les cas vous devez avoir `xdm` dans la liste des processus actifs après le redémarrage.

5.2 Questions sur l'accès

(Cette importante partie sera développée ultérieurement, nous travaillons dessus.)

Pour savoir si un utilisateur peut accéder à l'environnement d'un TX depuis la machine `nunux`, connectez vous (en autre chose que `root`) et lancez la commande :

```
        xsetroot -solid white -display murmure:0 &
ou
        xterm -display murmure:0 &
```

Essayez cela quand quelqu'un est connecté sur le TX et qu'il n'y a que l'invite de connexion de `xdm`. Suivant votre position, la possibilité d'accès au TX depuis une session sur la console sera plus ou moins une possibilité inattendue qu'un bug.

6 Erreurs et inconnues

6.1 Problèmes connus

Certains problèmes apparaissent alors, tout comme certaines possibilités intéressantes, qui pourraient s'avérer à leur tour des problèmes. Si vous découvrez d'autres problèmes de ce type, faite-le moi savoir.

talk

La discussion interactive fonctionnera si un utilisateur sur le TX se loge comme utilisateur sur la machine nunux, mais pas dans l'autre sens. Il existe une possibilité de supprimer ce problème mais je ne m'en souviens plus.

who

Un utilisateur logé sur un TX n'apparaîtra pas, même si cette commande est lancée depuis le TX. Ceci est sûrement la raison pour laquelle `talk` plante quand l'utilisateur sur la console tente de se connecter à l'utilisateur sur le TX.

xlock

Un appel standard de la commande `xlock` a pour unique effet de faire apparaitre un message disant que l'écran du TX est bloqué. L'option `-remote` doit donc être utilisée afin que `xlock` bloque effectivement le TX. Il faut noter que certains modes de `xlock` sont de gros consommateurs de ressources. `Qix` semble plus approprié aux TXs que les autres modes (consultez la FAQ de Art Mulder, cf. ci-dessous, pour de plus amples détails).

xv

Certains TX n'ont pas suffisamment de mémoire pour pouvoir afficher des images trop grandes ou trop complexes, en plus du en fond d'écran. Dans ce cas utilisez `xsetroot` pour supprimer ou changer l'image de fond avant d'utiliser `xv` (NdT : de tels problèmes sont aussi très fréquents avec le trop gourmand `Netscape`).

6.2 Terminaux testés

Les différentes procédures décrites dans ce document ont été testées sérieusement pour la connexion d'un terminal X Tektronix XP23 sur un 386DX-33MHz avec 16Mo de RAM tournant sous Linux 1.2.3 et XFree86 Version 3.1.1 (les fichiers de la distribution Slackware 2.2.0.1).

6.3 Enrichissement personnel

De plus ample informations sur X peuvent être trouvées sur Internet :

– David B. Lewis (`dbl@ics.com`) poste sur comp.windows.x la FAQ (Foire Aux Questions) de ce groupe, ainsi que sur `news.answers` et `comp.answers`. Ce document contient entre autre les adresses où trouver plus d'informations sur X.

– Steve Kotsopoulos (`steve@ecf.toronto.edu`) poste dans les mêmes groupes sa FAQ « X on Intel-based ».

– Art Mulder (`art@cs.ualberta.ca`, poste régulièrement sur le groupe `comp.windows.x` sa FAQ « Getting more performance out of X », qui contient de nombreux renseignements utiles pour l'installation de X sous Linux.

7 Remerciements

7.1 Remerciements originaux

Comme toujours les premiers remerciements vont à Linus B. Torvalds `torvalds@kruuna.helsinki.fi`.
De nombreux autres vont à Klaus ter Fehn `ktf@bc3.gun.de` pour avoir rendu possible ce document et à Douglas K. Stevenson `duck@catzen.gun.de` pour l'avoir rendu passable.

7.2 Remerciements du traducteur

Histoire de copier, je remercie aussi Linus B. Torvalds `torvalds@kruuna.helsinki.fi`.
Mais aussi les relecteurs

– Bernard Choppy (`choppy@imaginet.fr`) qui nous fait toujours un travail d'excellente qualité.

– Olivier Tharan (`tharan@galaxie.int-evry.fr`) qui vient de se lancer dans cette grande aventure de la traduction de HOWTO.

... et à Jean-Philippe LATRUFFE pour l'aide à la traduction.