

Guide pratique de l'utilisateur de X Window

Version française du *The X Window User HOWTO*

Hal Burgiss

hal@foobox.net

Version : 3.1.fr.1.0
3 Octobre 2006

Revision History

Revision 3.1.fr.1.0	2006-10-03	Revised by: YM
Première traduction française. Conversion au format docbook 4.2.		
Revision v3.1	2002-10-10	Revised by: HB
Quelques petits ajouts et mises à jour. (<i>Some minor additions and updates.</i>)		
Revision v3.0	2002-03-06	Revised by: HB
Réécriture du document original. Conversion au format DocBook. Très nombreuses modifications. (<i>Rewrite of the original document. Convert to DocBook. Many, many changes.</i>)		

Ce document donne aux utilisateurs de Linux les informations de base pour comprendre et configurer le système X-Window. Le document est sensé être de niveau débutant. On part du principe que les bases en configuration de logiciels sont acquises, et que le système X Window est installé et fonctionne.

1. Introduction

Le *système X-Window* est un environnement avancé, de traitement graphique et de réseau, qui a été conçu depuis le début comme système multi-utilisateurs. La première version de X est sortie en 1984. Si vous ne connaissez pas les concepts de base qui s'appliquent à X et aux composants associés, il faut d'abord lire le *X Window System Architecture Overview HOWTO*, <http://linuxdoc.org/HOWTO/XWindow-Overview-HOWTO/index.html> (<http://linuxdoc.org/HOWTO/XWindow-Overview-HOWTO/index.html>), pour avoir une idée de la façon dont les divers composants s'assemblent. On y trouvera également des propositions de définitions

de divers termes relatifs à X-Window dans [l'annexe](#), si les concepts tels que “affichages” [*displays*] ou “clients X” vous posent problème dans ce contexte.

Dans ce document, nous aborderons la configuration et l'utilisation de base de X-Window sous Linux. Nous regarderons également la façon la plus courante de lancer X avec Linux, et comment on peut configurer le démarrage, ainsi que les questions qui s'y rattachent. Nous n'étudierons ni la configuration du gestionnaire de fenêtre (par exemple fvwm) ni celle de l'environnement graphique (KDE et GNOME). Il y a bien trop de variables et les changements sont trop fréquents. Bien évidemment l'utilisateur interagit beaucoup plus directement avec ces composants que le serveur X lui-même ; des lectures complémentaires seraient donc bien utiles. Etudiez la documentation fournie en local et consultez les pages web respectives pour obtenir plus d'informations.

Voici quelques autres points importants à retenir :

- X est un système client-serveur et multi-utilisateur à tout point de vue, et pas uniquement une interface graphique.
- X n'est pas intégré dans le système d'exploitation, il s'installe au-dessus, comme les autres serveurs.
- X est un standard ouvert, et tourne sur de nombreuses plateformes.
- Ce que vous voyez en fait à l'écran est le résultat de divers composants, fonctionnant tous ensemble : le système d'exploitation, X, le gestionnaire de fenêtres, et le cas échéant, un environnement graphique comme GNOME ou KDE. Ceux-ci sont des composants prêts à l'emploi [*plug-and-play*], c'est-à-dire qu'il est possible de remplacer un composant particulier sans toucher aux autres.
- Chacun de ces composants a sa propre configuration. Cela donne un système très souple et potentiellement très robuste. Cela ajoute également de la complexité.

L'intérêt sera ici uniquement porté sur X tel qu'il est implémenté par le projet le projet XFree86, (<http://xfree86.org>), sous Linux. Il existe d'autres implémentations, notamment des versions commerciales. XFree86 v4.x est sortie depuis longtemps maintenant, nous parlerons donc de cette version. La majorité du document s'applique également à la version précédente (3.x), mais il peut y avoir parfois quelques différences.

Il est également à noter qu'il y a peut-être de nombreuses façons de démarrer X et de configurer un système Linux. Nous nous concentrerons sur les méthodes courantes que l'on trouve dans les distributions Linux. En outre, les revendeurs peuvent faire varier l'emplacement des fichiers de configuration et leur nom. Gardez ceci en tête si vous rencontrez de telles anomalies dans ce document. Si cela pose problème, votre revendeur a sûrement sa propre documentation. Et comme toujours, espérons que les pages de man correspondront à votre installation.

Nous verrons également de nombreux fichiers de configuration dans les sections suivantes. Ce sont tous des fichiers de type texte, vous pouvez les éditer avec votre éditeur préféré. Faites toujours une copie de

sauvegarde avant d'éditer les fichiers importants, vu que les ennuis arrivent toujours quand on ne les attend pas ("par exemple cp /etc/X11/XF86Config-4 /etc/X11/XF86Config-4.bak").

1.1. Nouvelles version et journal des changements

La version officielle actuelle de ce guide est disponible sur le site du projet de documentation Linux (*Linux Documentation Project* <http://www.linuxdoc.org/HOWTO/XWindow-User-HOWTO.html>). Des versions bêta peuvent être périodiquement disponibles sur <http://feenix.burgiss.net/ldp/x-user/>.

v3.1 : uniquement quelques petites modifications mineures. Comporte un lien vers <http://www.plig.org/xwinman/> qui est très utile pour l'achat de gestionnaires de fenêtre. Ajout d'un lien vers la fluxbox, un gestionnaire de fenêtres avec onglets. Ajout d'une courte section sur l'amélioration des performances en réseau. Vérification du fonctionnement de tous les liens.

v3.0 : document largement remanié avec de nombreuses nouvelles sections. Certaines sections ont été supprimées, pour se concentrer maintenant exclusivement sur X proprement dit (et non plus les clients comme les gestionnaires de fenêtre). Nouveau responsable de la maintenance également :-)

v2.0 : comporte des corrections de Guus Bosch, Brian J. Miller, et de moi-même, ainsi que de nombreuses nouvelles mises à jour et informations.

v1.4 : comporte des corrections de Anthony J., et quelques très bon conseils pour la sécurité de Tomasz Motylewski.

1.2. À faire

Une partie "résolution des problèmes" toute simple. Probablement pour la v3.2.

1.3. Retour d'information

Si vous avez des questions ou des commentaires à propos de ce document, n'hésitez pas à me contacter par courriel, Hal Burgiss, <hal@foobox.net>. Toutes suggestions, corrections ou ajouts sont les bienvenus. S'il y a des informations que vous souhaitez voir dans les prochaines versions, ou si vous souhaitez participer à une prochaine version, envoyez moi un message.

1.4. Au secours !!

J'ai assuré la maintenance de ce document parce qu'elle avait été abandonnée et un jour j'ai voulu faire une proposition de modification. Eh bien, pour faire court, cela a mené à une refonte majeure. Vous pouvez aider à l'amélioration de ce document en corrigeant les inexactitudes, en clarifiant les points obscurs et en proposant des améliorations. Il y a beaucoup de choses sur ce sujet que je ne connais peut-être pas ou que j'ai mal expliquées. Avec votre aide, nous améliorerons ce document et aiderons les

autres utilisateurs. Ce document a *besoin* de votre aide !

1.5. Remerciements

Merci à l'équipe de développement de XFree86 pour les efforts dépensés à fournir un GUI flexible et robuste. Merci également à toute la communauté de GNU/Linux et Open Source pour avoir permis cela.

Merci également à l'auteur original, Ray Briggleb.

Un certain nombre d'utilisateurs de comp.os.linux.x qui m'ont aidé d'une certaine façon, qu'il le sache ou non.

Finalement, <http://google.com/linux>, qui m'a sauvé de nombreuses fois avec leur incroyable référentiel d'informations. Utilisez le pour répondre aux questions non traités ici !

1.6. Copyright

Copyright © 2002, Hal Burgiss.

Sauf indication contraire, les guides pratiques de Linux sont couverts par leurs droits d'auteur respectifs. Les guides pratiques de Linux peuvent être reproduits et distribués entièrement ou partiellement, avec tout média physique ou électronique, tant que cette notification de copyright est maintenue sur toutes les copies. La redistribution commerciale est permise et encouragée ; cependant, l'auteur voudrait être avisé de telles distributions.

Toutes les traductions, produits dérivés, ou des travaux d'agrégat incorporant n'importe quel guide pratique de Linux doivent être couverts sous cette notification de copyright. C'est-à-dire, vous ne pouvez pas produire un travail dérivé à partir d'un guide pratique et imposer des restrictions additionnelles à sa distribution. Il est possible d'accorder des exceptions à ces règles sous certaines conditions ; veuillez contacter le coordonnateur des guides pratiques de Linux pour plus d'information.

En bref, nous souhaitons favoriser la diffusion de cette information par autant de canaux que possibles. Cependant, nous souhaitons maintenir copyright sur les guides pratiques, et voudrions être avisés de tout projet pour redistribuer les guides pratiques, celui-ci en particulier !

Certaines des termes mentionnés dans ce document sont des noms de marque. Sauf indication contraire, toutes les marques déposées appartiennent à leurs propriétaires respectifs.

“X Window System” est une marque déposée par le X Consortium, Inc [devenu le OpenGroup ?].

“XFree86” est une marque déposée par “The XFree86 Project, Inc (<http://xfree86.org>)”.

“Linux” est une marque déposée de Linus Torvalds.

1.7. Avertissement d'usage

Les informations et exemples donnés ici le sont à titre d'illustration. A utiliser à vos risques et périls. On a fait le maximum pour s'assurer que le contenu de ce document était exact. Si vous trouvez des inexactitudes, veuillez m'envoyer les modifications à faire.

Toute référence à une compagnie, un produit ou un nom de marque particulier ne doit pas être interprétée comme une forme de publicité.

2. XFree86

Quasiment toutes les distributions Linux sont fournies avec une implémentation du système X-Window de XFree86. C'est un projet qui nous fournit évidemment le serveur X, mais qui comporte également une suite étendue d'utilitaires et d'applications pour permettre l'implémentation d'un environnement entièrement fonctionnel de l'interface graphique.

En fait, la liste serait tout simplement trop longue s'il fallait énumérer tout ce qui est fourni avec XFree86. En plus du serveur X lui-même, voici quelques-uns des utilitaires les plus remarquables :

xdm le gestionnaire d'affichage de X.

xfs - le serveur de police de X.

twm - gestionnaire de fenêtre léger.

xterm - l'émulateur de terminal le plus connu. Egalement, **xterm3d** et **nxterm**.

xwd - utilitaire pour faire les captures d'écran et de fenêtre.

xf86config - utilitaire de configuration du serveur X.

xdpyinfo -utilitaire donnant des informations sur X. Il montre des informations très détaillées sur le serveur X.

xlsclients - liste des clients actuellement connectés au serveur X.

xlsfonts - liste des polices disponibles pour X.

appres - liste des “ressources” X qu'un programme va utiliser.

xfontsel - application permettant de voir ou de choisir les polices.

xprop - outil pour afficher les propriétés de la fenêtre, comme par exemple le nom de la classe du client.

xset - paramètre les préférences de l'utilisateur pour beaucoup de choses, y compris la souris, le clavier ou le son (sonnerie, sonnette)...

xsetroot - programme pour changer l'aspect de la “fenêtre root”, par exemple changer la couleur de fond.

xvidtune - application pour adapter les modes vidéo du serveur X et les paramètres liés à l'écran.

xwininfo - affichage d'informations sur une "fenêtre" sélectionnée.

xmodmap - utilitaire pour manipuler les valeurs/affectations des touches du clavier et celles des boutons de la souris

Enormément de polices. Et pas mal de documentation également.

Il y a en beaucoup d'autres. Nous n'aborderons que quelques uns de ces utilitaires ici. Mais n'hésitez pas à tester les autres. La plupart doivent avoir leurs propres pages de man.

2.1. Matériel

Le serveur X commande les dispositifs d'entrée (clavier, souris, etc.) et de rendu (affichage, écran).

La compatibilité matérielle est un sujet difficile, puisque c'est toujours très évolutif. Nous sommes forcés ici d'éviter les détails, puisqu'ils auront sûrement changés avant que vous ayez lu ceci. Et ce serait long et fastidieux de toute façon.

Commençons donc par quelques généralités. La *plupart* du matériel de type PC est compatible à un degré ou à un autre. Très utile ; -)

Principe de base : s'il s'agit d'un système qui utilise un protocole de longue date et banal (par exemple PS/2), la compatibilité devrait être bonne. Réciproquement, s'il utilise quelque chose de relativement nouveau, avec de la technologie innovante, il risque d'y avoir des problèmes. C'est dans la nature de la bête, avec d'un côté le développement des logiciels libres et de l'autre les fabricants qui s'intéressent surtout aux plateformes les plus répandues. Quelques fabricants sont plus coopératifs que d'autres aussi.

Maintenant, quelques indications générales:

- Les écrans - c'est simple. Intrinsèquement Linux n'a pas besoin vraiment d'être compatible avec l'écran. C'est le travail de la carte vidéo. N'importe quel moniteur compatible avec votre carte graphique fera l'affaire. Y compris les écrans plats.
- Les cartes graphiques - c'est beaucoup plus dur. La compatibilité avec le serveur X dépend du chipset. Nombre d'entre elles sont compatibles. Mais inévitablement il y a toujours des cartes plus récentes, ou même des cartes modifiées, qui ne le sont pas. De plus, certaines peuvent avoir une meilleure compatibilité et une meilleure optimisation que d'autres. Les dispositifs avancés tels que les sorties multiples d'affichages, la 3D, la sortie TV, le DRI, etc., ont une certaine compatibilité également, bien qu'il faudrait vérifier cela en priorité, car la compatibilité peut être limitée. Une liste des cartes compatibles est disponible ici : <http://xfree86.org/cardlist.html>.

Les pilotes open sources sont souvent développés incrémentalement. Par exemple, une carte particulière peut fonctionner correctement pour l'affichage de base, mais les dispositifs spécialisés tels

que la 3D peuvent ne venir que beaucoup plus tard dans le cycle de développement. C'est un cycle de développement tout à fait différent des pilotes propriétaires provenant du fabricant.

- Les claviers -- Tout clavier standard de type PC devrait aller très bien, y compris le PS/2, l'USB et beaucoup de systèmes infrarouges. De même pour beaucoup de "non standard" probablement ; -)
- Les souris et autres dispositifs de pointeur -- La plupart devraient être compatibles notamment le PS/2, le bus, le "série", l'USB et beaucoup de dispositifs infrarouges. Les souris optiques également. Unix a longtemps préféré les souris à trois boutons, même si les souris avec plus de boutons sont également compatibles. Beaucoup de souris à boule sont compatibles avec le serveur X par l'intermédiaire du "IMPS/2" (IntelliMouse), ou d'autres protocoles spécifiques, cependant certaines applications peuvent exiger une configuration supplémentaire. (Voir la section [liens](#).)
- Les ordinateurs portables ont leurs problèmes bien particuliers puisque le matériel tend à être très spécialisé, et souvent différent de ce qu'on trouve généralement sur les systèmes d'ordinateur de bureau. X est compatible avec beaucoup. Voir les informations sur <http://www.linux-laptop.net/>.

Vous pouvez vérifier "la liste de compatibilité matérielle" sur le site Web de votre distribution également. Ceci devrait donner une idée assez précise de ce qui *devrait* fonctionner avec votre version.

Les nouvelles versions de XFree86 auront évidemment une meilleure compatibilité matérielle. Si vous utilisez une version plus ancienne de Linux et n'avez pas la compatibilité matérielle totale, il faudrait regarder comment mettre à jour XFree86. Vérifiez d'abord si votre distribution a des mises à jour pour votre version.

2.2. XF86Config

Le fichier principal de configuration pour XFree86 est `XF86Config`, qui peut exister sur votre système sous le nom de `XF86Config-4` pour XFree86 v4.x, ou peut-être d'autres variantes (voir la page de man). Il est en général placé dans `/etc/X11/XF86Config`, bien que, encore une fois, le chemin puisse varier. Si un `XF86Config-4` et `XF86Config` existent, XFree86 v4.x utilisera le premier. C'est un fichier indispensable.

Le fichier `XF86Config` définit les dispositifs matériels, et d'autres composants cruciaux de l'environnement du serveur X.

Alors qu'il s'agit d'un fichier texte, et qu'il est modifiable, il est le plus souvent créé pendant l'installation par tout utilitaire que votre éditeur utilise dans ce but. XFree86 inclut également l'utilitaire **xf86config** pour faire cela, mais beaucoup de distributions ont leurs propres utilitaires similaires. Ces utilitaires peuvent être lancés après l'installation si besoin est, pour changer la configuration, ou si un nouveau matériel est installé. Commencez par lire la documentation installée en local. Si vous essayez d'éditer à la main ce fichier, n'oubliez surtout pas de faire d'abord une copie de sauvegarde puisque X ne démarrera pas si ce fichier n'est pas à son goût ; -)

Ce fichier contient diverses “sections”. Chaque section définit un certain aspect fondamental de XFree86, tel que “InputDevice” (souris, clavier, joystick, etc.), “moniteur”, ou “écran”. La page de man de XFree86Config décrit les sections et les valeurs communes pour chacun d'eux. Notez que les valeurs énumérées dans la page de man n'est pas une liste exhaustive. Il y a beaucoup d' “options” spécifiques à chaque dispositif. Visitez le site <http://xfree86.org> où vous trouverez des notes et des astuces pour votre matériel.

Le XFree86Config-4 actuel de l'auteur, tel qu'il a été généré par le programme d'installation de RedHat pour XFree86 4.1 :

```
Section "ServerLayout"
    Identifier "XFree86 Configured"
    Screen      0  "Screen0"  0 0
    InputDevice "Mouse0"  "CorePointer"
    InputDevice "Keyboard0" "CoreKeyboard"
EndSection

Section "Files"
    # L'emplacement de la base de données RGB.
    RgbPath      "/usr/X11R6/lib/X11/rgb"

    # Les entrées multiples de FontPath sont permises, elles sont
    # concaténées.
    # Par défaut, Red Hat 6.0 et suivantes utilisent maintenant un serveur de
    polices
    # indépendant du serveur X pour rendre les polices.
    FontPath "unix/:7100"
EndSection

# Section chargement des modules

Section "Module"
    Load "dbe"           # Double mise en mémoire tampon
    Load "GLcore"        # Support de OpenGL
    Load "dri"           # Infrastructure de rendu direct
    Load "glx"           # Interface pour le protocole OpenGL X
    Load "extmod"        # Diverses extensions requises
    Load "v4l"           # Vidéo pour Linux
#    Load "fbdevhw"
    Load "pex5"
    Load "record"
    Load "xie"
EndSection

Section "InputDevice"
    Identifier "Keyboard0"
    Driver      "keyboard"
```



```
        Option      "XkbLayout" "us"

#   Option  "AutoRepeat"      "500 5"

# Si vous utilisez XQUEUE, commentez la ligne ci-dessus, et décommentez la
# ligne suivante.
#   Option  "Protocol"        "Xqueue"

# Spécifiez quelles diodes électroluminescentes du clavier peuvent être
# contrôlées par
# l'utilisateur (par exemple, avec xset(1))
#   Option  "Xleds"           "1 2 3"

# Pour désactiver l'extension XKEYBOARD, décommentez XkbDisable.
#   Option  "XkbDisable"

# Pour personnaliser les paramètres XKB en fonction de votre clavier,
# modifiez les lignes
# ci-dessus (qui sont par défaut). Par exemple, pour un clavier non U.S.
# vous allez probablement utiliser :
#   Option  "XkbModel"        "pc102"
# Si vous avez un clavier US Microsoft Natural, vous pouvez utiliser :
#   Option  "XkbModel"        "microsoft"
EndSection

Section "InputDevice"
    Identifier "Mouse0"
    Driver     "mouse"
    Option     "Device"      "/dev/mouse"
    Option     "Protocol"    "IMPS/2"
    Option     "Emulate3Buttons" "off"
    Option     "ZAxisMapping" "4 5"
EndSection

Section "Monitor"
    Identifier "Sylvania F74"
    VendorName "Unknown"
    ModelName  "Unknown"
    HorizSync  30 - 70
    VertRefresh 55 - 120
    Option     "dpms"
#   Le bloc Modeline s'insère ici si nécessaire. Utilisez xvidtune pour
#   obtenir les bonnes valeurs.
EndSection

Section "Device"
    Identifier "ATI Rage 128"
    Driver     "r128"
    BoardName  "Unknown"
EndSection

Section "Device"
    Identifier "Linux Frame Buffer"
```

```

        Driver "fbdev"
        BoardName "Unknown"
EndSection

Section "Screen"
    Identifier "Screen0"
    Device "ATI Rage 128"
    Monitor "Sylvania F74"
    DefaultDepth 24
    Subsection "Display"
        Depth 24
        Modes "1400x1050" "1280x1024" "1152x864" "1024x768" "800x600"
    EndSubSection
    Subsection "Display"
        Depth 16
        Modes "1600x1200" "1400x1050" "1280x1024" "1152x864"
"1024x768" "800x600"
    EndSubSection
    Subsection "Display"
        Depth 8
        Modes "1024x768" "800x600" "640x480"
    EndSubSection
EndSection

Section "DRI"
    Mode 0666
EndSection

```

Le vôtre peut paraître tout à fait différent. Il s'agit juste d'une configuration possible avec des commentaires ajoutés gracieusement par RedHat (et par moi), et ce, pour une installation assez ordinaire. Il n'y a rien d'exotique ici, comme des écrans ou affichages multiples.

Il n'entre pas dans le cadre de ce document de donner des explications détaillées. Jetez un oeil à la page de man de `XF86Config`. Et en outre, pensez à visiter xfree86.org (<http://xfree86.org>) et à y rechercher les options spécifiques qui pourraient s'appliquer à votre carte ou à tout autre matériel.

Juste une petite note à propos de la section "Screen" ci-dessus. Remarquez qu'elle se décompose en trois sous-sections, identifiées par "Display". Chaque sous-section a une "Depth [profondeur]" différente indiquée (il y a un autre nom : ColorDepth). Les "Modes" varient également légèrement selon leur paramètre respectif de "Depth". La sous-section active de "Display" qui sera utilisée, est déterminée par le paramètre "DefaultDepth" (sauf s'il est surchargé en ligne de commande). La valeur par défaut définie dans cet exemple est "24", c'est donc la première sous-section qui sera utilisée. En outre, le "Mode" le plus élevé cité dans cette sous-section sera le mode par défaut (résolution), qui ici est le premier cité. Le premier mode cité détermine également la zone d'écran visualisable, qui peut être plus petite que le mode (résolution) lui-même. Dans ce cas, vous auriez un bureau virtuel qui serait plus grand que l'écran visualisable. Pour faire concorder la résolution et l'écran visualisable, choisissez la

plus grande valeur comme première valeur énumérée dans chaque "Mode".

Voici une deuxième note sur les "Modes" : ce que vous voyez est le résultat de mes choix pendant la configuration de **Xconfigurator** de RedHat. Ce sont des résolutions standard, mais ce n'est pas obligatoire ! La seule limite est ce que votre matériel peut accepter. Et il n'est pas obligatoire non plus d'utiliser des rapports standard hauteur/largeur. Quelque chose du genre 1355x1112 constitue un paramétrage valide (si votre matériel l'accepte et que vogue le navire !).

Le serveur de X rejettera tous les "Modes" qu'il juge erronés/invalides. Vous pouvez faire un tour des modes valides pour changer la résolution d'écran avec `Ctrl+Alt++` et `Ctrl+Alt+-` (avec les touches + et - du clavier).

Dans les versions précédant la v4.x, vous verriez également beaucoup de blocs "Modeline" qui permettaient de définir les possibilités du moniteur. Ces blocs pourraient ressembler à ceci :

```
# 1024x768 @ 100Hz, 80.21 kHz hsync
Modeline "1024x768" 115.5 1024 1056 1248 1440 768 771 781 802 -HSync
-VSync
```

La définition explicite du "Modeline" n'est plus requise depuis les versions de 4.x ;-). Il fallait parfois éditer à la main pour obtenir les valeurs optimales dans des versions antérieures de XFree86, cela n'est généralement plus nécessaire avec v4.x. *Le guide pratique des synchronisations vidéo sous XFree86*, <http://www.linuxdoc.org/HOWTO/XFree86-Video-Timings-HOWTO/index.html> en propose une explication intéressante, mais assez technique.

Si l'utilitaire de configuration que vous employez, quel qu'il soit, n'identifie pas automatiquement votre carte vidéo ou les caractéristiques de votre écran correctement, vous avez peu de chances d'obtenir une configuration optimale. Dans ce cas, il vous faudra peut-être entrer manuellement les valeurs correctes. Ca devrait se trouver dans votre manuel de l'utilisateur (que vous avez gardé, hein ?). Sinon, jetez un oeil au site Web du fabricant.

Encore une fois, l'édition manuelle du fichier est généralement inutile. Si vous pensez qu'il faut tout de même le faire, faites attention. Une toute petite erreur peut faire planter X. Toute modification de ce fichier nécessitera de redémarrer X pour qu'elle prenne effet.

Utiliser le fichier du `XFree86Config` de quelqu'un d'autre, est généralement une mauvaise idée puisqu'il est peu probable qu'il ait exactement le même matériel que vous.

2.3. xvidtune et le réglage des fréquences de l'écran

Vous voulez probablement obtenir le meilleur de votre matériel. Si X n'est pas configuré de façon optimale, pensez à réexécuter l'utilitaire de configuration de X de votre fournisseur pour essayer d'obtenir de meilleurs résultats. Il est très peu probable que vous puissiez endommager quoi que ce soit en faisant des essais. La plupart des écrans modernes ont maintenant des protections qui permettent d'éviter la fusion/l'accident nucléaire ; -)

Si vous en faites un peu trop cependant, X peut ne pas pouvoir démarrer. Pour cette raison, je préfère employer la commande "startx" pour démarrer X (voir ci-dessous) pendant les "essais". De cette façon si X plante, le gestionnaire d'affichage (connexion de l'interface graphique) ne fera pas une boucle en vous donnant des maux de tête épouvantables. La commande **startx** va juste tranquillement retourner à un écran de console texte, où un message d'erreur sera affiché.

Une autre manière de peaufiner les paramètres associés à cet écran est d'utiliser le programme **xvidtune** de XFree86. Il est interactif et peut être utilisé pour ajuster divers paramètres (voir la page de man). La boîte de dialogue simple possède des glisseurs et boutons qui permettent à l'utilisateur d'entrer des paramètres et de les régler. La partie supérieure gauche permet de faire le paramétrage horizontal de l'écran, et celle de droite le paramétrage vertical. Les boutons juste au-dessous des glisseurs peuvent être utilisés pour les régler. .

On l'utilise parfois pour ajuster la zone d'écran visualisable, par exemple pour la centrer, ou augmenter sa taille jusqu'à la bordure de l'écran. Quand le **xvidtune** est lancé, il prend par défaut les paramètres en cours.

Le coin inférieur gauche possède les boutons nécessaires pour appliquer ["Apply"] de nouveaux paramètres, pour tester ["Test"] de nouveaux paramètres, ou pour montrer ["Show"] les paramètres (c.-à-d. les afficher à l'écran), entre autres. Aucun changement fait ici n'est enregistré. Si les nouveaux paramètres sont "appliqués", ils le sont juste pour la session courante. Exemple de résultat donné par "Show" de **xvidtune** :

```
Vendor: Unknown, Model: Unknown
Num hsync: 1, Num vsync: 1
hsync range 0: 30.00 - 70.00
vsync range 0: 55.00 - 120.00
"1400x1050" 122.00 1400 1488 1640 1880 1050 1052 1064 1082 +hsync
+vsync
```

La dernière ligne est le "Modeline" employé pour piloter l'écran courant. Voir *Le guide pratique des*

synchronisations vidéo sous XFree86, <http://www.linuxdoc.org/HOWTO/XFree86-Video-Timings-HOWTO/index.html>, pour en savoir plus sur les “Modelines”.

Vous pouvez tester les modifications, et les appliquer à la session courante. Pour que les modifications deviennent permanentes, elles devront être ajoutées manuellement dans la section “Monitor” de `XF86Config` (ou de `XF86Config-4` pour v.4.x) avec un éditeur de texte.

xvidtune va dûment vous avertir des risques liés au jeu avec les paramètres de l'écran. Il est peu probable que vous endommagiez quelque chose avec les écrans modernes. Mais il est préférable de ne l'utiliser que pour faire des ajustements mineurs. A utiliser à vos risques et périls !

3. Faire tourner X

Le démarrage d'une session de X peut se faire de deux manières : soit on démarre la session de X via un gestionnaire d'affichage (comme **xdm**), et l'utilisateur se connecte via un écran d'interface graphique. Soit l'utilisateur démarre X manuellement en se connectant via une console texte, ce qui se fait, de manière classique, avec la commande **startx**, qui est un script shell simple contenant un **xinit**. X tourne avec les droits de superutilisateur dans les deux cas, puisqu'il a besoin de l'accès direct au matériel.

Il faut savoir que le choix de la méthode utilisée dépend du système “runlevel”. Le runlevel par défaut qui se lance au démarrage se trouve généralement dans `/etc/inittab` sous Linux :

```
# Lancer xdm en runlevel 5
x:5:respawn:/etc/X11/xdm ~nodaemon
```

Cela lancera **xdm**, et donc X, avec le runlevel 5. Il redémarrera s'il meurt ou s'arrête quelle qu'en soit la raison. On peut également utiliser la commande **init** pour changer de runlevel sans redémarrer (voir la page de man).

Voyons brièvement les deux approches, puis un peu de configuration supplémentaire pour paramétrer l'environnement de travail de l'utilisateur.

3.1. startx

startx démarrera X en appelant d'abord **xinit**. En soit, cette action vous donnera un bureau vide, flou et réduit au strict minimum sans qu'aucun gestionnaire de fenêtres n'ait été chargé. A la base, **xinit** prend deux ensembles d'arguments en ligne de commande : les caractéristiques du client (les programmes qui

vont être lancés, etc.), et les caractéristiques du serveur (les options du serveur X), séparés par "--". Si aucun programme de client n'est indiqué dans la ligne de commande, **xinit** cherchera un fichier `.xinitrc` dans le répertoire local de l'utilisateur, à exécuter comme script shell. S'il le trouve, celui-ci, à son tour, exécutera les commandes personnalisées de l'utilisateur nécessaires pour configurer son environnement, ou lancera les programmes que le fichier contient. Si ce fichier n'existe pas, **xinit** emploiera la commande initiale suivante :

```
xterm -geometry +1+1 -n login -display :0
```

Si aucun `.xserverrc` n'est trouvé dans le répertoire local de l'utilisateur, X lui-même sera lancé par la commande suivante :

```
X :0
```

Comme vous voyez, ce n'est pas très utile car il ne lance qu'un seul `xterm`. **startx** fournit plus de fonctionnalité et de souplesse à **xinit**. **startx** appellera **xinit** pour nous, et fournira également quelques options simples de configuration. Vous pouvez également lancer des commandes telles que la suivante, par exemple :

```
startx -- -dpi 100 -depth 16 #pousse X à 100 points par pouces  
                             #et la profondeur de couleur à 16 (syntaxe X  
v4)
```

Tout ce qu'il y a après les doubles tirets est passé comme argument directement au serveur X par l'intermédiaire de **xinit**. Dans cet exemple, vous pouvez forcer X à la résolution de votre choix, mais il continuera quand même à utiliser les fichiers de configuration dont nous parlerons plus tard dans ce document. Voir la page de man de Xserver pour plus d'options en ligne de commande.

Au lieu de lancer la même ligne de commande à chaque fois, il est plus facile d'utiliser les fichiers de configuration pour qu'ils nous stockent ce type d'information.

Si on jette un coup d'oeil au script **startx** (`/usr/X11R6/bin/startx` sur mon système), on voit qu'il emploie deux fichiers de configuration par défaut pour permettre d'établir l'environnement de X : `xinitrc` et `xserverrc`. Il regarde d'abord dans `/etc/X11/xinit/`, pour les gros fichiers systèmes. Il cherche ensuite dans le répertoire local de l'utilisateur des fichiers semblables, qui seront prioritaires s'il les trouve. Notez que ces derniers sont des fichiers “point” de type Unix (par exemple `~/.xinitrc`), et sont des scripts shell exécutables.

Normalement on n'aura pas besoin d'éditer les gros fichiers systèmes, mais si on veut on peut les copier dans le répertoire local comme point de départ, ou simplement démarrer à partir de zéro. Comme vous pouvez le deviner grâce aux noms, l'un permet de configurer le serveur X, et l'autre configure **xinit** en exécutant des commandes, en préparant l'environnement et éventuellement en lançant des programmes client comme **xterm** ou un gestionnaire de fenêtres (oui, c'est un client également).

3.1.1. xserverrc

Comme avec tous les fichiers de configuration de XFree86, c'est un fichier texte. Il s'agit habituellement d'une simple ligne de commande pour mettre en marche le serveur X. Il peut contenir n'importe quelle option en ligne de commande valide acceptée par votre installation de X. Si vous démarrez toujours X avec vos propres options, ça devrait être plus facile que de taper les options à chaque fois. Voici un `~/.xserverrc` possible :

```
exec X :0 -dpi 100 -nolisten tcp
```

Ca démarrera X sur le display 0, le premier “affichage”, avec une résolution de 100 points par pouces et désactive les connexions TCP. Voir la page de man de Xserver pour d'autres options valides. Ceci n'est qu'un exemple.

3.1.2. xinitrc

`xinitrc` s'emploie pour établir un environnement X approprié, et pour lancer d'autres programmes, qu'on appelle aussi des “clients” que l'on souhaitera peut-être avoir disponibles dès le démarrage de X. Vous avez probablement un fichier système `xinitrc` pour démarrer un ensemble prédéfini de programmes. Pour personnaliser, créez votre propre fichier dans votre répertoire local. Nommez le `.xinitrc`, assurez-vous que c'est un script exécutable, et faites **chmod +x**. Un exemple (à partir de l'original sur mon système, légèrement modifié) :

```
#!/bin/sh
# $XConsortium: xinitrc.cpp,v 1.4 91/08/22 11:41:34 rws Exp $

userresources=$HOME/.Xresources
usermodmap=$HOME/.Xmodmap

# fusionne les événements de touche par défaut avec les personnalisées
if [ -f $userresources ]; then
    xrdp -merge $userresources
fi

if [ -f $usermodmap ]; then
    xmodmap $usermodmap
fi

if [ -z "$BROWSER" ] ; then
    # nous avons besoin d'un navigateur sur le système
    BROWSER=`which netscape`
    if [ -z "$BROWSER" ] || [ ! -e "$BROWSER" ] ; then
        # pas encore trouvé
        BROWSER=
    fi
fi

if [ -z "$BROWSER" ] ; then
    # nous avons besoin d'un navigateur sur le système
    BROWSER=`which lynx`
    if [ -z "$BROWSER" ] || [ ! -e "$BROWSER" ] ; then
        # pas encore trouvé
        BROWSER=
    else
        BROWSER="xterm -font 9x15 -e lynx"
    fi
fi

export BROWSER

# Démarre quelques programmes sympas
if [ -f $HOME/.Xclients ]; then
    exec $HOME/.Xclients
else
    xclock -geometry 50x50-1+1 &
    xterm -geometry 80x50+494+51 &
    if [ -f /usr/X11R6/bin/fvwm ]; then
        exec fvwm
    else
        exec twm
    fi
fi

#eof
```


Voici en deux mots ce que fait ce script : il installe notre environnement de travail, avec le **xmodmap** (clavier) et le **xrdb** (paramètres des ressources d'application). Nous en dirons plus un peu plus bas. Ensuite la variable `$BROWSER` du shell est affectée pour un environnement à interface graphique (Netscape dans cet exemple) de sorte que toutes les applications qui pourraient avoir besoin de celle-ci aient à leur disposition un choix raisonnable. Ensuite la présence du fichier `Xclients` est vérifiée, dans les dossiers systèmes ainsi que dans le répertoire local de l'utilisateur. Dans notre exemple, c'est dans ce fichier que toutes les applications clientes sont démarrées, y compris un gestionnaire de fenêtre (voir ci-dessous). Elles auraient pu tout aussi bien démarrer ici si nous l'avions voulu. Si un fichier `Xclients` ne peut être trouvé, un gestionnaire de fenêtre est démarré pour nous. **fvwm** si disponible, sinon **twm**, le gestionnaire de fenêtre minimaliste de XFree86. Si pour une raison ou pour une autre, aucun des deux ne pouvait être démarré, le script se fermerait, et X ne démarrerait pas.

3.1.3. Xclients

Jusque là, le déroulement de l'ensemble des opérations s'est effectué d'une façon tout à fait classique et prévisible. Pour récapituler, nous avons appelé **startx**, qui a à son tour appelé **xinit**, qui a analysé le `xinitrc` pour les paramètres initiaux. La plupart des Linux devraient suivre cet ordre-là, bien que les différents valeurs et paramètres puissent différer.

Nous sommes maintenant au dernier maillon de la chaîne où l'utilisateur aura normalement indiqué ses préférences, y compris le gestionnaire de fenêtres et/ou l'environnement graphique à utiliser. Le système fournira par défaut des scripts raisonnables, bien que probablement inintéressants, si l'utilisateur ne l'a pas fait. C'est sans doute pourquoi vous êtes ici ; -)

Le gestionnaire de fenêtres, ou environnement graphique, est tout le temps l'application qui démarre en dernier. Si vous voulez qu'il y ait d'autres programmes à démarrer (comme **xterm**), ils doivent être démarrés avant le gestionnaire de fenêtres et mis en arrière-plan avec un "&". Ceci peut être fait dans le `~/.xinitrc` de l'utilisateur. Ou bien, comme dans l'exemple ci-dessus, les applications en question sont démarrées à partir d'encore un autre script. En voici un court à titre d'exemple, `.Xclients` :

```
#!/bin/bash
# ~/.Xclients, démarre mes programmes

xset s off s noblank
xset m 30/10 4
xset r rate 200 40

xscreensaver &
rxvt -geometry 80x50-50+150 &
```

```

echo Démarrage du gestionnaire de fenêtres...

if [ -x /usr/X11R6/bin/wmaker ]; then
    echo `date` : Essai de /usr/X11R6/bin/wmaker... |tee -a ~/.wm-errors 2>&1
    exec /usr/X11R6/bin/wmaker >> ~/.wm-errors 2>&1
fi

echo `date`: Echec, essai de fvwm... |tee -a ~/.wm-errors 2>&1

# Essayons maintenant le bon vieux fvwm (AnotherLevel ne fonctionne pas avec
fvwm1).
if [ -n "$(type -path fvwm)" ]; then
    # si ça marche, nous arrêtons ici
    exec fvwm >> ~/.wm-errors 2>&1
fi

echo `date`: Echec, essai de twm... |tee -a ~/.wm-errors 2>&1

# Bon, fvwm n'est pas là non plu...
# Utilisation de twm en dernier recours.
exec twm >> ~/.wm-errors 2>&1

# Mort dans l'oeuf, X ne se lancera pas... pffff !
echo `date`: Impossible de lancer le gestionnaire de fenêtres... |tee -a
~/.wm-errors 2>&1

# eof

```

Ca ne change pas beaucoup de ce que faisait `xinitrc`. Nous avons ajouté quelques gadgets, parmi lesquels le démarrage d'un écran de veille, un émulateur de console différent que cet utilisateur préfère (`rxvt`), avec encore plus de paramétrages de l'environnement (moniteur, souris et clavier) en utilisant `xset` cette fois, et un gestionnaire de fenêtres différent de celui qui était disponible avec les paramètres par défauts du système. Ce fichier étant dans le répertoire local de l'utilisateur, il ne sera pas écrasé pendant les mises à jour.

En fait, X est déjà démarré à ce stade. Nous nous contentons de mettre la dernière main à la configuration. Notez que le gestionnaire de fenêtres n'est pas mis en arrière plan avec “&” ici. C'est important ! Quelque chose doit être au premier plan, sinon X se fermera. Nous n'avons pas démarré d'environnement graphique comme KDE ou GNOME, dans cet exemple, mais si nous le faisons, là cette application finale devrait être **gnome-session** ou **startkde**. Puisque nous utilisons notre propre script ici, si nous voulions changer de gestionnaire de fenêtres, la seule chose à faire serait d'éditer ce fichier, et de redémarrer X. Les configurations données par les fournisseurs peuvent être plus complexes que ça, mais les mêmes principes s'appliquent.

Pour terminer sur ce point, ne pensez pas que, au démarrage, toutes les applications du client *doivent*

être lancées comme nous l'avons fait ici. Ca, c'est l'approche traditionnelle, que certains peuvent préférer. La plupart des gestionnaires de fenêtres lancent à leur façon bien à eux les programmes au démarrage, ce que font KDE et GNOME. Voir les documentations respectives.

3.2. Gestionnaires d'affichage

L'autre approche, plus commune, est l'ouverture de session via interface graphique, où X fonctionne avant l'ouverture de session. Cela se fait avec l'aide d'un "gestionnaire d'affichage", dont il y a de nombreuses implémentations. XFree86 inclut le **xdm** (gestionnaire d'affichage de X) dans ce but, cependant votre distribution peut en employer un autre tel que le **gdm** (GNOME) ou **kdm** (KDE).

Les gestionnaires d'affichage en font beaucoup plus que ne le permettent les ouvertures de sessions via interface graphique. On s'en sert également pour contrôler aussi bien "l'affichage" local que distant sur un réseau. Nous n'entrerons pas dans les détails ici, mais le sujet est bien traité dans *Remote X Apps Mini HOWTO* et *XDMCP HOWTO* (voir la section des [liens](#)). Pour ce que nous voulons faire ici, ils fournissent des services similaires comme getty et login, qui permettent à des utilisateurs de se connecter dans un système et de démarrer leur shell par défaut, mais dans un environnement à interface graphique.

Voici un exemple d'utilisation plus avancée d'un autre emploi possible d'un gestionnaire d'affichage, proposé par Diego Zamboni :

J'ai deux sessions de X fonctionnant avec des résolutions différentes. Je passe de l'une à l'autre selon que mon ordinateur portable est relié à un moniteur externe ou qu'il utilise son propre écran à cristaux liquides.

Voici mon fichier `/usr/lib/X11/xdm/Xservers` qui lance les deux affichages :

```
:1 local /usr/X11R6/bin/X :1 -layout 1024x768
:0 local /usr/X11R6/bin/X :0 -layout 1600x1200
```

Ensuite je définis pour le serveur l'apparence suivante [ServerLayout] : "1024x768" et "1600x1200" dans mon fichier `/etc/X11/XF86Config-4`, comme suit :

```
Section "ServerLayout"
    Identifier      "1600x1200"
    Screen          "Screen0" 0 0
    InputDevice     "Mouse0" "CorePointer"
    InputDevice     "Keyboard0" "CoreKeyboard"
EndSection

Section "ServerLayout"
```

```

        Identifieur      "1024x768"
        Screen           "Screen1" 0 0
        InputDevice      "Mouse0" "CorePointer"
        InputDevice      "Keyboard0" "CoreKeyboard"
    EndSection

    Section "Screen"
        Identifieur      "Screen0"
        Device            "S3 Savage/MX"
        Monitor           "Monitor0"
        DefaultDepth      16

        Subsection "Display"
            Depth         16
            Modes         "1600x1200" "1280x1024" "1024x768"
        EndSubsection
    EndSection

?## et toc !

    Section "Screen"
        Identifieur      "Screen1"
        Device            "S3 Savage/MX"
        Monitor           "Monitor0"
        DefaultDepth      16

        Subsection "Display"
            Depth         16
            Modes         "1024x768" "800x600"
        EndSubsection
    EndSection

```

Notez l'utilisation des identifiants [Identifiers] ici. Diego démarre deux “affichagees” séparés ici. Il peut alors choisir lequel il veut lorsqu'il se connecte.

La plupart des gestionnaires d'affichage viennent du vénérable **xdm** de XFree86, avec en plus leurs propres perfectionnements. Voyons rapidement les plus répandus d'entre eux.

3.2.1. xdm

xdm peut être configuré à l'aide des fichiers de configuration situés dans `/etc/X11/xdm/`, `/usr/X11R6/lib/X11/xdm/`, ou à des emplacements semblables selon votre système. Ce sont de gros fichiers système. Le fichier `xdm-config` est le fichier principal de configuration, et décrit surtout où trouver les fichiers secondaires de configuration :

```

! $XConsortium: xdm-conf.cpp /main/3 1996/01/15 15:17:26 gildea $
DisplayManager.errorLogFile: /var/log/xdm-errors
DisplayManager.servers: /etc/X11/xdm/Xservers

```

```

DisplayManager.accessFile:      /etc/X11/xdm/Xaccess
! Tous les affichages devraient utiliser une autorisation
! mais ce n'est pas absolument certain.
! Les terminaux X seront configurés de cette manière, donc par défaut
! n'utiliser d'autorisation que pour les affichages locaux 0, 1, etc.
DisplayManager._0.authorize:    true
DisplayManager._1.authorize:    true
! Les trois ressources suivantes définissent l'affichage 0
! comme console
DisplayManager._0.setup:        /etc/X11/xdm/Xsetup_0
DisplayManager._0.startup:      /etc/X11/xdm/GiveConsole
DisplayManager._0.reset:        /etc/X11/xdm/TakeConsole
!
DisplayManager*resources:       /etc/X11/xdm/Xresources
DisplayManager*session:         /etc/X11/xdm/Xsession
!
! SECURITE : Ne tenez pas compte d'éventuelles requêtes de XDMCP ou de
Chooser
! Commentez cette ligne si vous voulez gérer les terminaux X avec xdm
DisplayManager.requestPort:     0

```

“!” signale les commentaires. La commande qui démarre le serveur X est située dans `/etc/X11/xdm/Xservers` dans cet exemple précis comme défini par “`DisplayManager.servers`”, et est l'équivalent de `xserverrc` qui a été employé pour démarrer le serveur X avec **startx**, mais la syntaxe est légèrement différente ici. Le contenu du fichier `/etc/X11/xdm/Xservers` sur mon système est simplement :

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

Cela démarre X sur le premier affichage local (désigné par 0). C'est ici, à la fin, qu'il faut mettre les options en ligne de commande que vous souhaiteriez ajouter.

Ci-dessous un exemple du fichier `/etc/X11/xdm/Xsetup_0` qui ne sert que pour configurer l'écran d'ouverture de session. Notez que nous utilisons un script shell, et qu'il appelle **xv** (un programme d'affichage graphique) pour avoir comme arrière-plan une belle image (à la place du triste arrière-plan noir et blanc), et si cela échoue, le **xsetroot** est alors appelé pour essayer au moins d'avoir comme arrière-plan une couleur bleue plus jolie. Ceci ne configure pas l'outil d'ouverture de session lui-même - mais juste d'autres choses que l'on souhaiterait avoir sur l'écran pendant l'ouverture de session.

```
#!/bin/sh
xconsole -geometry 480x100-0-0 -daemon -notify -verbose -fn \
  '-schumacher-clean-medium-r--*-10-*-*-*-*-*-*' -exitOnFail &

/usr/X11R6/bin/xv -quit -root /usr/share/pixmaps/Backgrounds/InDreams.jpg \
  || xsetroot -solid darkblue
```

Le fichier `/etc/X11/xdm/Xresources` contrôle les “ressources” de X utilisées pendant l'ouverture de session. Dans ce contexte, les “ressources” sont des préférences de l'utilisateur portant sur certains éléments comme la police ou les couleurs (décrites avec plus de détails ci-dessous). Ci-dessous un bout de code qui paramètre les polices pour l'outil d'ouverture de session :

```
#if WIDTH > 800
xlogin*greetFont: -adobe-helvetica-bold-o-normal--24-240-75-75-p-138-
iso8859-1
xlogin*font: -adobe-helvetica-medium-r-normal--18-180-75-75-p-103-iso8859-1
xlogin*promptFont: -adobe-helvetica-bold-r-normal--18-180-75-75-p-103-
iso8859-1
xlogin*failFont: -adobe-helvetica-bold-r-normal--18-180-75-75-p-103-iso8859-
1
#else
xlogin*greetFont: -adobe-helvetica-bold-o-normal--17-120-100-100-p-92-
iso8859-1
xlogin*font: -adobe-helvetica-medium-r-normal--12-120-75-75-p-69-iso8859-1
xlogin*promptFont: -adobe-helvetica-bold-r-normal--12-120-75-75-p-69-
iso8859-1
xlogin*failFont: -adobe-helvetica-bold-o-normal--14-140-75-75-p-82-iso8859-1
#endif
```

Comme vous pouvez le voir helvetica est utilisé comme police préférée, avec différentes tailles de point et de dpi (points par pouce) selon la taille d'écran. C'est personnalisable pour convenir à tous les besoins. (Voir ci-dessous pour en savoir plus sur les conventions concernant les noms de polices dans X). Divers autres aspects peuvent également être configurés.

`/etc/X11/xdm/Xsession` est à peu près l'équivalent du `xinitrc` pour le **startx**. Il établira également un environnement par défaut pour le clavier, etc. Il peut également démarrer KDE ou GNOME, ou tout autre programme client de X. C'est le fichier principal de configuration du système. Il doit également

vérifier le répertoire local de l'utilisateur pour trouver un fichier `~/.xsession`, et un éventuel fichier `~/.Xclients`, qui contiendrait l'environnement et les programmes de démarrage préférés de l'utilisateur, tout comme `~/.xinitrc` le faisait avec **startx**. Encore une fois, les fichiers dans le répertoire local de l'utilisateur peuvent être créés ou modifiés par l'utilisateur n'importe quand et doivent être des scripts shell exécutables.

Nous n'allons pas proposer d'exemple de fichier `~/.xsession`, étant donné qu'il ressemblerait beaucoup aux exemples de fichiers `~/.xinitrc` et `~/.Xclients` présentés ci-dessus.

Nous avons vu rapidement les principaux fichiers de configuration de **xdm**. N'hésitez pas à lire la page de man, et à regarder ce qui est installé localement, pour plus d'informations. Voyons maintenant **gdm** et **kdm**. Nous surlignerons juste les différences significatives, étant donné qu'ils fournissent grosso modo les mêmes fonctionnalités.

3.2.2. gdm

gdm est le gestionnaire d'affichage par défaut pour GNOME. **gdm** a été écrit à partir de rien, mais fonctionne comme **xdm**. Le fichier principal de configuration est `gdm.conf` ; il se trouve en général ici : `/etc/X11/gdm/gdm.conf`. Il ne ressemble pas visuellement au fichier `xdm-config`. Les commentaires sont précédés par un "#", et le fichier a des sections, avec des entêtes de section entre crochets. La commande pour démarrer X est dans la section "[servers]" :

```
[servers]
0=/usr/bin/X11/X
#1=/usr/bin/X11/X
```

Notez qu'il y a potentiellement deux affichages de paramètres, mais le second est commenté. On ajoutera les éventuelles options de démarrage supplémentaires de X ici, par exemple "-dpi 100". L'écran d'ouverture de session et l'outil d'ouverture de session sont configurés dans la section "[greeter]".

Les clients et les programmes lancés au démarrage sont déterminés par le bloc "SessionDir" dans la section "[daemon]". Sur mon installation, celui-ci pointe sur `/etc/X11/gdm/Sessions/`, qui contient plusieurs scripts courts. Si je regarde mon script par défaut (Default), je vois en fait qu'il exécute `/etc/X11/xdm/Xsession`, qui à son tour exécutera `~/.xsession`, s'il est présent. Ainsi à cette dernière étape, ce que fait **gdm** est tout à fait semblable à ce que fait **xdm**.

GNOME fournit l'utilitaire **gdmconfig** pour gérer de nombreux aspects du comportement de **gdm**.

3.2.3. kdm

kdm est le gestionnaire d'affichage de KDE. Le fichier principal de configuration pour **kdm** est `kdmrc` et il se trouve généralement ici : `/etc/kde/kdm/kdmrc`. Comme cela est le cas pour `gdm.conf`, `kdmrc` utilise “#” pour signaler les commentaires, et a des sections avec des entêtes entre crochets également. La configuration **kdm** peut également être éditée avec l'utilitaire **kcontrol**.

Le bureau est configuré dans la ou les section(s) “[Desktop*]”, et par la directive “setup” qui doit pointer vers un fichier comme `/usr/share/config/kdm/Xsetup` ou `/etc/X11/xdm/Xsetup_0`. On aboutira à la même chose qu'avec le `Xsetup_0` de **xdm** : à savoir l'exécution de tous les programmes que l'utilisateur aimerait éventuellement avoir, comme **xconsole**.

La commande pour lancer le serveur X est la directive “Xservers” dans “[General]”. Encore une fois, cela doit pointer vers un fichier tel que `/etc/X11/xdm/Xservers`, et utilisera la même syntaxe que **xdm** :

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

C'est ici que toute option en ligne de commande pour le serveur X doit se mettre.

L'outil d'ouverture de session lui-même est configuré dans la (ou les) section(s) “[X*-Greeter]”. Une version compilée par défaut est utilisée si l'utilisateur n'en indique pas.

KDE fournit l'utilitaire **kdmdesktop** pour gérer divers aspects du comportement de **kdm**, principalement l'arrière-plan d'ouverture de session.

4. Pour en savoir plus sur la configuration de X

Avant de jeter un oeil à divers mécanismes de configuration pour les serveurs et les clients X, il faut remarquer que les environnements graphiques comme KDE se sont répandus dès leur apparition en partie parce qu'ils peuvent gérer la plupart des configurations d'interactions utilisateurs, grâce à des commandes “faciles à utiliser” par interface graphique. Et en fait, les applications conformes qui font partie des bureaux respectifs seront mieux configurées par les outils de la configuration du bureau, ou par leurs propres méthodes de configuration graphique. Ainsi, par exemple, le **gtop**, une application cliente de GNOME, se configure mieux par l'intermédiaire de GNOME ou des menus **gtop** eux-mêmes. Mais ce n'est pas vrai pour toutes les applications de X.

4.1. Ressources X

Le serveur X peut stocker diverses valeurs de configuration pour les programmes clients de sorte qu'elles sont facilement disponibles si nécessaires. Si l'application est compatible, elle les emploiera en tant que valeurs par défaut à chaque fois que ce programme sera appelé. Ces valeurs sont connues sous le nom de “ressources [Resources]”, et servent souvent pour définir les préférences utilisateurs, application par application, pour les polices, les couleurs, la position à l'écran (la géométrie) et divers autres attributs. Cela facilite la personnalisation des applications.

Les ressources sont de type chaîne de caractères (par exemple `Netscape*blinkingEnabled:False`) qui peut être lue à partir du disque dans divers endroits quand X se lance ; elles peuvent aussi être définies interactivement en ligne de commande. Les composants des programmes sont nommés hiérarchiquement, chaque objet dans la hiérarchie étant identifié par une classe ainsi que par un nom d'instance. Au niveau supérieur de la hiérarchie se trouvent la classe et le nom de l'instance de l'application elle-même. Le nom de la classe de l'application est toujours identique au nom du programme, mais avec la première lettre en majuscule (par exemple Vim ou Emacs) bien que quelques programmes qui commencent par la lettre “X” ont également leur deuxième lettre en majuscule pour des raisons historiques (par exemple XTerm). Chaque définition indiquera une classe (ou instance), avec la ressource et la valeur qui correspondent. Plus bas dans la hiérarchie se trouvent les divers attributs qui constituent les aspects paramétrables de l'application.

Traditionnellement la plupart des programmes X étaient configurés de la sorte. Ce n'est plus aussi vrai aujourd'hui avec l'arrivée des environnements graphiques qui ont souvent leurs propres mécanismes de configuration.

Par exemple, supposons que nous préférons lancer `xterm` avec un arrière-plan bleu. Donc si nous le lançons depuis la ligne de commande, nous le ferions ainsi :

```
xterm -bg blue &
```

Si c'est notre préférence, il serait plus facile de mettre cette préférence dans un fichier quelque part, et faire utiliser notre préférence par le système. De cette façon chaque fois que nous démarrerions `xterm`, il utiliserait notre valeur préférée, et nous n'aurions plus besoin de la ligne de commande options (à moins de vouloir surcharger).

La syntaxe de base des ressources de X s'écrit comme suit :

```
<program><binding><widget><binding><widget><...><resource>:<value>
```

Ce qui, dans la vraie vie, ressemble en général à ça :

```
xterm*fontMenu*background: darkblue
```

Ce que cela fait devrait paraître évident. L'utilisation des “*” dans la définition s'appelle “liaison lâche [loose binding]” et joue le rôle de joker. Cela signifie qu'il peut y avoir des lacunes dans la hiérarchie des composants.

```
xterm*background: darkblue
```

Ceci donnerait également un fond bleu foncé pour le fontMenu de **xterm**, mais également pour toutes les autres propriétés de **xterm** qui ont également un attribut “background” (par exemple window background, etc.), peu importe où ils peuvent être dans la hiérarchie des composants. De même :

```
*background: darkblue
```

Ceci définirait le fond pour tous les programmes qui ont cette propriété (pas uniquement **xterm**). L'emploi d'un “.” au lieu d'une “*” sera plus précis, et permettra d'écarter les lacunes de joker dans la hiérarchie. En outre, l'application doit accepter l'attribut particulier du composant. Pour “Background”, c'est à peu près sûr, mais beaucoup d'applications auront des ressources plus spécialisées qui ne sont pas aussi évidentes. Il est préférable de regarder la documentation locale (pages de man, etc.), ou regarder si une application a des exemples d'inclus. Par exemple, **Netscape** est généralement livré avec un fichier `Netscape.ad` qui a un ensemble très complet de définitions de ressource qui peuvent être adaptées.

On verra que les ressources de X sont stockées à plusieurs endroits (voir ci-dessous) et sont traitées par la commande **xrdb** (voir la page de man).

4.1.1. Les fichiers App Defaults

Une méthode pour stocker les ressources des applications préférées est d'utiliser les fichiers nommés pour l'application dans un dossier "app-defaults". Par exemple, sur mon système, ceux-ci se trouvent dans `/usr/X11R6/lib/X11/app-defaults/`, cependant ce répertoire peut être différent selon les options que votre fournisseur a choisies. Ce répertoire contient un certain nombre de fichiers pour les applications X connues telles que le **xterm**, **xclock**, **xcalc**, **xload**, etc. Au final, c'est un nombre relativement petit d'applications au vu du système dans son ensemble. Par conséquent, toutes les applications n'utilisent pas cette méthode. En fait, la majorité ne l'utilise pas.

Chaque fichier contiendra la définition des ressources pour cette application. Le serveur X charge ceux-ci de lui-même pendant son démarrage. En court exemple tiré de `xTerm-color` :

```
! $XFree86$

#include "XTerm"

*VT100*colorMode: on
*VT100*dynamicColors: on

! A décommenter pour utiliser de la couleur pour l'attribut "underline"
!*VT100*colorULMode: on
!*VT100*underLine: off

! A décommenter pour utiliser de la couleur pour l'attribut "bold"
!*VT100*colorBDMode: on

*VT100*color0: black
*VT100*color1: red3
*VT100*color2: green3
*VT100*color3: yellow3
*VT100*color4: blue3
*VT100*color5: magenta3
*VT100*color6: cyan3
*VT100*color7: gray90
*VT100*color8: gray30
*VT100*color9: red
*VT100*color10: green
*VT100*color11: yellow
*VT100*color12: blue
*VT100*color13: magenta
*VT100*color14: cyan
*VT100*color15: white
```

```
*VT100*colorUL: yellow
*VT100*colorBD: white
```

Ici on a surtout différentes définitions de couleur. Le classname de l'application n'est pas explicitement énoncé, et est déterminé par le nom de fichier. Il faut donc faire comme si chaque ligne commence par `XTerm-color*`. En outre, remarquez en haut, la ligne `#include "XTerm"`, qui “inclut” les définitions de ressource pour **XTerm**, fichier beaucoup plus long avec un ensemble de définitions plus divers. (On ne l'a pas inclus ici du fait de sa longueur, mais ça vaut le coup de regarder.) Ces fichiers fournissent les paramètres par défaut du système, et d'une manière générale, sont rarement édités par l'utilisateur.

4.1.2. Xdefaults

Une autre méthode couramment pratiquée pour l'obtention des préférences au niveau ressources, est d'utiliser un fichier `Xdefaults`, nommé parfois `Xresources`. Celui-ci peut exister comme fichier système, par exemple `/etc/X11/Xresources`. Naturellement, l'utilisateur est libre de créer une version personnelle dans son répertoire local, par exemple `~/.Xdefaults`. La version de l'utilisateur surchargera tout paramètre des fichiers système, et restera inchangé après les mises à niveau du système. Evidemment, c'est là que vous mettrez vos propres préférences.

Les fichiers `Xresources` sont lus pour la base de donnée des ressources avec la commande **xrdb**. Par exemple:

```
xrdb -merge ~/.Xresources
```

Ceci peut être fait interactivement à la ligne de commande, ou être placé dans un script et être lancé automatiquement au démarrage de la session X. Dans le cas des fichiers système, ceci devrait être pris en charge par les scripts de démarrage fourni par la distribution. Généralement, ces derniers vérifieront également le répertoire local de l'utilisateur (voir l'exemple de `xinitrc` ci-dessus). Du coup, on aura probablement rien d'autre à faire que de créer le fichier avec un éditeur de texte.

Voici un exemple pour vous montrer une infime partie de tout ce qu'on pourrait faire avec un fichier `.Xdefaults`:

```
! Ceci est un commentaire ;-)  
  
#ifdef COLOR  
*customization: -color  
#endif  
  
!! Ratissons large, pour récupérer toutes les applications compatibles  
!! Signal visuel, et non sonore  
*visualBell: True  
*scrollTtyOutput: False  
*scrollKey:      True  
  
! Voir Netscape.ad pour plus de ressources paramétrables  
Netscape*noAboutSplash:      True  
Netscape*documentFonts.sizeIncrement: 5  
Netscape*documentFonts.xResolution*iso-8859-1: 120  
Netscape*documentFonts.yResolution*iso-8859-1: 120  
netscape-navigator*geometry: 960x820+240+140  
  
emacs*Background: DarkBlue  
emacs*Foreground: Wheat  
emacs*pointerColor: Orchid  
emacs*cursorColor: Orchid  
emacs*bitmapIcon: on  
emacs*font: 10x20  
  
! Couleurs pour GVim, etc.  
!! Les versions GTK de gvim n'utiliseront pas ces paramètres.  
Vim*useSchemes:      all  
Vim*sgiMode:         true  
Vim*useEnhancedFSB:  true  
Vim.foreground:      Black  
!Vim.background:     lightyellow2  
Vim*background:      white  
! geometry: largeur x hauteur  
Vim.geometry: 88x40  
Vim*font: -misc-fixed-medium-r-normal--20-200-75-75-c-100-iso8859-15-*5  
Vim*menuBackground: yellow  
Vim*menuForeground: black  
  
rxvt*backspacekey: ^?  
rxvt*background: Black  
rxvt*foreground: wheat  
rxvt*cursorColor: Orchid  
rxvt*geometry: 100x18+40+300  
rxvt*title: Linux  
rxvt*reverseVideo: false  
!rxvt*backgroundPixmap: ~/penguinitis.xpm  
rxvt*scrollBar: true  
rxvt*reverseWrap: true  
rxvt*font: -*lucidatypewriter-medium-*-*-*14-*-*-*-*-*  
rxvt*fullCursor: true  
rxvt*saveLines: 1500
```

```

rxvt*menu: ~/rxvt.menu

XTerm*saveLines: 1500
! Ne pas effacer l'écran après la fermeture du programme
XTerm*VT100*titeInhibit: true

! Corrige les affectations de touche pour xterm
xterm*VT100.translations: #override \
    <Key>BackSpace: string(0x7F) \n\
    <Key>Insert: string(0x1b) string("[2~")\n\
    <Key>Delete: string(0x1b) string("[3~")\n\
    <Key>Home: string(0x1b) string("[1~")\n\
    <Key>End: string(0x1b) string("[4~")\n\
    <Key>Page_Up: string(0x1b) string("[5~")\n\
    <Key>Page_Down: string(0x1b) string("[6~")\n\
    <KeyPress>Prior : scroll-back(1,page)\n\
    <KeyPress>Next : scroll-forw(1,page)

! Ghostview
Ghostview*Font: *-helvetica-bold-r-normal--12-*-*-*-*-*
Ghostview*BorderColor: white
Ghostview*Text*Font: rk14
Ghostview*Background: #d9d9d9
!Ghostview*Foreground: white
ghostview.form.pageview.page.background: white
ghostview.form.pageview.page.foreground: black
.ghostview.zoom.form.page.background: white
.ghostview.zoom.form.page.foreground: black

! xscreensaver !
! Temps dépassé au bout de 12 minutes, mode cycle toutes les 2 minutes
xscreensaver.timeout: 12
xscreensaver.cycle: 5
! Tourner en priorité faible, et fondu entre les modes
xscreensaver.nice: 12
xscreensaver.fadeSeconds: 2

XFontsel.menu.options.showUnselectable: False

```

Si tout va bien, ces quelques exemples vous donneront quelques idées pour démarrer. X n'a pas besoin d'être redémarré si **xrdb** est utilisé interactivement en ligne de commande après des modifications. Les effets sont immédiats.

Les ressources sont parfois également disponibles comme options en ligne de commande. Voir ci-après. Les options en ligne de commande surchargeront toutes les définitions existantes de ressource.

4.2. xmodmap, le clavier et les souris

Le clavier et la souris, ainsi que d'autres périphériques d'entrée possibles, sont définis dans XF86Config (ou XF86Config-4). Il y a une disposition des touches du clavier qui est définie à partir de la langue préférée :

```
Section "InputDevice"
    Identifier "Keyboard0"
    Driver      "keyboard"
    Option      "XkbLayout" "us"
EndSection
```

Ceci nous donne notre disposition de clavier par défaut. Une liste des étiquettes valides de disposition se trouve dans `/usr/X11R6/lib/X11/xkb/symbols`. En outre, l'utilitaire `setxkbmap` peut être employé pour faire des modifications interactivement.

Avec X, les possibilités de personnalisation sont énormes, et on peut modifier les actions des touches du clavier et du pointeur de la souris à notre convenance. L'utilitaire pour ce faire est **xmodmap** (voir la page de man). Vous n'aimez pas l'emplacement de la touche de verrouillage majuscule ? Déplacez là !! ; -)

Comme **xrdb**, **xmodmap** peut être lancé en ligne de commande. Autre possibilité : les paramètres préférés peuvent être stockés dans un fichier. En général, c'est le fichier `~/.Xmodmap`, ou quelque chose de similaire. Si les scripts de démarrage de votre X n'analysent pas ces fichiers, faites les modifications qu'il faut pour qu'ils le fassent (probablement à partir de `~/.xinitrc` ou de `~/.xsession`).

On peut visualiser les actions du pointeur et des touches avec : **xmodmap -pk -pp |less**. Toutes les valeurs actives de "keycode" s'afficheront, avec les valeurs correspondantes de "keysym", et tous les noms de keysym que le **xmodmap** connaît (par exemple "retour arrière" [BackSpace]). Cela devrait également vous donner une idée de la façon dont le **xmodmap** comprend les événements touches et souris. Il y a deux "keysyms" par "keycode". La seconde est la valeur lorsque la "majuscule" est activée. L'utilitaire **xev** de XFree86 peut être employé pour afficher interactivement beaucoup d'informations sur les touches et les événements souris. Faites attention à la valeur de "keycode". C'est ce que vous devrez connaître afin de modifier les actions.

xmodmap est souvent utilisé pour faire des petites améliorations clavier, par exemple pour avoir des fonctions retour arrière/supprimer [BackSpace/Delete] bien adaptées. Il peut également être utilisé pour faire des modifications importantes comme les mappages internationaux. On peut seulement modifier

les mappages des touches et des événements souris -- on ne peut pas affecter des macros aux événements touches (il se pourrait que votre gestionnaire de fenêtre ou votre environnement graphique ait en partie cette fonctionnalité).

L'installation de claviers internationaux est bien expliquée dans <http://www.linuxdoc.org/HOWTO/mini/Intkeyb/index.html>. En outre, une recherche dans <http://google.com/linux> donnera beaucoup d'exemples utiles.

La page de man possède de nombreux exemples rapides sur diverses approches. Voici ce à quoi pourrait ressembler un `~/Xmodmap` hypothétique :

```
! /home/hal/.Xmodmap, dernier changement 03/10/01
!
! Définir le retour arrière à 22 et la touche "supprimer" à 111
keycode 22 = BackSpace
keycode 111 = Delete
!
! Mon clavier gère les touches Maj gauche et droite différemment.
! Faire que la droite agisse comme la gauche pour éviter la gymnastique
! des doigts.

keycode 63 = Alt_L
keycode 113 = Meta_L
!
! Coder en dur les valeurs des touches du pavé numérique en chiffres
! comme si le Verr Num était toujours activé
! puisque je ne l'utilise jamais pour autre chose
keycode 79=7
keycode 80=8
keycode 81=9
keycode 83=4
keycode 84=5
keycode 85=6
keycode 87=1
keycode 88=2
keycode 89=3
keycode 90=0
keycode 91=period
keycode 86 = plus
! Désactiver le Ver Num vu qu'on en a plus besoin
keycode 77 =
!
! Verr maj est près de tab. Je la tape par erreur de temps en temps,
! et ne l'utilise pas de toute façon. Donc faire Verr Num agir comme Tab.
keycode 66 = Tab
clear lock
```



```
!  
! Inversion des boutons de la souris pour les gauchers  
pointer = 3 2 1
```

Comme dans beaucoup de fichiers XFree86, “!” représente un commentaire. Une autre utilisation possible, est de redéfinir ces touches “Windows”, qui sont agaçantes, pour qu'elles servent à quelque chose. J'espère que cela donne une idée des choses qu'on pourrait faire, si on en a envie, pour avoir un clavier plus agréable.

En parlant de la touche `Verr Num`, X la neutralisera systématiquement lorsqu'il démarrera. Qu'importe la façon dont on a paramétré le BIOS, ou Linux avant que X se lance. Les indications données ci-dessus ne représentent donc qu'une façon de faire. Il y a également un utilitaire de disponible sur Internet qui s'appelle soit `numlockx`, soit `setnumlock`, si votre distribution ne comporte ni l'un ou ni l'autre. Il peut se mettre dans un fichier de démarrage pour activer `Verr Num` automatiquement, c'est comme on veut.

Les gestionnaires de fenêtres et les environnements bureau permettent également la personnalisation du clavier et de la souris (dans la mesure où X la reconnaît correctement). Cela peut être une méthode plus facile pour configurer certaines personnalisations.

4.2.1. Mappage des touches spéciales

Il y a de nombreux mappages de touches spéciales traditionnellement utilisés dans XFree86.

- `Ctrl-Alt-Retour Arrière` - Fermera le processus de serveur X d'une façon correcte. C'est une manière rapide, facile et légitime de redémarrer X. Notez qu'elle ne remet pas en marche le gestionnaire d'affichage (si utilisé) à seulement X lui-même.
- `Ctrl-Alt-Fx` - où “x” correspond à un chiffre valide d'un terminal (en général 1-6). C'est ce qu'on utilise pour ouvrir une console texte, alors que X est toujours en cours d'exécution. Pour revenir à X, faites “`Alt-Fx`”. Ici, “x” est égal au dernier terminal plus une unité (par exemple `Alt-F7` s'il y a six terminaux de disponibles).
- `Ctrl-Alt+` (du pavé numérique) et `Ctrl-Alt-` (du pavé numérique) - On passera d'un “mode” de résolution d'écran valide existante à un autre, par exemple `1024x768` -> `600x800`. Notez que la taille d'écran elle-même ne change pas : seuls la vue et la résolution changent. Pas si utile que ça pour la plupart des cas. Vous ne pouvez pas changer la résolution d'écran de manière permanente sans redémarrer X.

Il est possible que le gestionnaire de fenêtre, l'environnement graphique ou tout autre composant du système puisse attraper ces derniers, et en changer le comportement standard. En outre, `Ctrl-Alt-`

Supprimer peut également être attrapé. Celui-ci devrait fermer X (et le système), s'il est disponible.

4.2.2. Souris et pointeurs

Ainsi que cela a déjà été dit, Linux et Unix font une utilisation massive des trois boutons de la souris. Si une souris a seulement deux boutons, alors le troisième bouton (c.-à-d. celui du milieu) peut être simulé en appuyant sur les deux boutons en même temps. C'est une option de configuration qui se trouve dans `XF86Config` sous le nom de directive "Emulate3Buttons" :

```
Section "InputDevice"
    Identifier "Mouse0"
    Driver      "mouse"
    Option      "Device" "/dev/mouse"
    Option      "Protocol" "PS/2"
    Option      "Emulate3Buttons" "on"
EndSection
```

Au bout du compte, un troisième bouton est très pratique et je recommanderais personnellement d'en avoir un. Sur les souris à molette, la "molette" agit en tant que le troisième bouton, si on appuie dessus. Beaucoup de souris à molette standard semblent utiliser le protocole "IMPS/2".

Plus particulièrement, le troisième bouton (milieu) est, dans pratiquement toutes les applications Linux, le bouton pour "coller". Le copier/coller fonctionne un peu différemment sous Linux. Le bouton gauche est le bouton de copie. Maintenez le et sélectionnez le texte. Il est automatiquement copié dans le "presse-papiers" de X. Puis, le bouton du milieu collera à partir de ce dernier. C'est un processus très simple. En double-cliquant on copiera les mots un par un, et en triple-cliquant on copiera chaque ligne d'un texte. Si pour une raison ou pour une autre, cela ne fonctionne pas, c'est soit une application mal implémentée, soit un bogue quelconque. Certaines anciennes versions de Netscape posaient des problèmes, par exemple. Pour coller depuis le clavier, il faut faire maj+insert.

Le "glisser/déposer" n'est pas une fonction native de X lui-même. Mais, il est implémenté par certaines boîtes à outils et environnements graphiques. Il ne faut pas s'attendre à ce que ça marche avec des applications non conformes (c.-à-d. les applications non-KDE dans KDE par exemple).

4.3. xset

xset est encore un autre utilitaire XFree86 pour paramétrer des préférences utilisateur. **xset** est un peu un attrape-tout et est utilisé pour changer divers paramètres indépendants du serveur X. C'est surtout une méthode en ligne de commande pour configurer certaines des propriétés déjà définies dans `XF86Config` (mais pas tout !).

xset est couramment utilisé pour activer ou désactiver le DPMS ainsi que les intervalles préférés ; pour changer dynamiquement le `FontPath` ou le relire ; pour gérer les DEL du clavier ; pour régler la vitesse de mouvement de la souris (ou tout autre pointeur) ; pour paramétrer les taux de “ répétition ” et d’ “ autorépétition ” du clavier ; et pour gérer l'extinction d'écran intégré à X. Voir la page de man, naturellement, pour des explications plus détaillées, et d'autres utilisations de **xset**.

Encore un fois, **xset** peut être utilisé interactivement en ligne de commande. Mais, la plupart du temps, les paramètres préférés sont stockés dans un fichier de configuration du démarrage, comme `.xinitrc` ou `.xsession`. Un exemple très rapide :

```
# Désactiver l'extinction d'écran
xset s off

# Activer l'économie d'énergie par le DPMS
xset +dpms

# Peaufiner le rongeur...
xset m 30/10 4

# Accélérer le clavier
xset r rate 200 40
```

Votre environnement graphique a peut-être une interface graphique pour **xset**.

5. Polices et Couleurs

Comprendre les polices et les couleurs peut être plus difficile dans X que dans les autres systèmes.

5.1. Les polices démystifiées

X connaît divers types de police, y compris les bitmaps, les Type 1, et depuis la v4.x, les TrueType. Soit

le serveur X peut gérer les polices lui-même, soit parfois cette tâche est concédée à un serveur de police (il en existe un grand nombre). **xfs** (X Font Server) est le serveur de police le plus couramment utilisé sous Linux.

Un serveur de police n'est pas obligatoire, car X peut gérer la plupart des rendus de police. Les serveurs de police sont traditionnellement utilisés pour servir les polices de multiples hôtes sur un réseau, mais ils sont parfois également utilisés pour fournir des fonctionnalités supplémentaires. De plus, un serveur de police peut fournir une légère amélioration des performances du fait qu'il permet de transférer le rendu des polices à un processus séparé.

X reconnaît les polices en fonction des polices qui se trouvent dans le "FontPath". Ceci est paramétré initialement dans `XF86Config`. Si le serveur X gère lui-même les tâches liées aux polices (c.-à-d. sans serveur de police), on aura une liste de répertoires qui contiennent les fichiers de police, par exemple :

```
FontPath      "/usr/X11R6/lib/X11/fonts/misc:unscaled"
FontPath      "/usr/X11R6/lib/X11/fonts/100dpi:unscaled"
FontPath      "/usr/X11R6/lib/X11/fonts/75dpi:unscaled"
FontPath      "/usr/X11R6/lib/X11/fonts/Type1"
FontPath      "/usr/X11R6/lib/X11/fonts/misc"
FontPath      "/usr/X11R6/lib/X11/fonts/100dpi"
FontPath      "/usr/X11R6/lib/X11/fonts/75dpi"
```

Si un serveur de police est utilisé, le "FontPath" se dirigera vers l'interface de connexion [socket] que le serveur de police écoute (voici un exemple possible) :

```
FontPath "unix/:7101"
```

Dans ce dernier cas, les répertoires de police qui se trouvent être disponibles seront configurés dans le serveur de police (voir la documentation locale), qui emploiera un schéma semblable de type répertoire comme on l'a montré pour `XF86Config`.

Une fois que des polices appropriées ont été installées, elles doivent "être préparées". Pour la plupart des polices, ceci signifie lancer (en super utilisateur) l'utilitaire **mkfontdir** (voir la page de man) dans le répertoire où les polices sont situées. Les polices de type 1 et de TrueType nécessitent des étapes

supplémentaires (voir ci-dessous). Votre fournisseur l'a fait pour toutes les polices qui sont incluses dans votre distribution. Cela devra donc seulement être fait pour les polices que vous ajouterez. Pour que les polices nouvellement ajoutées deviennent visibles pour X, vous devrez lancer les commandes de **xset** appropriées pour soit modifier le FontPath existant, soit le relire (voir la page de man). Ou, réinitialiser votre serveur de police.

Exemple : Préparation des polices, et réinitialisation du serveur de police après avoir ajouté de nouvelles polices :

```
su
<mot de passe>
mkfontdir /usr/X11R6/lib/X11/fonts/my_new_fonts/
/etc/init.d/xfs restart
```

La première commande peut ne pas être nécessaire sur les distributions les plus récentes (puisque le script init le fait dans certains cas). Et la configuration de serveur de police devrait être modifiée, si c'est un nouveau répertoire. Exemple : réinitialisation sans serveur de police :

```
su
<mot de passe>
mkfontdir /usr/X11R6/lib/X11/fonts/my_new_fonts/
xset +fp /usr/X11R6/lib/X11/fonts/my_new_fonts/
xset fp rehash
```

Le “**xset +fp**” n'est pas nécessaire si le répertoire fait déjà partie du FontPath.

xlsfonts | **less** peut être utilisé pour faire la liste des polices connues, et donc disponibles, pour X et ses clients. Lancez **xlsfonts** | **less**, et vous pourrez également avoir une idée de la définition de police telle que l'entend X. Les ressources de police sont indiquées tout à fait explicitement, et cela peut sembler complexe au début. La description logique de police de X (*X Logical Font Description*, “XLFDF”) est la description complète pour n'importe quelle police donnée. Le XLFDF ressemble à ceci :

```
-adobe-helvetica-medium-r-normal-*-*120-*-*p*-iso10646-1
```

Où chaque champ de gauche à droite est :

`foundry` - font foundry, fonderie de police, la société ou la personne qui a fait cette police.

`family` - font family, famille de police, le surnom courant de cette police.

`weight` - font weight, graisse de police (gras, moyen, etc.)

`slant` - font slant, inclinaison de la police (italique, oblique, romain (normal), etc.)

`width` - font width, largeur de police (normal, condensé, étendu, etc.)

`adstyle` - additional style, style supplémentaire (sans serif, serif, etc.)

`pxlsz` - pixel size, taille en pixel, le nombre de pixels verticaux dans un caractère.

`ptSz` - point size, taille approximative en point du texte (similaire à `pxlsz`).

`resx` - résolution horizontale, en dpi.

`resy` - résolution verticale, in dpi.

`spacing` - spacing, espacement, seulement utile, apparemment avec les polices Schumacher.

`avgWidth` - average width, largeur moyenne d'un caractère de la police.

`registry` - registry, le registre reconnu qui cite cette police.

`encoding` - encoding, encodage de nationalité.

“ * ” agit comme un caractère joker. En fait, si tous les champs ne sont pas indiqués, le serveur de X prendra la première police correspondante qu'il trouvera dans le `FontPath`. C'est pourquoi il est préférable de classer les polices, dans le `FontPath`, en mettant les polices préférées en premier puisque certains programmes donneront des indications délibérément “vagues” sur les polices de sorte que votre système ait une certaine liberté.

Le programme **xfontsel** (X Font Selector) peut être utile. Vous allez voir en le lançant maintenant. Vous ne verrez rien d'utile dans la fenêtre principale au début, mais voyez ce que ça donne en maintenant le bouton gauche enfoncé sur le bouton `foundry`. Si toutes vos polices sont en règle, vous verrez un menu avec des choix tels que `adobe`, `b&h`, `bitstream` et ainsi de suite. Choisissez en comme par exemple `b&h` et vous remarquerez que la police dans la fenêtre du bas se transforme en quelque chose d'intelligible. C'est comme ça qu'on choisit les polices avec ce programme ; en commençant par la gauche, qui offre le choix le plus général, et en se déplaçant vers la droite, jusqu'aux options les plus spécifiques. Le choix d'une option de l'extrémité droite n'aura pas beaucoup de sens avant que la fonderie, par exemple, ne soit sélectionnée, parce que les options sont généralement ordonnées selon leurs dépendances les unes envers les autres.

Quand vous choisissez dans `family`, vous verrez la plupart des options grisées, et seulement trois restantes. Cela signifie que ces trois-là sont les seules familles de la police faites par cette fonderie.

Certaines familles apparaissent dans plus d'une fonderie, par exemple, *Adobe* et *Bitstream* font tous deux une version de la police Courier. Maintenant vous pouvez choisir le `wght`, et ainsi de suite. Après un certain temps, vous arriverez finalement à la police que vous voulez. Vous n'avez pas nécessairement besoin de compléter toutes les options pour choisir une seule police, il n'y pas autant de polices que ça sur votre système ! Les options que vous ne choisissez pas seront représentées par une `*` indiquant que n'importe quelle option fera l'affaire, et donne à X une certaine marge.

Quand vous êtes satisfait de votre choix de police, appuyez sur le bouton `select`, et votre choix sera placé dans le presse-papiers de X, prêt à être collé dans votre document ou dans ce que vous êtes en train de faire. Par exemple, ouvrez une fenêtre `xterm` et saisissez quelque chose comme `xterm -font` suivi d'un guillemet d'ouverture. Dirigez ensuite le curseur sur cette tâche à l'écran, et cliquez avec le bouton du milieu de votre souris (ou cliquez sur les deux, le gauche et le droit, si vous êtes handicapé du bouton du milieu). Cela collera le choix à partir du presse-papiers, qui devrait être la police que vous avez précédemment choisie. Entrez ensuite le guillemet de fermeture, et faites `Entrée`. Par exemple, un `xterm` avec une police Courier ressemblerait à ceci : `xterm -font "-adobe-courier-medium-r-*- *-14-*-*-*-*-*-*"`. C'est pas joli, ça ?

Si vous avez trouvé une police qui vous plaît, ceci peut être utilisé de manière permanente en plaçant la définition de la police dans le fichier de configuration approprié (voir ci-dessus).

Notez que, pour que `xfontsel` ne montre qu'un nombre limité de police, on utilisera l'option en ligne de commande `-pattern`, suivie des spécifications de police entre guillemets, comme on l'a expliqué ci-dessus.

L'utilitaire `xfd` est également utile pour examiner les polices individuellement. Si on le lance avec une ligne de commande telle que `xfd -fn fixed`, il montrera le jeu de caractères complet pour cette police.

KDE et GNOME ont leurs propres utilitaires qui montrent un peu plus de finesse ; -)

5.1.1. Les polices Type 1 et TrueType

Les polices fournies avec XFree86 sont d'une utilité limitée pour un bon nombre d'entre nous, étant donné qu'à peu près le seul endroit où vous trouverez des polices de ce type, c'est dans le système de fenêtre X lui-même. Malheureusement, parmi les accros des médias, les concepteurs de sites Web et les fous de police, nombreux sont ceux qui travaillent avec des systèmes d'exploitation qui reposent sur d'autres formats. Et puis, il ne semble pas que les distributions s'efforcent beaucoup de rendre les polices par défaut plus belles non plus.

Les polices de type 1, généralement les plus utilisées avec les formats de document Postscript, sont la norme traditionnelle dans des environnements d'Unix et de Linux. On a normalement un choix relativement intéressant pour démarrer. Sinon, c'est extrêmement facile d'en trouver d'autres gratuitement sur Internet ; pour démarrer, essayez `ftp://ftp.cdrom.com/pub/os2/fonts/`. Les types 1 sont des polices extensibles, et présentent en grande partie les mêmes avantages que les polices TrueType plus connues. Si le choix de polices TrueType installé n'est pas intéressant, le type 1 est ce qu'il vous faut

pour la plupart des applications à interface graphique. Mais répétons-le, ce n'est pas standard sur les autres plateformes, et cela peut présenter des problèmes lors de la consultation des documents (par exemple des pages Web) qui sont conçus pour “ d'autres plateformes ”.

Les polices TrueType ont été lancées par Apple, et ont été par la suite mises sous licence par Microsoft. Les gens venant des plateformes autres qu'Unix connaissent donc déjà ces polices de qualité. Malheureusement, il n'y a pas beaucoup de polices TrueType de qualité sous une licence appropriée, et il n'y en a donc pas beaucoup d'incluses avec les distributions de Linux. Et celles qui le sont, ne sont pas souvent d'aussi bonne qualité. En outre malheureusement, TrueType est un peu devenu une norme sur le Web et en d'autres lieux, et ne pas avoir de bonnes polices TrueType peut être un inconvénient. Avec XFree86, il semble également que TrueType donne un meilleur résultat que Type1.

Ca, c'était la mauvaise nouvelle. La bonne nouvelle, c'est que n'importe quelle police TrueType incluse avec n'importe quelle version de Windows, ou avec n'importe quelle application Windows, devrait fonctionner sous Linux. Cependant il faudra prendre quelques mesures supplémentaires pour les intégrer. Cela est particulièrement utile quand, en parcourant le Web, les polices bitmaps de X s'ajustent mal.

Nous n'entrerons pas dans les détails d'installation et de configuration des polices ici, car cela est bien détaillé dans d'autres documents. Voir le *Font HOWTO*, <http://www.linuxdoc.org/HOWTO/Font-HOWTO.html>, pour des informations générales sur les polices, et des astuces pour le Type 1. Voir le *Font De-Uglification Mini HOWTO*, <http://www.linuxdoc.org/HOWTO/mini/FDU/index.html>, pour diverses astuces en rapport avec X, et particulièrement pour TrueType.

5.2. Les couleurs

Retournons à notre terminal et essayons quelque chose. Ouvrez un **xterm** avec une ligne de commande comme suit :

```
xterm -fg DarkSteelBlue1 -bg red3 &
```

Aïe ! ça n'est peut-être pas très joli, et on ne va sans doute pas faire un excellent travail avec ça, il n'empêche que ça montre bien un aspect intéressant de la configuration de X : les noms de couleur. même si cela n'est pas particulièrement précis, c'est une manière simple de se rappeler de nombreuses de couleurs. Notez que les noms de couleur ne sont jamais sensibles à la casse.

Le serveur X traitera en fait les valeurs des couleurs avec une notation sous forme hexadécimale Rouge-Vert-Bleu (RVB). Cela ressemble à quelque chose comme “#0aff0a” en hexadécimal. Pas aussi simple à

se rappeler. Mais X donne une méthode mnémotechnique pour se rappeler des définitions de couleurs valides. Celles-ci sont stockées dans une table de texte, par exemple `/usr/X11R6/lib/X11/rgb.txt`, qui est définie dans `XF86Config` dans la section “files”.

Si vous êtes intéressés, allez jeter un oeil avec un éditeur de texte. Il y a beaucoup, beaucoup de nuances définies. J'ai compté 83 nuances de bleu dans le mien, par exemple. Petit échantillon :

```
176 226 255      LightSkyBlue1
164 211 238      LightSkyBlue2
141 182 205      LightSkyBlue3
 96 123 139      LightSkyBlue4
202 225 255      LightSteelBlue1
188 210 238      LightSteelBlue2
162 181 205      LightSteelBlue3
110 123 139      LightSteelBlue4
191 239 255      LightBlue1
178 223 238      LightBlue2
154 192 205      LightBlue3
104 131 139      LightBlue4
```

Ce fichier peut être adapté à vos besoins si vous le désirez, mais cela est rarement nécessaire pour la plupart d'entre nous. Cependant, il est important de l'avoir, puisque quelques applications dépendent de lui.

Les environnements graphiques ont un utilitaire avec interface graphique pour choisir les couleurs.

6. Gestionnaires de fenêtres et bureaux

Nous ne nous étalerons pas sur la configuration des gestionnaires de fenêtres et des environnements. Il y a tout simplement trop à dire pour essayer de couvrir cela dans un seul document. Il est important de se rendre compte que ce sont deux choses différentes. Il y a beaucoup, beaucoup de gestionnaires de fenêtres disponibles.

6.1. Les gestionnaires de fenêtres

Les possibilités de configuration des gestionnaires de fenêtres sont énormes. De nombreux aspects de l'interaction avec l'utilisateur peuvent être gérés par le gestionnaire de fenêtres.

Quelques-uns des gestionnaires de fenêtres les plus répandus :

aewm : <http://www.red-bean.com/~decklin/aewm/>

AfterStep : <http://www.afterstep.org/>

BlackBox : <http://sourceforge.net/projects/blackboxwm>

Enlightenment : <http://www.enlightenment.org/pages/main.html>

Fluxbox : <http://fluxbox.sourceforge.net/>

fvwm : <http://www.fvwm.org/>

IceWM : <http://www.icewm.org/>

olwm (OpenLook Window Manager) : <http://www.plig.org/xwinman/olvwm.html>

Sawmill : <http://sawmill.sourceforge.net/>

WindowMaker : <http://www.windowmaker.org/>

XFce : <http://xfce.org/>

Il en existe également beaucoup, beaucoup de moins connus. <http://www.plig.org/xwinman/> a une liste à jour des gestionnaires de fenêtres, et d'informations sur le sujet. Il y a toujours freshmeat (<http://freshmeat.net>) aussi.

GNOME ainsi que KDE ont leur gestionnaire de fenêtres par défaut, mais en acceptent d'autres également s'ils sont conformes. Votre distribution en a probablement plusieurs. Essayez les tous si vous n'avez pas déjà un favori. Votre distribution a probablement également une méthode de changement dynamique entre les gestionnaires de fenêtres (et entre les environnements graphiques également).

6.2. Les environnements graphiques

Les environnements graphiques ne sont pas vraiment nouveaux, mais leur popularité a augmenté avec l'arrivée des deux grands noms : KDE et GNOME. Dans une certaine mesure, la fonctionnalité d'environnement graphique recouvre celle du gestionnaire de fenêtres. Tous les deux peuvent gérer l'arrière plan des fenêtres du super utilisateur, du menu des fenêtres du super utilisateur, des icônes, des barres d'outils, etc. D'une manière générale, si un environnement graphique est en cours d'exécution, il gère tout ça. C'est le principe des environnements graphiques : intégrer les divers composants avec cohésion et cohérence. Les environnements graphiques apportent également des caractéristiques au niveau de l'interopérabilité et de la facilité d'utilisation qu'un gestionnaire de fenêtres simple n'a pas.

Ah, autre point : les environnements graphiques essayent également de configurer autant que possible les sessions de X. N'importe lequel de leurs clients conformes pourra plus que probablement être configuré par l'environnement graphique, ou aura sa propre configuration qui s'adapte au style de l'environnement graphique. Le but est au moins d'éviter en partie la configuration de fichiers textes à l'apparence désordonnés que nous avons vue dans les sections ci-dessus, et de faciliter un tant soit peu la vie de l'utilisateur.

Cette fonctionnalité supplémentaire repose sur un compromis, car elle consomme des ressources mémoire et système pour surveiller tout ceci. Si votre ordinateur a beaucoup de mémoire et qu'il est

rapide, ce n'est pas un problème. Mais si vous avez peu de mémoire, cela peut causer un ralentissement (voir la [section performance](#) ci-dessous). Un minimum de 64M de RAM est probablement limite avec KDE ou GNOME.

Alors, avez-vous besoin d'un environnement graphique ? C'est à l'utilisateur de choisir. Ils ne sont certainement pas indispensables pour lancer X, mais c'est vrai qu'ils apportent des fonctionnalités que beaucoup d'utilisateurs veulent ou attendent dans une interface graphique. Lequel est le meilleur ? Eh bien, c'est à vous de décider !

KDE existe depuis plus longtemps que GNOME, et certains diraient peut-être qu'il est un peu plus mûr. KDE est basé sur la boîte à outils de composants QT. Voici une citation de la page d'accueil de KDE (<http://kde.org>) :

KDE est un environnement graphique Open Source puissant pour les postes de travail Unix. Il combine la facilité d'utilisation, des fonctionnalités modernes, et un design graphique exceptionnel avec la supériorité technologique du système d'exploitation Unix.

KDE est une suite d'environnement graphique mature fournissant une base solide à un nombre toujours croissant d'applications pour les postes de travail Unix. KDE a développé un cadre de développement de qualité pour Unix, qui permet la création rapide et efficace d'applications.

GNOME est basé sur la boîte à outils GTK+. Et voici une citation de la page d'accueil de GNOME (<http://gnome.org>) :

GNOME signifie " GNU Network Object Model Environment ". Le projet GNOME a pour but de construire un environnement graphique complet et convivial basé entièrement sur les logiciels libres. GNOME fait partie du projet GNU, et GNOME fait partie du mouvement Open Source (tm). L'environnement graphique se compose de petits utilitaires et d'applications de taille plus importante sous un même aspect et une convivialité cohérents. GNOME emploie GTK+ comme boîte à outils d'interface graphique pour toutes les applications compatibles GNOME.

XFce est un environnement graphique plus léger, moins fourni en fonctionnalités, qui ne se fait pas autant remarquer que les autres. XFce est également basé sur la boîte à outils GTK+. Et voici une citation de la page d'accueil de XFce (<http://xfce.org>) :

Le projet XFce a au départ été lancé parce que j'avais besoin d'un environnement simple, léger et efficace pour mon système Linux.

Je crois que l'environnement graphique doit être fait pour augmenter la productivité des utilisateurs. Par conséquent, le but est de garder la plupart des ressources système pour les applications, et de ne pas consommer toute la mémoire et les capacités de l'UC pour l'environnement graphique.

Ils ont tous leur propre documentation bien détaillée. S'il vous manque quelque chose dont vous avez besoin sur votre système, jetez un oeil à leurs pages d'accueil respectives.

7. X et la ligne de commande

Que serait un système d'exploitation de type Unix sans interface en ligne de commande ? La ligne de commande peut être utile, et elle est disponible d'office avec le X. En fait, pour beaucoup, elle fait partie intégrante de leur environnement de travail de X.

Tout programme de X peut être lancé directement en ligne de commande juste en tapant le nom du programme dans une console **xterm**, ou dans toute autre terminal. La plupart des applications ont un ensemble très complet d'options en ligne de commande, comme la couleur d'arrière plan, la police, la géométrie (positionnement à l'écran), etc. Les options en ligne de commande surchargent les paramètres compilés par défauts, et les paramètres des ressources système.

Beaucoup de programmes traditionnels de X utiliseront les mêmes noms de base pour les options en ligne de commande. Toutes les applications écrites en utilisant le MIT X Toolkit Intrinsic (Xt) (comme ceux fournis avec XFree86) acceptent automatiquement les options suivantes. Certaines applications non-Xt utilisent également ces dernières, ou quelque chose de proche. Par exemple, "géométrie" n'est pas loin d'être une option universellement admise.

-display [*hôte*]:*display*[.écran]

Cette option indique au serveur X les affichages à utiliser. C'est fréquemment utilisé lorsque les applications s'exécutent sur un système et sont affichées sur un autre. L'application doit savoir où s'afficher. ça peut se faire également avec la variable "\$DISPLAY", qui emploie la même syntaxe.

-geometry *geometrie*

Taille et position initiales de la fenêtre, dans un format tel que `width x height +horz_offset +vert_offset` ou `+horz_offset -vert_offset`. Notez que si vous indiquez un décalage négatif horizontal ou vertical, la fenêtre sera placée en comptant vers l'arrière à partir de la droite ou du bas de l'écran, respectivement, au lieu de partir du coin supérieur gauche.

-font *nom de police*

Police à utiliser pour afficher le texte dans votre fenêtre (voir la section police ci-dessous).

-bg *couleur*

Couleur à utiliser comme arrière-plan de la fenêtre. En fait, il s'agit d'un "nom de couleur" (voir ci-dessous).

-fg *couleur*

Couleur à utiliser pour le premier plan de la fenêtre (c.à.d. la police, etc.).

-name *nm de ressource*

Utile pour indiquer le nom sous lequel les ressources de cette application se trouveront (par exemple comme indiqué dans `.Xdefaults`). C'est utile pour distinguer deux appels de la même

application. Par exemple, deux **xterms** peuvent “ être nommés ” différemment de sorte qu'ils puissent hériter de différentes ressources basées sur les noms indiqués dans la base de données des ressources.

-title *chaîne de caractère*

Il s'agit du titre qui sera utilisé pour la fenêtre dans votre affichage. Il est généralement utilisé par votre gestionnaire de fenêtre pour donner un titre explicite en haut de la fenêtre. A ne pas confondre avec l'option “-name”.

-iconic

Ouvre la fenêtre sous forme d'icône.

-xrm *chaîne de caractère-ressource*

Cette option indique un nom et une valeur de ressource pour surcharger tous paramètres par défauts qui pourraient avoir été spécifiés (c.-à-d. par l'intermédiaire de `.Xresources` ou semblable). En outre utile pour paramétrer les ressources de X qui n'ont pas d'option explicite en ligne de commande. Par exemple, la ligne de commande “`xterm -xrm xterm*background: blue &`” est pareille au niveau fonctionnel que “`xterm -bg blue &`”.

Ce sont les plus importantes. Il y en a d'autres. Beaucoup de programmes auront leurs propres options supplémentaires qui leur sont spécifiques. Aujourd'hui, beaucoup d'applications plus récentes n'adhèrent pas nécessairement aux normes Xt, et utilisent leurs propres options, ou celles fournies par leur boîte à outils respective. Au pire, les pages de man sont une bonne référence pour la syntaxe de commande, et il faut les compter parmi vos amis. Autre possibilité : l'application dispose d'une option en ligne de commande “ `--usage` ” ou “ `--help` ” pour donner la liste des options disponibles.

```
$ gnome-terminal --usage
Usage: gnome-terminal [-?] [--disable-sound] [--enable-sound]
  [--espeaker=HOSTNAME:PORT] [--version] [--usage] [--gdk-debug=FLAGS]
  [--gdk-no-debug=FLAGS] [--display=DISPLAY] [--sync] [--no-xshm]
  [--name=NAME] [--class=CLASS] [--gxid_host=HOST] [--gxid_port=PORT]
  [--xim-preedit=STYLE] [--xim-status=STYLE] [--gtk-debug=FLAGS]
  [--gtk-no-debug=FLAGS] [--g-fatal-warnings] [--gtk-module=MODULE]
  [--disable-crash-dialog] [--sm-client-id=ID] [--sm-config-prefix=PREFIX]
  [--sm-disable] [--tclass=TCLASS] [--font=FONT] [--nologin] [--login]
  [--geometry=GEOMETRY] [-e COMMAND] [-x COMMAND] [--foreground=COLOR]
  [--background=COLOR] [--solid] [--pixmap=PIXMAP] [--bgscroll]
  [--bgnoscroll] [--shaded] [--noshaded] [--transparent] [--utmp]
  [--noutmp] [--wtmp] [--nowtmp] [--lastlog] [--nolastlog] [-t TITLE]
  [--icon=ICON] [--termname=TERMNAME] [--start-factory-server]
  [--use-factory]
```

7.1. xterm et ses amis

Tôt ou tard, la plupart d'entre nous avons besoin d'accéder à la “ ligne de commande ” pour une raison ou une autre. Pour certains, cela pourrait même être une habitude de travail avec X. En plus de pouvoir lancer des applications de X depuis l'invite de commande, il y a également de nombreux programmes qui fonctionnent en “ mode texte ” sous Linux.

C'est possible par l'intermédiaire “ des émulateurs de terminaux ” comme **xterm**. Chez Microsoft, son homologue le plus proche, celle qui porte le nom de commande MS-DOS, qui est un jeu d'enfant à côté. Les consoles Linux permettent la couleur, le copié/collé à la souris (et quelques souris à mollette), les arrière-plans pseudo transparent et utilisant pixmap, les barres de défilement, les menus et généralement un tas d'autres dispositifs. Tandis que **xterm** est l'émulateur de terminal le plus connu, il y a beaucoup de programmes similaires. Pour en citer quelques uns : **Eterm**, **rxvt**, **aterm**, **konsole** (KDE) et **gnome-terminal**.

Le cas typique, c'est que quand une fenêtre d'émulateur de terminal est ouverte, un shell est lancé pour que l'utilisateur interagisse avec. Le shell par défaut pour quasiment tous les Linux, est bash. Donc, en fin de compte, l'utilisateur interagit avec X, le terminal, et le shell en même temps. Chacun peut avoir sa propre influence. Par exemple, comment les frappes sur les touches sont gérées puisqu'elles passent du matériel, au serveur X, au terminal, au shell et finalement sont retournées à l'utilisateur.

La configuration du terminal se fait vite et facilement par l'intermédiaire de la variable “ \$TERM ”, qui est définie dans un des fichiers de configuration du shell de l'utilisateur. Sinon le terminal lui-même en aura une de compilée par défaut. La valeur par défaut en est le plus souvent “ xterm ” :

```
$ echo $TERM
xterm
```

Normalement c'est suffisant, car l'installation faite par votre fournisseur est a priori correcte. La variable de “ \$TERM ” est en fait une référence à une entrée de la base de données “ termcap ” (man termcap), qui se trouve en principe dans `/etc/termcap`. A moins que vous fassiez quelque chose de vraiment peu commun, vous n'aurez probablement rien à changer. Certaines configurations supplémentaires des terminaux peuvent être faits avec la commande **stty** (voir la page de man). La configuration des terminaux n'entre pas du tout dans le cadre de ce document.

L'application de terminal elle-même (par exemple **xterm**) aura également diverses options de configuration. Le mieux pour les paramètres permanents est qu'ils soient stockés dans un `~/.Xdefaults` ou dans un fichier semblable pour celles de ces applications qui l'acceptent. D'une

manière générale, les applications pouvant être configurées via une interface graphique (par exemple **gnome-terminal**), le seront ainsi.

En outre, vous interagissez aussi avec le shell, qui peut avoir ses propres effets, en particulier sur la façon dont les frappes de touches sont gérées par l'invite de commande. Pour bash, le réglage se fera dans `~/ .inputrc`. Là encore, ceci n'entre pas dans le cadre de ce document, mais jetez un oeil aux références locales ou en ligne de bash (ou d'autre shell).

Les émulateurs de terminaux comme **xterm** exigent une police à espacement constant. Oubliez alors les polices de TrueType ou Type 1.

8. X ; le réseau et la sécurité

Ainsi qu'il a été mentionné, X est essentiellement un protocole de gestion de réseau avec des capacités d'affichage graphique. Cela rend possible certaines utilisations intéressantes. Et signifie également qu'il y a des aspects de sécurité inhérents, comme il en est avec n'importe quel environnement en réseau. Et si jamais vous vous connectez à l'Internet, vous vous retrouvez en plein dans un réseau énorme, et hostile ; -)

Les clients X se relient aux serveurs X par l'intermédiaire de divers protocoles de gestion de réseau, y compris TCP/IP. même avec des connexions simplement locales. Les utilisations possibles ici sont d'exécuter une application sur un ordinateur, et de l'afficher sur d'autres. Ou bien même, de vous connecter à un système distant, et avoir son affichage sur votre écran local, avec les applications cliente qui utilisent l'unité centrale et la RAM de l'ordinateur distant.

Sans précautions, ceci peut vous laisser complètement exposé à toutes sortes d'actes de malveillance et d'abus. Par exemple, tout utilisateur connecté à votre système peut accéder à votre " affichage ", ce qui signifie qu'il peut voir ce que vous faites s'il veut. Heureusement, la plupart des versions récentes de Linux sortent avec par défaut quelques protections au niveau sécurité. Mais il est préférable de s'assurer soi-même qu'on est bien protégé.

La gestion de réseau et la sécurité de X sont bien expliquées dans le *The Remote X Apps Mini HOWTO*, <http://www.linuxdoc.org/HOWTO/mini/Remote-X-Apps.html>, nous n'aurons donc pas besoin d'essayer de rebrasser tout ça ici. Sa lecture en est recommandée. Voir d'autres références dans la [section consacrée aux liens](#) dans l'annexe plus bas.

Quelques précautions à prendre :

- Ne jamais lancer X comme super utilisateur. Le nombre de choses désagréables qui peuvent se produire, augmente très nettement une fois connecté en super utilisateur. Apprenez à lancer autant que possible en tant que simple utilisateur, et utilisez " su " pour être super utilisateur seulement si

nécessaire. Ceci peut sembler faire beaucoup de travail en plus (et c'est probablement le cas au début), mais une fois qu'on a appris comment faire les choses correctement, cela devient vite une deuxième nature.

Petite anecdote racontée par un ami : il a eu un client dont le nouveau système s'est arrêté de “ fonctionner ”. Curieusement, il a constaté que le répertoire entier `/dev` était absent, il l'a réinstallé et tout est revenu dans l'ordre. De retour quelques jours plus tard, il a constaté que le système était connecté en super utilisateur à X, et que quelqu'un avait cliqué sur `/dev` dans le gestionnaire de fichier, et l'avait déplacé vers son bureau. Joli coup !

- S'il vous arrive de vous connecter à un réseau avec des utilisateurs non sûrs, veillez à bien avoir un pare-feu entre eux et vous. C'est encore plus vrai avec Internet. La configuration des pare-feux n'entre pas dans le cadre de ce document, mais est traitée dans beaucoup d'autres endroits, y compris le site Web de votre fournisseur. <http://linuxdoc.org> a de nombreux guides sur la sécurité qui peuvent également vous aider. <http://linuxsecurity.com/docs/> est intéressant également.
- Vous pouvez désactiver les connexions TCP avec la ligne de commande de serveur X “ `-nolisten tcp` ”. Cependant ceci n'est d'aucune aide pour les connexions locales. Pour **xinit/startx** :

```
exec X :0 -dpi 100 -nolisten tcp
```

Se trouve dans `~/.xserverrc`. Et pour **xdm**, dans `/usr/lib/X11/xdm/Xservers` :

```
:0 local /usr/X11R6/bin/X :0 -nolisten tcp
```

9. Parlons performance

Comme on l'a dit, ce que nous appelons X, est en fait une convergence de divers composants : le serveur X, le gestionnaire de fenêtres, l'environnement graphique, etc. Avec Windows, l'interface graphique de l'environnement est étroitement intégrée dans le système d'exploitation lui-même. Ce n'est pas le cas de Linux qui suit la tradition d'Unix de combiner divers composants indépendants pour aboutir à un certain résultat final. Nous avons donc des choix avec chaque composant, sa configuration et son implémentation. En bref, beaucoup de souplesse. C'est là que vous intervenez. Vous pouvez essayer diverses possibilités et voir celle qui vous en donnera pour votre argent.

Pour le matériel peu performant, cela donne beaucoup de latitude pour diminuer la demande sur les ressources système disponibles. C'est bien parce que, s'il en a l'occasion, X peut être tout à fait gourmand en ressources système. Si vous avez récemment installé une nouvelle distribution Linux, on vous a probablement donné un environnement graphique par défaut avec plein d'accessoires. Et quelque chose qui aura besoin probablement d'une quantité considérable de mémoire et d'unité centrale pour arriver à un niveau convenable de performance. Si vous avez un ordinateur puissant, ceci ne devrait pas être un problème.

On dit souvent que Linux fonctionne très bien avec relativement peu de mémoire. C'est vrai jusqu'à un certain point. Cela ne signifie pas cependant que ce sera le cas avec toute configuration. Donc si vous voulez utiliser des applications gourmandes en mémoire, vous devrez avoir de la mémoire. Ou alors vous devrez faire des sacrifices pour obtenir un niveau satisfaisant de performance. Il est tout à fait possible de faire tourner X avec des performances convenables sur 16 mégas de RAM, et même moins si vous voulez vraiment pousser. Mais vous devrez vous imposer quelques restrictions.

Voyons certains des composants et comment diminuer la demande sur les ressources système, au cas où vous seriez avec du matériel peu puissant, ou si la performance n'est pas au niveau de vos espérances.

9.1. Le matériel

- Pas vraiment de surprise ici, mais la performance globale du système sera meilleure avec une carte graphique rapide, un disque dur rapide, et avec beaucoup beaucoup de mémoire (si vous voulez *à la fois* un système rapide *et* tape-à-l'oeil).
- Les cartes graphiques sont évidemment nécessaires, et les performances vidéo du serveur X sont liées aux puces de la carte, et au pilote de XFree86 correspondant. Le simple fait qu'une carte soit supportée par XFree86 ne signifie pas qu'elle est aussi bien optimisée que d'autres cartes ! La performance peut également être meilleure avec une intensité de couleur inférieure (voir ci-dessous). Cela peut valoir la peine d'aller faire un tour sur xfree86.org pour voir s'il y a des notes liées à votre carte en ce qui concerne les performances ou d'autres questions.
- Et vous pourriez essayer d'autres versions de XFree86. Actuellement la v4.2 vient juste de sortir. Quelques cartes peuvent encore avoir de meilleures performances avec la v3.3.6 du fait que le développement de la v4.x est incrémental. Si vous utilisez une v4.x et que la performance n'est pas bonne, assurez vous que vous utilisez la dernière version disponible.

9.2. La mémoire

Plus vous avez de mémoire, mieux c'est. X utilise beaucoup le cache pour améliorer la performance. Mais mettre en cache exige de la mémoire, et s'il n'y en a pas beaucoup au départ, il faudra réduire les besoins en mémoire. Quelques astuces pour ceux avec peu de mémoire ou avec des problèmes de performances :

- Utilisez la commande **free** pour s'assurer que mémoire et transfert de mémoire [swap] sont intégralement reconnus.
- Assurez vous que vous n'avez pas d'autres services système qui s'accaparent la mémoire ou l'UC. Utilisez **top** ou **ps** pour voir ce qui est en cours d'exécution, et désactivez ce que vous pouvez pour libérer de la mémoire et de l'UC. Encore une fois, avec une installation par défaut on peut avoir de nombreuses choses qui tournent dont on n'a pas vraiment besoin.
- Assurez vous d'avoir beaucoup d'espace de swap. Avec très peu, ou même peu, de mémoire, le swap est d'autant plus important. Le principe de base est d'avoir deux fois plus d'espace de swap que de mémoire physique. Avec peu de mémoire, ce n'est pas assez. Essayez quatre fois la mémoire physique. Ou plus. Si vous ne pouvez pas créer plus de partitions de swap, voir la page de man de **mkswap** sur la création de fichiers de swap à la place. Le bruit constant du disque est un symptôme du manque d'espace de swap, et le système sera ralenti en conséquence, parfois considérablement. Sinon, il se peut bien que ce soit le symptôme d'un système de mémoire virtuelle dont le noyau se comporte mal (essayez un autre noyau dans ce cas).
- Les performances du disque sont importantes pour la performance du swap. Assurez vous que votre disque a l'accès direct à la mémoire [DMA] d'activé s'il le permet, et que par ailleurs son paramétrage et ses performances sont à la hauteur. Voir la page de man de **hdparm**. Disque lent + carte lente + peu de mémoire = système lent.
- Ne pas employer KDE ou GNOME si la mémoire est juste. Tous les deux exigent beaucoup de mémoire, et ne sont pas nécessaires pour simplement faire tourner le X. Considérez ces derniers comme des outils pour améliorer la facilité d'utilisation. 32M seront sans doute insuffisants. Avec 64M, ça devient à peu près honnête, ça dépend des autres applications qui sont utilisées, et d'autres variables. 128M *devraient* convenir dans la plupart des situations. Pour être à l'aise, il faudra 256M ou plus. Les gestionnaires de fichiers comme Nautilus et gmc peuvent également être gourmands en mémoire.
- Utilisez un gestionnaire de fenêtres léger. WindowMaker, BlackBox, IceWM, fvwm (et variantes), XFce, tous sont réputés pour donner de bonnes performances avec peu de mémoire. Il y en a sûrement d'autres également. Faites des essais. fvwm est généralement considéré comme le plus léger des légers.

Un très joli bureau est toujours très possible même sans KDE ou GNOME. En fait, la plupart des applications KDE et GNOME peuvent tout de même être utilisées même si KDE et GNOME ne sont pas exécutés (du moment que les bonnes bibliothèques sont installés).

- Ne pas utiliser les thèmes ou les arrière plans de fantaisie. Le sobre et le simple sont moins exigeants pour les ressources. Utilisez un arrière plan fait d'une couleur pleine. évitez les pixmaps ou les gradients pour tout type de fond, y compris les menus, les barres de titre, etc.
- Limitez la taille d'écran et la profondeur de couleur. 800x600x16 ne poussera pas X dans ces retranchements et sera plus facile pour les ressources du système que des valeurs plus élevées. même si une profondeur de couleur de 24 est préférable, on ne remarquera probablement pas la différence avec 16 pour la majorité des applications.
- Certaines applications exigent beaucoup plus de mémoire que d'autres. Parmi les goinfres on peut citer **Netscape**, **Mozilla**, les suites bureautiques, et **the Gimp**. **Netscape** est plus rapide que **Mozilla** (mais pas aussi bien). **Netscape-Navigator** utilise moins de mémoire que **Netscape-Communicator**.

Fermez ces applications dès qu'elles ne sont plus utilisées. Utilisez les navigateurs textes comme **lynx** ou **w3m** partout où c'est possible, par exemple pour lire la documentation HTML localement installée. Beaucoup plus rapide, et beaucoup moins de mémoire exigée.

- En outre, utilisez des clients en mode texte pour les courriels (**mutt** ou **pine**) et les news (**slrn** ou **trn**). Là aussi, plus rapide et beaucoup moins de mémoire utilisée, et ceux-ci sont après tout des protocoles basés sur le texte.
- **rxvt** utilise moins de mémoire que **xterm**, **konsole** ou **gnome-terminal**.
- Si vous lancez une session de X pendant de longues périodes (genre des jours ou des semaines), redémarrez X de temps en temps pour libérer la mémoire immobilisée en cache.
- Désactivez “backing store” et “save-unders” pour réduire l'utilisation de mémoire (mais cela pénalise la performance). Vérifiez aussi les paramètres du gestionnaire de fenêtres. Allez voir quels modules sont chargés dans la section “modules” de `XF86Config` également. Dans votre installation, beaucoup de modules inutiles, ou dont vous ne pouvez pas tirer profit, peuvent se trouver activés (par exemple “v4l”, alias “Video4Linux”).
- Les serveurs de polices peuvent fournir une légère poussée au niveau performance en déchargeant le rendu de la police au serveur de polices, tout en libérant le serveur X pour faire d'autres choses. Mais, le serveur de police emploiera une petite quantité de mémoire supplémentaire également. Vous pouvez donc essayer l'une ou l'autre méthode pour voir s'il y a une différence.
- Pour finir, la RAM est bon marché maintenant. Achetez-en ; -) Un nouveau disque aussi.
- La RAM est encore insuffisante pour X ? Jetez un oeil à tinyX : <http://www.superant.com/smalllinux/tinyX01.html>. On dit qu'il tourne avec une RAM pas plus grosse que 4 mégas.

9.3. X sur le réseau

X n'est pas vraiment sympa sur le réseau. En d'autres termes, c'est un gros mangeur de bande passante. Ceci ne devrait pas poser de problème si vous êtes sur un LAN, mais si vous essayez d'utiliser X sur Internet, ça peut en poser.

- **lbxproxy**, le proxy X à faible bande passante, met en oeuvre diverses optimisations pour améliorer les performances en cas de faible bande passante ou de temps de réponse élevés. Voir la page de man.
- VNC (Virtual Network Computing), possède une partie des avantages de X pour afficher des applications sur des systèmes distants, mais est plus adapté au réseau. Votre installation de Linux doit avoir les paquets client et serveur de VNC de disponibles.

9.4. Autres astuces

D'autres astuces pour de meilleures performances :

- Utilisez **xset** pour accélérer le clavier. ça peut donner la sensation que le système est plus réactif même s'il ne l'est pas vraiment. Les réglages par défaut m'ont toujours semblé mous.
- **renice X** pour lui donner une priorité plus élevée. Les autres plateformes accordent une priorité d'exécution plus importante à l'interface graphique pour obtenir une meilleure réactivité. Mais c'est au détriment d'autres processus. Linux est façonnable à volonté. Vous pouvez inclure le serveur de polices (si utilisé), et les processus clefs de KDE et d GNOME également.

```
nice -n -10 X :0
```

Ceci ne changera pas grand chose sur les systèmes qui sont la plupart du temps à vide.

Ceci ne fonctionne pas très bien avec **startx** puisque X s'exécute comme super utilisateur, et vous n'êtes pas super utilisateur, n'est-ce pas ? Il faudrait donc utiliser quelque chose comme **sudo** pour que ça se fasse automatiquement.

10. 10. Annexe

10.1. Terminologie et utilisation

Il y a quelques concepts et terminologies de base qu'il faut bien connaître. Ces termes apparaîtront ici, dans les pages de manuel, et dans les fichiers d'aide et de documentation.

- Le “ serveur de X ” est le pilote de bas niveau qui interagit avec votre carte graphique et tout autre matériel, qui contrôle les “ affichages [display] ” et les divers composants liés aux “ affichages ” (clavier, souris, etc.). Et qui, naturellement, gère également les requêtes de nombreux clients. Il existe différents serveurs de X pour différentes puces.

Les serveurs X Servers sont référencés sous la forme suivante :

hôte.domaine:numéro_affichage.numéro_écran

Un exemple ressemblerait à : *mon_ordinateur:0.0*

Si l'hôte (et le domaine) est omis, on considère que c'est localhost. "L'hôte" peut être un hôte à distance. Si l'"écran" est omis, alors on considère que c'est "0" (le premier écran). Dans sa forme la plus courte, le serveur X est souvent juste représenté par ":0", qui est le premier "affichage" local. X permet les "affichages" et les "écrans" multiples.

L'"écran" et l'"affichage" ont des significations spéciales avec les serveurs X, différentes de leurs utilisations plus courantes.

- Quand X est appelé, le serveur de X initialisera un ou plusieurs "affichage". Oui, X peut avoir plus d'un "affichage" disponible (bien que ce ne soit pas une configuration commune pour l'utilisateur lambda). Chaque "affichage" est une instance séparée de "X". L'"affichage" inclut non seulement les composants visuels évidents, mais également le clavier, la souris et tout autre type de composant d'entrée. L'utilisateur peut accéder à un seul display à la fois par l'intermédiaire des mêmes clavier et moniteur. Les "affichages" peuvent résider localement, ou sur un hôte géré via le réseau "quelque part", ou sur tous les deux. Il est possible que si les "affichages" multiples sont disponibles, l'utilisateur puisse choisir lequel il souhaite lorsqu'il ouvre une session. Chaque "affichage" peut avoir sa propre configuration unique (par exemple sa résolution). Mais encore une fois, la configuration la plus typique est juste un "affichage" avec un "écran", et c'est comme cela que la plupart d'entre nous utilisent X.
- Dans le domaine des serveurs X, l'"écran" signifie la sortie graphique primaire avec laquelle vous regardez X. Et il peut y avoir plus d'un "écran", tout comme il peut y avoir plus d'un "affichage". Des "écrans" supplémentaires sont utilisés dans "les affichages multi-dirigés" par exemple. En fait vous pouvez même avoir plus d'un ordinateur qui fonctionne sous un simple serveur X. Ceci n'entre pas dans le cadre de ce document, mais vous devez être conscient de ce degré de souplesse car c'est un ingrédient important du protocole de X.
- "Bureau" peut signifier différentes choses dans différents contextes. Souvent, le "bureau" désigne ce qui s'appelle plus correctement "environnement graphique de bureau" ou "environnement graphique". Les plus connus sont KDE, GNOME, et moins connu CDE, qui sont des applications de haut niveau qui gère une grande partie de l'interaction utilisateur dans une session de X. Ils fournissent un aspect et une convivialité cohérents, ainsi qu'une configuration cohérente et intègrent leur propre ensemble d'utilitaires pour les tâches courantes.

Le "bureau" désigne parfois seulement la zone d'écran visible. On est plus proche là du sens qu'en donne Windows. Les environnements X sont cependant capables d'avoir des "bureaux" virtuels multiples et on peut passer des uns aux autres à volonté. Cela aide à organiser vos différentes tâches. Chaque "bureau" peut avoir ses propres fenêtres et clients qui lui sont spécifiques. En ce moment j'ai sept bureaux WindowMaker (WindowMaker les appelle des "zones de travail" [WorkSpaces]), et j'utilise l'un d'entre eux exclusivement pour la rédaction ce document. Ce "bureau" a treize fenêtres particulières à l'heure actuelle (des pages de man, des fenêtres de navigateur, l'horloge, gvim, des xterms, etc.).

- Les "clients" représentent tout programme qui se connecte au serveur X, et a besoin d'un serveur X pour certaines tâches (par exemple pour s'afficher). Souvent, ceux-ci s'affichent dans leurs propres "fenêtres", mais pas toujours. Par exemple, si j'utilise `CTRL-N` pour ouvrir une nouvelle fenêtre de Mozilla, c'est un seul client X mais avec deux fenêtres. Si j'exécute un utilitaire de X en ligne de commande comme `xev` pour voir les événements des touches et de la souris, celui-ci fonctionne dans la fenêtre de `xterm`, et n'a donc pas de fenêtre propre, mais c'est tout de même un "client". Les

clients peuvent être des applications qui s'exécutent localement, ou des applications qui fonctionnent sur un autre système au travers du réseau, mais qui sont affichées localement.

- Le “ gestionnaire de fenêtres ” est un type particulier d'application cliente et un composant de l'interface graphique définissable par l'utilisateur. C'est essentiellement avec lui que l'utilisateur interagit. Le gestionnaire de fenêtres fournit des fonctionnalités telles que les bordures et décorations de fenêtres, les menus, les icônes, les bureaux virtuels, les barres de bouton, les barres d'outils, et permet à l'utilisateur de personnaliser ces derniers. Il est techniquement possible d'exécuter X sans gestionnaire de fenêtres (toutefois ce n'est pas très fonctionnel), mais pas le contraire. Les gestionnaires de fenêtres ne doivent pas être confondus avec les “ environnements graphiques ” comme KDE. Les environnements graphiques incluent leur propre gestionnaire de fenêtres préféré, mais c'est configurable. Les responsabilités entre les gestionnaires de fenêtres et les environnements graphiques se recouvrent en partie.
- La “ fenêtre de root ” est l'arrière plan de votre écran. Malgré son nom, ce n'est pas une fenêtre, elle ne se comporte comme aucune autre fenêtre, mais en fait les applications s'exécutent dessus, ou bien on y met une image, ou peut-être juste une couleur unie. Toutes les autres fenêtres sont des enfants de cette fenêtre parent. La fenêtre de root peut tout à fait être plus grande que la zone d'écran visualisable.
- Le “ curseur ” est la flèche, ou tout indicateur quelle que soit sa forme, qui représente l'emplacement de votre souris ou de tout autre dispositif de pointage. Le curseur agit souvent pour donner un retour contextuel quant à ce qui se produira quand vous utiliserez la souris sur tel point de l'écran.
- La “ fenêtre ” est un cadre dans lequel toute application donnée s'exécute et qui “ est contrôlé ” par le gestionnaire de fenêtres. Ceci inclut quasiment tout excepté la dénommée fenêtre de root. même les fenêtres qui ne semblent pas avoir du tout de cadres, de titres, ou de bordures normales sont contrôlées par votre gestionnaire de fenêtres. “ La fenêtre active ” est la fenêtre que vous utilisez actuellement. Cette fenêtre répondra quand vous tapez au clavier, et, si la tradition est respectée, on peut la repérer par le fait que le curseur de souris pointe dessus, mais ce n'est pas toujours le cas. On dit que la fenêtre active a le “ focus ”. La plupart des gestionnaires de fenêtres mettront en évidence d'une façon ou d'une autre la fenêtre “ active ”, ou qui a le focus, pour la différencier des autres fenêtres.
- Les “ menus ”, les “ icônes ” et les “ barres de tâches ” se comportent dans X de la même manière qu'ils se comportent dans d'autres systèmes de fenêtrage, et les mêmes principes généraux s'appliquent.
- Les fenêtres qui exécutent seulement des applications texte s'appelle “ les émulateurs de terminaux ”, par exemple **xterm** et diverses applications similaires. C'est la “ ligne de commande ” bien connue dans un environnement X. En gros, ils émulent un affichage de console en texte, et ont quelques avantages dès leur utilisation dans X. Ce sont des applications beaucoup plus complexes et sophistiquées que la simple boîte DOS sous Windows.
- “ Widgets ” est le terme employé pour décrire des composants de commande d'interface graphiques tels que les boutons, les glisseurs, les menus, les barres de défilement, les boîtes de liste, les cases à cocher, etc. Les boîtes à outil [Toolkits] sont des bibliothèques contenant un ensemble varié de widgets avec le même aspect et la même convivialité. Parmi les exemples courants, on a GTK+ (employés par GNOME, Mozilla et d'autres), Xaw (l'ensemble X Athena Widget), Tk, Motif et QT (utilisé par KDE). Les applications sont construites avec telle ou telle boîte à outils. Parfois la même application peut être construite avec différentes boîtes à outils, selon les options de compilation au niveau temps.

- La “ géométrie ” d'une fenêtre est une manière simple d'exprimer la taille de la fenêtre et le positionnement à l'écran. Cela peut ressembler à “ 60x20+10+50 ”, qui est la `WIDTH x HEIGHT +VERT_OFFSET +HORZ_OFFSET`. Même si le plus souvent les deux paires de valeurs sont indiquées, il est permis d'utiliser juste l'une ou l'autre.
- Dans le jargon de X, les “ressources” sont des attributs d'applications qui peuvent être définis. Les “ressources” généralement disponibles sont les polices, les couleurs, la taille, le titre des fenêtres, etc., etc.

10.2. Liens et autres références

- La meilleure source d'information sur Xfree86 est, évidemment, <http://xfree86.org>. N'oubliez pas également les pages de man que vous avez déjà d'installées. (X, Xserver, XF86Config, XFree86, xdm, xinit, xmodmap, startx, xauth, Xsecurity, etc, etc). Elles sont souvent d'une qualité tout à fait honnête, même si certaines sont un peu techniques.

Quelques pages de xfree86.org à visiter :

Documentations et informations d'aide : <http://www.xfree86.org/support.html> pour de nombreux sujets et version.

Lisez-moi : <http://www.xfree86.org/current/README.html>

Notes de sortie : <http://www.xfree86.org/current/RELNOTES.html>

DRI: <http://www.xfree86.org/current/DRI.html>

Statut : <http://www.xfree86.org/current/Status.html>

Souris : <http://www.xfree86.org/current/mouse.html>

Liste des cartes acceptées : <http://xfree86.org/cardlist.html>

- Autres documents sur le sujet, proposés par le LDP :

- Si vous démarrez tout juste, vous trouvez le *X Window System Architecture Overview HOWTO*, <http://www.linuxdoc.org/HOWTO/XWindow-Overview-HOWTO/> (<http://www.linuxdoc.org/HOWTO/XWindow-Overview-HOWTO/index.html>) très utile. Il traite assez bien tous les concepts de base.
- *The Remote X Apps Mini HOWTO*, <http://www.linuxdoc.org/HOWTO/mini/Remote-X-Apps.html>, propose des explications très intéressantes sur l'exécution de X à distance, et sur les questions de sécurité relatives..
- Le *XDMCP HOWTO*, <http://www.linuxdoc.org/HOWTO/XDMCP-HOWTO/index.html>, traite le X Display Manager Control Protocol, pour exécuter X à distance. Egalement, *The XDM and X Terminal mini-HOWTO*, <http://www.linuxdoc.org/HOWTO/mini/XDM-Xterm/index.html>.
- Le *XFree86 HOWTO*, <http://www.linuxdoc.org/HOWTO/XFree86-HOWTO/index.html>, traite

succinctement l'installation et la configuration initiale.

- Le *XFree86 Video Timings HOWTO*, <http://www.linuxdoc.org/HOWTO/XFree86-Video-Timings-HOWTO/index.html>, va vraiment au fond des choses pour le paramétrage précis de votre écran. Généralement peu utile avec XFree86 v4.x.
 - Le *Xinerama HOWTO*, <http://www.linuxdoc.org/HOWTO/Xinerama-HOWTO.html>, traite les affichages multidirigés.
 - Le *Font HOWTO*, <http://www.linuxdoc.org/HOWTO/Font-HOWTO.html>, traite plusieurs sujets en rapport avec les polices.
 - Le *Font De-Uglification Mini HOWTO*, <http://www.linuxdoc.org/HOWTO/mini/FDU/index.html>, traite un ensemble de question sur les polices de X.
 - Le *International Keyboard HOWTO* ; <http://www.linuxdoc.org/HOWTO/mini/Intkeyb/index.html>
 - Le *Linux Infrared HOWTO* : <http://www.linuxdoc.org/HOWTO/Infrared-HOWTO/index.html> (<http://www.linuxdoc.org/HOWTO/Infrared-HOWTO/index.html>)
-
- Vous êtes à la recherche d'informations sur un gestionnaire de fenêtre ou voulez essayer quelque chose de nouveau ou de différent : <http://www.plig.org/xwinman/>
 - Configuration et astuces sur les souris à molette : <http://koala.ilog.fr/colas/mouse-wheel-scroll/>
 - Linux et les ordinateurs portables : <http://www.linux-laptop.net/>
 - La sùrie O'Reilly sur X Window ! Visitez <http://www.ora.com/> pour les livres les plus complets sur X.
 - Le site web du Consortium X est <http://www.x.org/> ou alors il a peut-être déménagé sur <http://www.opengroup.org/>.
 - <http://www.x11.org/> est en quelque sorte un bureau central pour tout ce qui touche à X.
 - Et pour tout le reste : <http://google.com/linux/>. Une documentation incroyable à portée de clavier.