



X Window



1 - Introduction

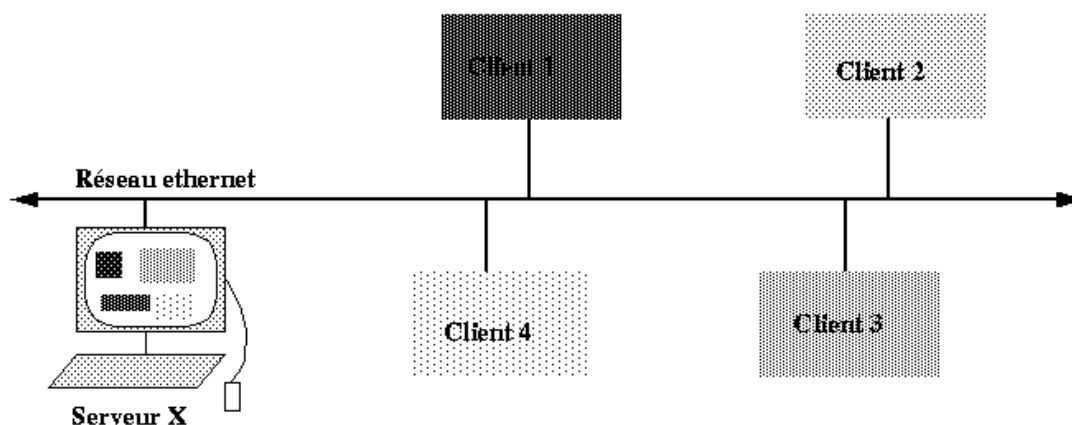
1.1 - Présentation générale

X-Window est une interface graphique multi-fenêtre distribuée, on l'appelle souvent X. X-Window est issu de travaux réalisés au Massachusetts Institute of Technology (MIT) dans le milieu des années 80. L'objectif initial était de créer un système permettant de travailler sur des machines hétérogènes en réseau tout en disposant d'une interface utilisateur graphique commune.

Très rapidement le monde industriel s'est rendu compte de l'intérêt de X et un consortium unissant de gros constructeurs informatiques (DEC, HP, IBM, SUN,...) et le MIT a été créé. Le consortium gère l'évolution de X et garantit sa pérennité

1.2 - L'architecture

X-Window est construit autour d'un modèle client/serveur. Un client X est un demandeur de ressources graphiques ; c'est un programme d'application qui se déroule quelque part sur une machine du réseau. Un serveur X assure la gestion des ressources graphiques d'un poste de travail et les interactions avec les clients X (gestion de l'écran, du clavier et de la souris). La figure ci-dessous montre les relations entre un serveur X et quatre clients X s'exécutant sur des machines différentes du réseau.

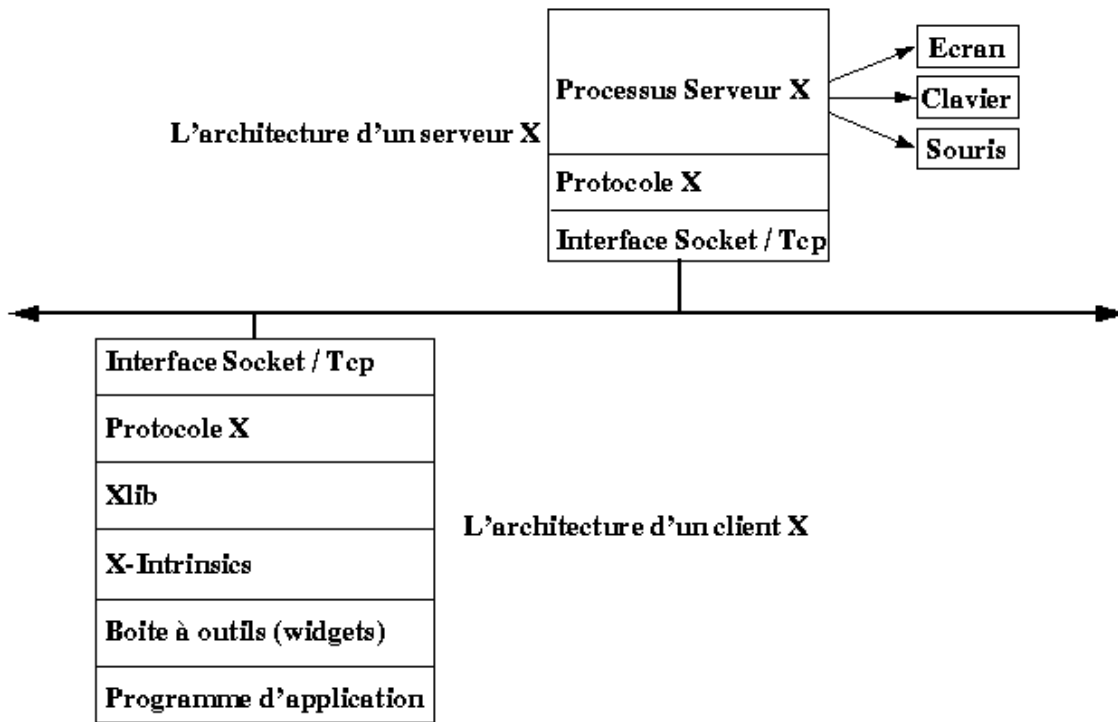


Le serveur X analyse les données en provenance des clients et sait dans quelle fenêtre (portion d'écran) faire l'affichage correspondant.

Ainsi il est possible visualiser le résultat de plusieurs clients, en provenance de n'importe quelle machine du réseau, sur un seul serveur (un seul écran). De plus, ce dernier analyse la position du pointeur de la souris pour déterminer à quel client transmettre les informations saisies au clavier par l'utilisateur.

1.3 - Le fonctionnement

La figure ci dessous présente les divers éléments constituant l'environnement des applications X-Window :



La partie communication est basée sur l'interface socket qui permet de faire dialoguer des processus ne résidant pas sur une même machine. L'aspect communication proprement dit est donc entièrement sous traité.

Pendant leur exécution les applications (les clients X) doivent échanger des messages avec un serveur X indépendamment du type des machines utilisées : c'est le rôle du protocole X. Il définit quatre types de paquets d'échanges entre serveurs et clients :

- les requêtes d'un client vers un serveur ;
- les réponses du serveur aux requêtes des clients ;
- la signification par le serveur des événements du type : une touche du clavier a été enfoncée ou on a cliqué sur un bouton de la souris ;
- l'envoi par le serveur d'éventuelles erreurs .

Le protocole X constitue la base de l'échafaudage X-Window, sa pérennité est garantie par le consortium.

La Xlib est une interface de programmation du protocole X, elle est écrite en langage C. Les appels à la Xlib génèrent du protocole X. La Xlib fait partie du standard garanti par le consortium, elle est installée sous le nom de libX11.a.

La manipulation de fenêtres par la Xlib est relativement fastidieuse pour le programmeur. Un niveau supérieur, destiné à faciliter la programmation d'applications a donc été créé : X Toolkit Intrinsics. Cette bibliothèque, connue sous le nom d' Intrinsics ou Xt, introduit la notion de widgets. Un widget est une structure de données accompagnant une fenêtre



X Window



permettant ainsi de qualifier aisément certains paramètres (taille, fond, contours ...). Une hiérarchie des widgets est introduite par Xt, ainsi, en détruisant une fenêtre, on peut d'un seul coup faire disparaître toute sa descendance. La bibliothèque Xt est installée sous le nom de libXt.a et fait partie du standard garanti par le consortium.

Un dernier niveau de bibliothèque, appelé boîte à outils, est disponible pour le programmeur. L'idée est d'aller toujours plus loin dans les facilités offertes pour la programmation, des widgets taillés sur mesure pour tel ou tel type d'applications apparaissent. Les boîtes à outils les plus connues sont les Athena's Widgets (incluses dans la livraison de base du MIT) et Motif de l' OSF. L'une comme l'autre définissent des widgets de type bouton de commande, boîte de dialogue, barre de défilement ...

En utilisant un widget de type texte, un programmeur créera très facilement une fenêtre dans laquelle on pourra éditer un fichier avec, par défaut, les commandes d'édition du style emacs . Les boîtes à outils introduisent des notions de présentation des applications (le fameux look and feel), Motif semblant avoir emporté l'adhésion du monde industriel.

1.4 - Comment se procurer un serveur X

En général, les logiciels permettant l'affichage (à travers un réseau éventuel) de clients X sur un serveur sont composés de plusieurs couches. La première couche est la couche de transport, par exemple TCP/IP . La seconde est la couche de gestion de l'écran proprement dite, le serveur. Ces deux couches (qui peuvent être combinées en un seul logiciel) forment le minimum pour utiliser X11 . Certains systèmes offrent également la possibilité de gérer les interactions avec l'utilisateur de façon locale avec un logiciel nommé window-manager . Dans le cas contraire, cette couche peut fonctionner à distance.

UNIX

Les systèmes UNIX, lorsqu'ils sont livrés avec un écran graphique, le sont généralement également avec un serveur X autorisant l'affichage. Ce serveur est généralement nommé X. De plus, un client window-manager , tel que twm , tvtwm , olwm , olvwm , mwm , ctwm , ou encore fvwm et des applications sont également disponibles.

Par exemple, le système Linux , disponible pour PC, permet de faire tourner XFree86 , implémentation libre de X Window.

DOS

Il faut ici se procurer un logiciel faisant office de serveur, les plus connus sont :

Xvision (VisionWare Soft, Inc), PC-Xware (NCD, Inc.), PC DECwindows (DEC), PC Xsight (Locus Computing), Micro X-Lite (StarNet Comm. Corp.), Xappeal (Xtreme), Xoftware (AGE Logic), PC X-Kit (Xlink)



X Window



Des logiciels supplémentaires (par exemple de couche TCP/IP) sont parfois nécessaires, comme Trumpet Winsock . Une rumeur non vérifiée prétend que Microsoft distribue une version redistribuable d'un serveur X dans les milieux académiques.

MS-WINDOWS

De la même façon que pour le dos, il faut se procurer un logiciel faisant office de serveur, les plus connus sont :

HCL-eXceed (Hummingbird Software), eXcursion (DEC), MultiView/X (JSB Corp.), PC-Xview (NCD Inc.), Xoftware (AGE Logic), eXodus (White Pine Software), ReflectionX (WRQ, Inc.)

2 - Configuration d'une session

2.1 - La gestion des login sous X-Window

Au boot du système, plus précisément dans un des scripts `/etc/rc*.*` , un processus nommé **xdm** (pour X Display Manager) est lancé pour gérer les connexions sous X-Window et permettre aux utilisateurs de lancer au login un minimum de clients X .

Ce processus **xdm** restera actif en permanence sur le serveur Unix. Il existe d'autres gestionnaires de connexions graphiques plus récents comme **kdm** ou **gdm**.

2.2 - Le fichier `$HOME/.xsession`

Ce fichier permet aux utilisateurs de lancer un certain nombre de clients X-Window dès le login, dans la mesure où ils ont réalisé une connexion gérée par **xdm**. Dans le cas contraire il faut utiliser le fichier `$HOME/.xinitrc`.

```
$cat $HOME/.xsession
xclock &
xterm -ls &
wmaker
$
```

La première ligne lancera le client X « xclock ».

La seconde ligne lancera le client X « xterm » : il s'agit en fait d'un terminal virtuel, qui apparaît sous la forme d'une fenêtre de 80 colonnes et 25 lignes, dans le coin gauche en haut de l'écran, dans laquelle s'exécute un shell. Ce shell sera considéré comme un shell de login puisque l'option `ls` « login shell » a été choisie d'où interprétation par ce shell des fichiers `/etc/profile` et `$HOME/.profile`.

La troisième ligne est le gestionnaire de fenêtre que nous allons voir juste après. Toutes les commandes sauf la dernière doivent être lancées en arrière plan.



X Window



Lorsque l'utilisateur met fin à l'exécution de la dernière commande, la session X11 se termine. (En l'occurrence ici, il s'agit de wmaker).

Exercice : Modifiez votre .xsession pour avoir la même chose qu'au dessus.

2.3 - Le gestionnaire de fenêtres

Un serveur X agit comme un multiplexeur de fenêtres. Pour disposer d'un système convivial, il faut pouvoir déplacer, iconiser, mettre au premier plan ... les fenêtres gérées par le serveur X : c'est le rôle du gestionnaire de fenêtres.

Les fenêtres ont un bandeau qui, en général, indique le nom de l'application (Xman, Xlatex 3.3 ...). Aux extrémités du bandeau on trouve des carrés à cliquer qui servent à iconiser ou retailler la fenêtre (à l'aide du pointeur de la souris). Les fenêtres peuvent également être déplacées ou exposées au premier plan en cliquant dans le bandeau. Le gestionnaire de fenêtres est un client X comme un autre.

Exercice :

- Supprimez la ligne wmaker de votre fichier .xsession (n'oubliez pas d'enlever le & sur la dernière ligne !!), alors ?
- Modifiez votre fichier pour avoir twm a la place de wmaker, alors ?
- Essayer maintenant xfwm.

3 - Configurer les clients X : Les Ressources

3.1 - Introduction

Dans l'environnement X-Window le mot ressource est utilisé pour spécifier des valeurs qui sont utilisées par les applications, et qui peuvent être facilement modifiées par l'utilisateur. Dans le principe il convient de distinguer les ressources standards (communes à toutes les applications) et les ressources particulières (inventées par le créateur de l'application).

3.2 - Les ressources standards

Toute application s'appuyant sur la bibliothèque Xt Intrinsics hérite des ressources suivantes :

- font pour spécifier les polices de caractères utilisées ;
- geometry pour indiquer le positionnement et la taille des fenêtres ;
- background pour qualifier le fond d'écran d'une fenêtre ;
- borderwidth pour donner la taille en pixels du contour des fenêtres ;
- display pour indiquer le nom de la station qui héberge le serveur X ciblé ;
- ...

Ces ressources sont appelées les ressources standards.

3.3 - Les ressources des applications

En plus des ressources standard, les programmeurs définissent des ressources propres à leurs applications. Ces ressources servent à personnaliser les utilisations, l'utilisateur ayant



X Window



la maîtrise de nombreux paramètres. Ainsi l'application xterm fournie par le consortium X prévoit pour l'utilisateur la possibilité de spécifier ses choix pour les ressources suivantes :

- scrollBar pour indiquer si l'utilisateur désire ou non la présence d'une barre de défilement sur le côté de sa fenêtre ;
- saveLines pour mentionner le nombre de lignes sur lequel portera l'effet d'une barre de défilement ;
- ...

Le manuel d'une application X-Window contient toujours la liste des ressources propres à cette application ainsi que leurs valeurs par défaut.

3.4 - Le nommage des ressources

Les ressources d'une application X-Window sont associées aux fenêtres qui la composent sous la forme d'une arborescence respectant les imbrications des fenêtres composantes. Les fenêtres composant une application ont un nom (utilisable à l'extérieur du programme) donné par le programmeur, il doit figurer dans le manuel de l'application.

On considère ici un exemple qui crée une fenêtre contenant deux sous-fenêtres qui sont des boutons de commandes. Pour personnaliser l'application exemple l'utilisateur pourra procéder de la façon suivante :

```
exemple.bouton1.font : 6x13
exemple.bouton1.borderWidth : 5
exemple.bouton1.label : Com A
exemple.bouton2.font : 9x15
exemple.bouton2.borderWidth : 5
exemple.bouton2.label : Com B
```

L'utilisateur a ici utilisé les ressources standard font et borderWidth , il demande une police de caractères spécifique pour chaque bouton de commande. Par contre pour le bord de la fenêtre il stipule la même valeur pour bouton1 et bouton2 et aurait pu utiliser la notation suivante :

```
exemple*borderWidth : 5
```

La notation * permet de s'affranchir de tout ou partie de la hiérarchie des fenêtres. L'exemple ci-dessus créerait toutes les fenêtres de l'application avec un bord de 5 pixels quel que soit leur niveau d'emboîtement. La seconde ligne commençant par Exemple cible toutes les applications de la classe Exemple dont exemple est un membre. C'est une notation fréquemment utilisée.

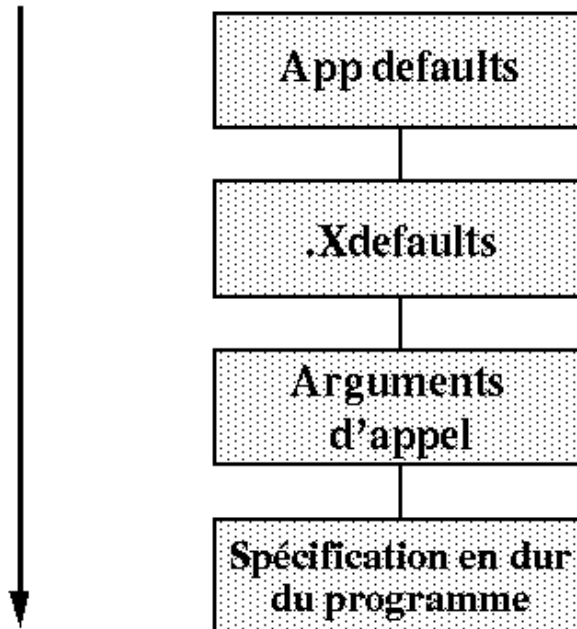
Dans le jargon X-Window on dit que exemple est une application et que exemple.bouton2 est un widget, label étant une ressource propre à l'application exemple. Les (sous) fenêtres ne sont que la partie visible des widgets, un widget est une structure de données accompagnant une fenêtre et permettant de la qualifier à l'aide de ressources.

3.5 - Où spécifier des ressources

Il y a divers endroits pour spécifier les ressources des applications X-Window. L'auteur d'une application utilisera un fichier de ressources qu'il placera dans le répertoire /usr/lib/X11/app-defaults, mais il pourra également spécifier dans son programme des ressources inaltérables.

L'utilisateur pourra personnaliser l'application en positionnant des ressources dans le fichier .Xdefaults situé dans son répertoire d'accueil. Il pourra également indiquer des ressources en arguments de la ligne de commande qui lance l'application ciblée. Lorsqu'une même ressource est citée dans plusieurs endroits, l'ordre de préséance est celui de la figure suivante :

**Les dernières
ressource
spécifiées
sont
prioritaires**



3.6 - Option sur la ligne de commande

On peut évidemment lancer un client X sur la ligne de commande avec les options que l'on a choisies:

```
xclock -bg green -fg tomato
```

pour tuer cette tâche lancée en avant plan, on utilise le bouton de fermeture de la fenêtre ou <ctrl C>

```
xclock -bg tomato -fg green &
```

pour tuer cette tâche lancée en arrière plan, on peut toujours utiliser le bouton de fermeture de la fenêtre.

Ces deux méthodes constituent plutôt des « sorties de secours » utilisées quand un client X pose des problèmes. En général, les clients X disposent d'un bouton ou d'un item dans un menu déroulant autorisant une sortie plus élégante avec notamment destruction des fichiers temporaires éventuellement créés.



X Window



Exercice : Essayez de changer la couleur d'affichage de l'application `xterm` sur la ligne de commande.

3.7 - Fichiers de ressources

Le système X-Window permet de choisir, à travers un mécanisme de ressources et de fichiers de ressources, les options qui seront automatiquement positionnées lors du lancement d'un client particulier. Cette configuration peut être réalisée directement dans le fichier **\$HOME/.Xdefaults** en suivant une syntaxe rigoureuse.

Une ligne est de la forme `<Nom application>*<option> :<valeur>` par exemple :

```
xclock.foreground :green.
```

Exercice

Lancer un client ***xclock*** sans option puis le tuer.

Editer le fichier ***\$HOME/.Xdefaults*** pour mettre la couleur d'avant-plan des horloges à *tomato*.

Lancer un client ***xclock*** sans option puis le tuer.

Lancer un client ***xclock -bg black -fg green*** puis le tuer.

On remarquera que les options choisies sur la ligne de commandes sont prioritaires par rapport à celles choisies dans le fichier ***\$HOME/.Xdefaults*** qui elles-mêmes sont prioritaires par rapport aux options par défaut.

3.8 - La commande ***xrdb***

`xrdb` permet de prendre en compte la personnalisation des ressources par un utilisateur. Cette commande utilise la syntaxe du préprocesseur du langage C pour permettre de présenter des fichiers de ressources adaptés à différents cas. `xrdb` connaît (entre autres) les symboles suivants :

- **COLOR**, qui est positionné si l'écran utilisé est répertorié du genre `StaticColor`, `PseudoColor`, `TrueColor` ou `DirectColor` (cf. la commande `xdpyinfo`) ;
- **SERVERHOST**, le nom de la machine hébergeant le serveur X ;
- **VENDOR**, une chaîne permettant d'identifier le fournisseur du serveur X utilisé (elle est hélas souvent très longue).

Dans un fichier de ressources, l'utilisateur pourra s'autoriser des écritures du genre :

```
#ifdef COLOR
    Mwm*background :    SlateGrey
    Mwm*foreground :    yellow
#endif
```

On utilise en général `xrdb` en lui fournissant le nom d'un fichier de ressources en argument comme dans l'exemple suivant :

```
xrdb /.Xresources
```




X Window



Le problème de cette commande est qu'elle interfère avec les autres mécanismes de ressources, on ne l'utilise donc que très rarement.

3.9 - Les instances d'une application

Le système X-Window permet aussi de nommer des instances d'applications et ces noms peuvent être utilisés dans les fichiers de ressources.

Exercice

Ajouter cette ligne au fichier *\$HOME/.Xdefaults*

```
gaston*background: seagreen
```

Lancer les commandes suivantes

```
xterm -name gaston -title 'Premier' &  
xterm -name gaston -title 'Second' &  
xterm -title 'Troisième' &
```

Tuer ces trois clients

3.10 - La taille et le positionnement des fenêtres

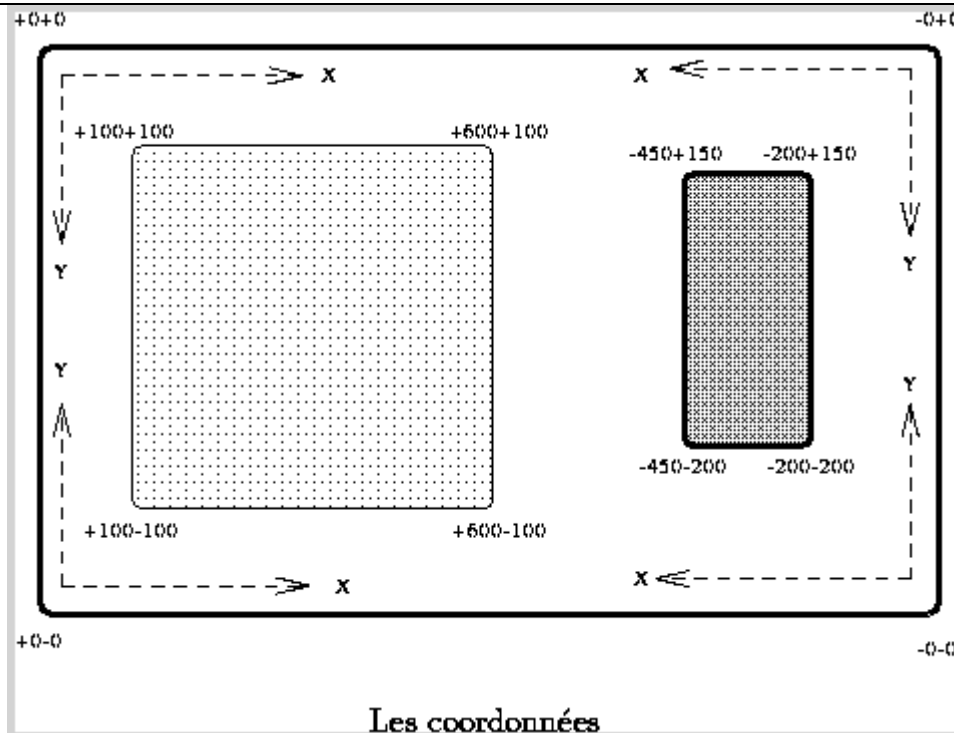
La ressource standard *geometry* permet de contrôler le positionnement et la taille des fenêtres sur l'écran. Cette ressource peut, bien sûr, être indiquée comme paramètre de la ligne de commande qui initialise une application au moyen de l'argument *-geo*. La forme générale de spécification de la ressource *geometry* est la suivante :

```
geometry largeur xhauteur +abscisse +ordonnée
```

En argument de la ligne de commande on peut utiliser :

```
xapplication -geo largeur xhauteur +abscisse +ordonnée
```

Les valeurs largeur, hauteur, abscisse et ordonnée s'expriment en pixels (en général). Les quatre coins de l'écran peuvent servir d'origine pour exprimer les coordonnées d'une fenêtre :



L'axe des X existe en deux versions, en fonction du sens de la flèche. On peut donc indiquer des valeurs positives ou négatives pour l'abscisse d'un point. De même l'axe des Y est orienté dans les deux sens et des valeurs positives ou négatives peuvent être stipulées pour l'ordonnée d'un point. Pour indiquer la situation d'une application, on ne stipule que les coordonnées d'un seul point, celui-ci est compris comme étant le coin supérieur gauche si les deux coordonnées sont positives, comme étant le coin supérieur droit si seule la valeur de l'abscisse est négative, ...

On peut bien sûr indiquer les coordonnées d'une fenêtre sans préciser sa taille, ou inversement :

`xclock -geo -0+0` -> horloge dans le coin supérieur droit de l'écran

`xedit -geo 600x300` -> une fenêtre xedit de 600 pixels par 300 pixels

La taille des fenêtres s'exprime en général en pixels, toutefois quelques applications comme xterm l'expriment en caractères (et la taille en pixels dépend alors de la police de caractères utilisée) :

`xterm -geo 80x40` -> une fenêtre xterm de 80 caractères par 40

Il peut être intéressant de connaître la dimension et les coordonnées d'une fenêtre visualisées par un serveur X (pour les répercuter dans un `.xinitrc`, `.Xresources` ...). Une application est particulièrement intéressante pour cela : `xwininfo`, qui est fournie par le consortium, et demande à l'utilisateur de cliquer dans la fenêtre d'une application et donne, entre autres résultats, les coordonnées de celle-ci :

```
xwininfo : Please select the window about which you
           would like information by clicking the mouse in that window.
xwininfo : Window id : 0x340004f "Xlatex 3.3"
```

Corners : +267+56 -305+56 -305-460 +267-460
 -geometry 580x384+262+27

Exercice : Modifiez votre fichier `.xsession` pour que l'application `xclock` apparaisse au démarrage d'une session en bas à droite.

4 - La couleur

4.1 - Les principes de bases

Les principes de création de couleurs sont un peu surprenants pour les non initiés. Avec seulement 3 couleurs de base (rouge, vert et bleu), on arrive à créer un nombre très important de couleurs. Par exemple, en mélangeant du rouge et du vert on obtient du jaune, du blanc pouvant être obtenu en mélangeant à dose égale du rouge, du vert et du bleu.

Un serveur X est capable de gérer une colormap par fenêtre, même si la mémoire vidéo d'un terminal ne peut en contenir qu'une seule, et, en passant le pointeur de la souris d'une fenêtre à l'autre, on peut assister à des effets de couleurs surprenants. X fournit une colormap par défaut, il est du ressort des applications d'en bâtir d'autres.

4.2 - Le nommage des couleurs

X supporte le nommage symbolique des couleurs et l'utilisateur peut citer `blue`, `green`, ... à la place du codage RGB (Red,Green,Blue sur 24 bits). La valeur du code correspondant est trouvée dans une colormap, qui par défaut, peut être bâtie à partir du fichier `/usr/lib/X11/rgb.txt` dont voici un extrait :

Nom symbolique	Red	Green	Blue
white	255	255	255
black	0	0	0
gray	190	190	190
light gray	211	211	211
blue	0	0	255
royal blue	65	105	255
dark green	0	100	0
yellow	255	255	0
gold	255	215	0
beige	245	245	220
orange	255	165	0

...

Les terminaux X arrivent en général avec leur propre fichier `rgb.txt` qui peut différer de celui fourni par le consortium. Il est du ressort d'un administrateur système de tenter de faire converger les différents `rgb.txt` fournis (consortium, fabricants de terminaux X) de façon à proposer une table des couleurs homogène pour un site donné (au moins au niveau des noms symboliques).



X Window



Les noms symboliques de couleurs pourront être utilisés par l'utilisateur dans le nommage des ressources de type background, foreground ou bordercolor, la conversion en valeur de pixel sera faite automatiquement comme dans l'exemple suivant :

```
xterm -bg cyan -fg DarkSlateBlue
```

Exercice : Modifier le fichier .xsession pour que l'application xclock prenne les couleurs que vous aurez choisis dans rgb.txt.

5 - Les polices de caractères

5.1 - Introduction

L'introduction massive des terminaux bitmaps et des imprimantes lasers a conduit à l'émergence de nouvelles techniques pour représenter les caractères que nous visualisons. Les polices de caractères (fonts) héritées de la typographie sont maintenant d'usage courant dans l'environnement informatique. Ainsi les caractères n'ont plus un aspect figé comme du temps des terminaux ascii , leurs représentations dépendent de la famille de police utilisée. La ressource standard font permet de choisir les polices de caractères utilisées pour toutes les applications X qui manipulent du texte.

5.2 - Le nom des polices de caractères sous X-Window

Plusieurs champs rentrent dans la composition du nom d'une police de caractères, ils sont séparés entre eux par le caractère - (tiret). Le nom d'une police vue par X-Window est complexe pour l'utilisateur, mais d'une part, il garantit l'unicité dans le nommage et, d'autre part, il fournit aux spécialistes un aperçu rapide des caractéristiques de la police ciblée. Les principaux champs constitutifs du nom d'une police sont les suivants :

- FOUNDRY : c'est le nom de la société ou organisation qui a fourni la police (le "fondeur"). ADOBE est un exemple connu ;
- FAMILY : identifie la famille de police utilisée, Helvetica et Times sont des exemples. Les noms de familles de polices sont en général hérités de la tradition typographique ;
- WEIGHT-NAME : spécifie la graisse de la police, en général pour une famille on dispose de deux types de graisse : bold (gras) et medium (normal);
- SLANT : indique le positionnement des caractères de la police, les possibilités les plus connues étant Roman et Italic (r ou i) qui donnent respectivement des caractères droits ou penchés ;

Un certain nombre d'autres champs précisent la taille du corps de la police en pixels, et font appel à des connaissances typographiques.

Exemples de nom de police :

```
-adobe-courier-bold-r-*-_*_*_*_*_*_*_*_*_*_*
```

```
-adobe-courier-bold-r-normal-*-19-*_*_*_*_*_*_*_*_*_*
```



X Window



5.3 - Localisation des polices de caractères et facilités de nommage

Les polices de caractères se trouvent en général dans le répertoire `/usr/lib/X11/fonts` qui contient des répertoires par groupes de polices (100dpi, 75dpi, PEX , ...), le répertoire 75dpi regroupant toutes les fontes dont la résolution est de 75 points par pouce. Un mécanisme d'alias permet de faciliter le nommage des polices de caractères les plus fréquemment utilisées. Dans les répertoires se trouvant sous `/usr/lib/X11/fonts` un fichier nommé `fonts.alias` contient une correspondance entre un nom d'alias et le nom étendu de la police de caractères. L'exemple ci-dessous montre un sous-ensemble du contenu du fichier `fonts.alias` se trouvant dans `/usr/lib/X11/fonts/misc` :

```
7x13 -misc-fixed-medium-r-normal--13-120-75-75-c-70-iso8859-1
7x13bold -misc-fixed-bold-r-normal--13-120-75-75-c-70-iso8859-1
7x14 -misc-fixed-medium-r-normal--14-130-75-75-c-70-iso8859-1
7x14bold -misc-fixed-bold-r-normal--14-130-75-75-c-70-iso8859-1
8x13 -misc-fixed-medium-r-normal--13-120-75-75-c-80-iso8859-1
8x13bold -misc-fixed-bold-r-normal--13-120-75-75-c-80-iso8859-1
8x16 -sony-fixed-medium-r-normal--16-120-100-100-c-80-iso8859-1
9x15 -misc-fixed-medium-r-normal--15-140-75-75-c-90-iso8859-1
```

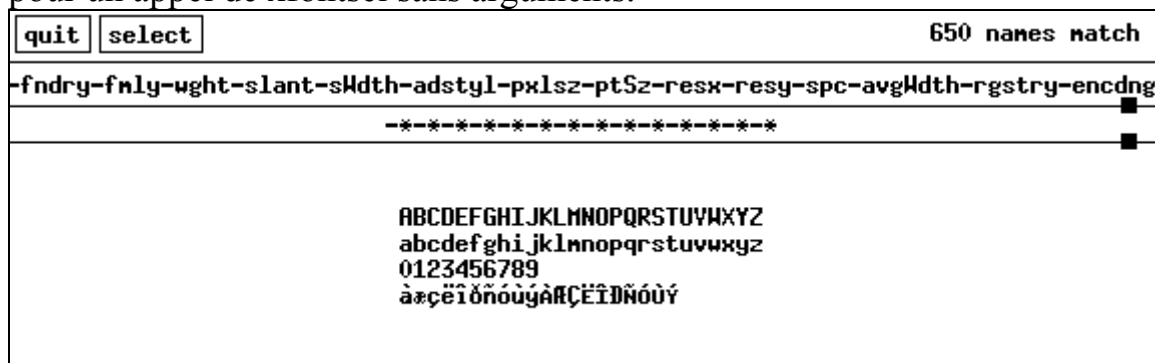
Il est maintenant possible d'utiliser le nom d'alias en lieu et place du nom complet d'une police de caractère.

Exercice : Essayez la ligne ci dessous

```
xterm -fn 8x16
```

5.4 - Choisir une police de caractères

L'application `xfontsel` fournie par le consortium X permet de visualiser le rendu des polices accessibles par un serveur X, les figures suivantes donnent un exemple d'utilisation pour un appel de `xfontsel` sans arguments.



La fenêtre de `xfontsel` est découpée en 4 parties :

les deux boutons de commande `quit` et `select` et une zone de commentaires `650 names match` ici ;

- une barre avec des prises de menus déroulants, un menu par éléments constitutifs du



X Window



nom d'une police ;

- une zone contenant le nom de la police visualisée ;
- le rendu de la police dont le nom figure dans la zone précédente ;

Le champ spc du bandeau de menu de l'application xfontsel permet de choisir entre les polices fixes (où tous les caractères ont le même encombrement) et les polices variables. Les polices fixes sont obtenues en choisissant l'item m, les polices variables en choisissant l'item p. En sélectionnant une police pour l'application xterm, l'utilisateur fera attention de ne sélectionner que des polices fixes car xterm ne sait pas s'accommoder des polices variables ;

Exercice : Exécutez cette application et testez les différentes polices.

6 - xterm

xterm est l'émulateur de terminal de X-Window. Les options de démarrage de xterm sont très nombreuses, les plus importantes sont les suivantes :

- cr couleur pour indiquer la couleur du curseur de texte. Par défaut il utilise la même couleur que pour l'affichage du texte.
- l indique à xterm de créer un fichier trace (log) de la session en cours. Le fichier trace créé s'appellera Xterm. numéro_de_processus et sera situé dans le répertoire d'accueil de l'utilisateur (sauf s'il utilise l'option -lf suivie d'un nom de fichier) ;
- +ls indique que le shell déroulé n'est pas un shell de connexion mais un simple sous shell. Les fichiers .login ou .profile ne sont pas consultés. L'option (-ls) indique à xterm d'initialiser des shells de connexion. La valeur de l'option par défaut est déterminée par la valeur de la ressource XTerm.VT100.LoginShell;
- ms couleur pour indiquer la couleur du pointeur de la souris. Par défaut il prend la même couleur que celle utilisée pour afficher le texte ;
- +sb pour demander à xterm de ne pas placer une barre d'ascenseur sur le côté gauche de la fenêtre. L'option -sb crée une barre d'ascenseur qui porte sur les 64 dernières lignes affichées (l'option -sl nombre_de_lignes permettant de s'assurer un ascenseur plus important). La valeur par défaut de l'option sb est déterminée par la valeur de la ressource XTerm.VT100.ScrollBar
- e commande pour indiquer à xterm de lancer automatiquement une application qui se visualisera dans la fenêtre xterm. Il est très important que l'argument -e soit le dernier indiqué sur la ligne de commande.

Quelques options standard de X-window sont d'un intérêt évident pour xterm :

- bg couleur pour stipuler la couleur de fond de la fenêtre ;
- fg couleur pour indiquer la couleur du texte de la fenêtre ;
- title titre pour positionner le titre de la fenêtre ;
- geometry pour indiquer la taille de la fenêtre;



X Window



-name nom de l'application par laquelle on cherchera les ressources.

...

Exercice

On va remettre un peu d'ordre dans nos deux fichiers \$HOME/.xsession et \$HOME/.Xdefaults de façon à obtenir l'environnement suivant dans nos sessions X-Window :

Au login une horloge dans le coin gauche en bas de l'écran.

Au login une fenêtre xterm avec un shell « de login » dans le coin droit en haut de l'écran avec 80 colonnes et 25 lignes et comme titre « xterm avec shell de login ».

Toutes les instances d'applications xclock ont la couleur CadetBlue en background et la couleur DarkKhaki en foreground.

Toutes les instances de l'application xterm ont la couleur black en foreground.

7 - Les outils liés au serveur X

7.1 - xdpinfo

La commande xdpinfo donne des informations sur poste de travail qui héberge un serveur X. Elle fournit (entre autres) le nom du display, l'identification du distributeur du serveur X, la résolution de l'écran, ...

Exercice : Essayez cette commande. Quelle est la résolution ?

7.2 - xkill

La commande xkill permet de supprimer un client en lui expédiant un KillClient. xkill sans arguments, invite l'utilisateur à choisir la fenêtre indésirable en cliquant dedans. xkill -all demande la suppression de tous les clients connus par un serveur X, ce n'est pas vraiment la façon conseillée pour quitter une session.

Exercice : Lancez un xterm en arrière plan et utilisez xkill pour le tuer.

7.3 - xmodmap

xmodmap est l'outil de base pour définir (ou redéfinir) tout ou partie d'un clavier. Il permet d'associer à une touche du clavier une liste de symboles (keysyms) appartenant à un character set (Latin1 pour nous). Chaque touche du clavier renvoie un code unique (keycode) (dont la valeur est comprise entre 0 et 255), et il est facile de l'identifier. La liste des keycodes d'un clavier est en général donnée par la documentation du fournisseur. En général on appelle xmodmap de la façon suivante : xmodmap fichier . Nous n'entrerons pas dans les détails ici.

7.4 - xset

La commande `xset` permet de gérer quelques préférences de l'utilisateur à propos du clavier, de la souris et de l'écran. Elle autorise également la personnalisation des chemins d'accès aux polices de caractères. Les paramètres qui peuvent être ajustés par `xset` sont les suivants :

- le son de la sonnette du terminal ;
- le bruit des touches du clavier ;
- la répétition automatique des touches du clavier ;
- les led du clavier ;
- la vitesse de déplacement du pointeur de la souris ;
- les chemins d'accès aux polices de caractères ;

Mais certains paramètres nécessitent d'être sur une vraie station X et non pas sur un serveur X sous windows.

Exercice : Exécutez cette commande.

7.5 - xsetroot et wmsetbg

La commande `xsetroot` permet de garnir le fond d'écran par un bitmap approprié. Celui-ci peut être un fond gris (par défaut), un fond grillagé (option `-mod`) ou un fond personnalisé (option `-bitmap`) comme dans les exemples suivants :

`xsetroot` (fond gris)

`xsetroot -mod 25 25` (grillage de 25 sur 25)

`xsetroot -bitmap mesbitmap/machin.xbm` (un bitmap quelconque)

`wmsetbg` quand à lui permet des choses plus élaborées, comme par exemple mettre une image jpeg en fond d'écran.

Exercice : Essayez les différentes options de cette commande.

Testez maintenant `wmsetbg`. (pour les options `man wmsetbg`), vous pourrez trouver quelques images jpg dans le répertoire `/usr/share/wallpapers`.

8 - Droits d'accès au serveur X

8.1 - Le principe

Le serveur X-Window, c'est à dire le processus qui s'exécute sous Windows95 sur votre PC, peut accepter ou refuser les requêtes provenant des clients X. Ce comportement peut être modifié grâce au client X nommé **xhost** qui doit s'exécuter sur l'ordinateur où le serveur X est actif.

Pour une station de travail ou pour un terminal X dialoguant avec un serveur Unix tout se règle donc avec cette commande **xhost**.

Pour nos PC émulés en station X-Window, c'est un peu plus compliqué car cela dépend de la configuration du logiciel d'émulation :

Premier cas de figure

Au niveau du logiciel Exceed, on a inhibé le contrôle d'accès par liste : seuls les clients X lancés depuis le PC où s'exécute le serveur X sont autorisés à dialoguer avec lui : situation figée et sécurité maximum

Deuxième cas de figure

Au niveau du logiciel Exceed, on a validé le contrôle d'accès par liste : des clients X lancés depuis d'autres hôtes du réseau pourront dialoguer avec le serveur X-Window et donc afficher sur l'écran qu'il gère : liste des hôtes autorisés gérée par l'administrateur Exceed et sécurité moyenne

Troisième cas de figure

Au niveau du logiciel Exceed, on a validé le contrôle d'accès par liste et on a autorisé l'utilisateur du PC à modifier lui même la liste le temps d'une session: des clients X lancés depuis d'autres hôtes du réseau pourront dialoguer avec le serveur X-Window et donc afficher sur l'écran qu'il gère : liste modifiée avec le client **xhost**, meilleure convivialité mais sécurité moindre

Utilisation de la commande xhost

xhost		liste des hôtes autorisés
xhost	+	autorisation globale
xhost	-	interdiction globale
xhost	+zebulon	ajout de l'hôte zebulon à liste des hôtes autorisés
xhost	-zebulon	retrait de l'hôte zebulon de la liste des hôtes autorisés

Exercice

Sur un même écran, ouvrir deux sessions :

la première est une session telnet classique, la seconde est une session X

Sur session telnet, affichez le contenu de la variable DISPLAY.

Sur session X, affichez le contenu de la variable DISPLAY.

Sur la session telnet, exécutez un xterm, possible ?

Sur la session X, exécutez un xterm, possible ?

Sur la session telnet, modifiez ... a votre avis ?

Exécutez un xterm sur la session telnet.

Exécutez un xterm sur la session telnet avec l'affichage sur l'écran de votre voisin.

Grâce à la commande import capturez l'affichage de la machine de votre voisin.

9 - Les environnements

Les système X window a suivi des évolutions majeures depuis ces dernières années, essentiellement pour les postes de travail, en effet Xfree86 supporte maintenant l'accélération matérielle OpenGL par exemple. De plus des environnements ont fait leur



X Window



apparition comme ceux que nous venons de voir : wmaker et xfce, mais aussi des beaucoup plus élaborés comme gnome et kde que vous allez tester.

Exercice

Gnome

Modifiez votre `.xsession` et remplacez son contenu par l'unique ligne :

`gnome-session`

Testez cet environnement.

KDE

Modifiez votre `.xsession` et remplacez son contenu par l'unique ligne :

`startkde`

Testez cet environnement.

Attention : Ces environnements sont relativement lourds pour le système, donc une fois testés, remettez un gestionnaire de fenêtre plus léger (wmaker ou xfwm).

10 - Quelques clients X

On vous demande ici, de tester ces différents programmes.

Des yeux : **xeyes**

Editer des textes :

xedit : Très simple, pas très esthétique et moins puissant que vi. C'est ce que peut donner une application basée sur la librairie Toolkit (Xt) du projet ATHENA.

nedit : Très simple, esthétique et puissant. Basé sur son propre toolkit.

gedit : Basé sur les mêmes librairies que gnome (GTK)

Capter et afficher tous les événements traités par le serveur X avec **xev**

Connaître les clients actifs et leur positionnement à l'écran : **xlsclients**

Indicateur de nouveau courrier électronique : **xbiff**

Calculatrice : **xcalc**

Voir ou traiter une image : **gqview**, **kview**, **electric eyes (ee)**, **gimp**, ...

Utilitaires Window Maker : **wmmail** et **wmtime**

Lecteur de courrier plus complets : **balsa** et **kmail**