

Le système graphique X

Matthieu Herrb

Fondation X.Org



Toulibre, 20 mai 2009

Agenda

- 1 Présentation générale
- 2 Architecture
- 3 Affichage
- 4 Entrée
- 5 Développement de X



Agenda

- 1 Présentation générale
- 2 Architecture
- 3 Affichage
- 4 Entrée
- 5 Développement de X



Introduction

- Le système multi-fenêtres X est à la base des interfaces graphiques pour Linux et autres Unix.
- Composant de bas-niveau souvent mal connu des utilisateurs (normal) et des développeurs...



Présentation

- Développé dans les années 80 pour les stations de travail Unix.
- Indépendant du matériel et du système d'exploitation.



Agenda

- 1 Présentation générale
- 2 Architecture**
- 3 Affichage
- 4 Entrée
- 5 Développement de X

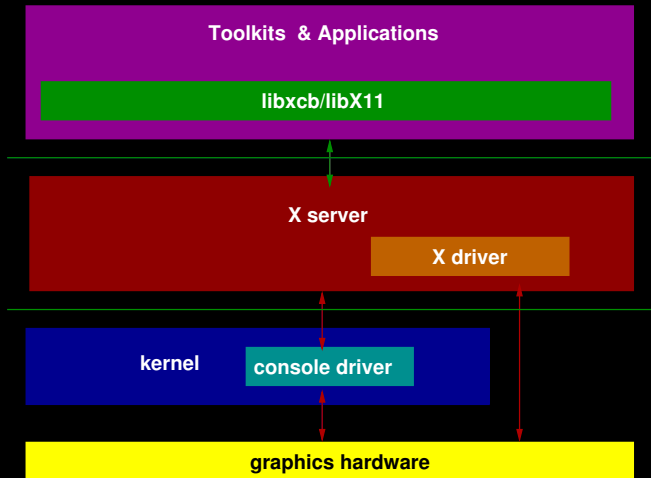


Composants d'un bureau Unix

- noyau
- serveur X
- applications :
 - window manager (décoration fenêtres)
 - file manager (affichage des icônes sur le bureau)
 - task bar, menu bar (affichage des barres de menu & état)
 - applications proprement dites (OpenOffice, Eclipse, Firefox, Thunderbird, The Gimp...)
- protocoles :
 - ICCCM (Inter-Client Communication Convention Manual)
 - DBUS/HAL
 - XDND (X Drag and Drop)
 - Corba (Gnome)
 - DCOP (KDE)



Composants



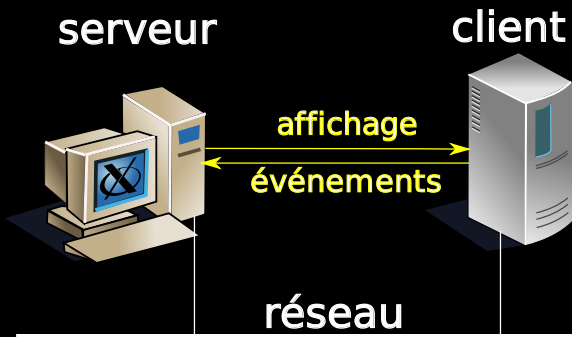
Client/Serveur

serveur : gère le matériel graphique : carte video, écran, clavier, souris, etc.

clients : applications qui affichent des choses et récupèrent des événements sur les périphériques d'entrée.



X : Client/Serveur



Fonctionnalités du serveur X

Graphique de bas niveau :

- fenêtres rectangulaires organisées en arbre,
- tracés 2D simples,
- texte (obsolète),
- etc.

Pas de menus, décorations ou autres → toolkits



Extensions

Ajoutent des fonctionnalités au protocole de base.
Nombreuses extensions obligatoires de nos jours : Render,
Composite, XKB, DRI, etc.)



Architecture du serveur

DIX : Device Independent X

DDX : Device Dependant X

(répertoire xserver/hw + drivers)



Sécurité : problématiques

- Le serveur X tourne avec les privilèges root (accès au matériel) et exécute des requêtes de clients moins privilégiés
 - filtrer les requêtes pour ne pas exécuter du code malveillant.
- Un client peut tourner sur une machine distante non digne de confiance
 - éviter que de donner accès à un client à des données d'autres clients via les protocoles inter-clients.



Sécurité : mécanismes

Contrôle d'accès au serveur / Authentification des clients

- Adresse IP (xhost)
- Magic cookie (xauth)
- Kerberos, Secure RPC,... (peu déployé)

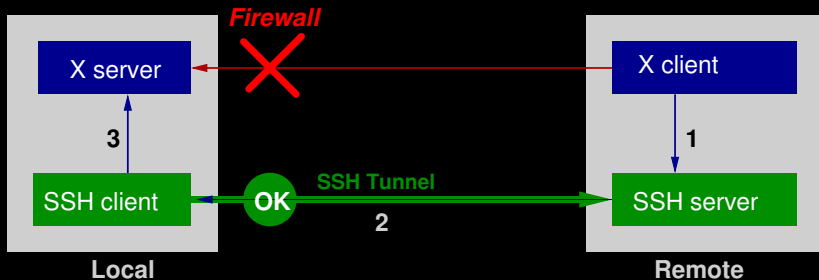
Contrôle de l'accès aux ressources :

- XSecurity (obsolète) clients trusted / untrusted
- XACE : framework pour implémenter des politiques, pas utilisable seul.



Proxys et compression

Exemple : proxy SSH



Serveurs X spéciaux

- rootless : Xmingw, XDarwin serveurs pour d'autres systèmes multi-fenêtres.
- poupées russes : Xnest, Xephyr serveurs qui affichent dans une fenêtre X. Applications : test, plugging X pour navigateur web (XRX).
- Proxies : lbxproxy (obsolète) NX server : compression de connexions distantes.
- Virtuels : Xvfb, Xprint : affichage dans un pixmap.
- DMX : serveur multi-écrans distribué (mur d'images à partir de plusieurs PCs)
- ...



Agenda

- 1 Présentation générale
- 2 Architecture
- 3 Affichage**
- 4 Entrée
- 5 Développement de X



Rendu classique

Requêtes de base du protocole X :

- `XDrawLine`, `XDrawRectangle`, `XDrawArc`, etc. : tracés élémentaires.
- `XLoadFont`, `XDrawString`, `XDrawText`, etc. : affichage de texte.

Tracés au pixel près.

Gestion des polices de caractères coté serveur.



Render

- Composition selon Porter & Duff dans une fenêtre.
 - Notion de canal α
 - Permet la transparence, l'anti-aliasing
- Implémenté dans la bibliothèque **pixman**.
- Affichage du texte coté client (**libXft2**)
 - utilise **fontconfig** et **FreeType2**.
 - Anti-alias
- Accélération matérielle possible sur les chipsets modernes.

Dans les applications : utiliser **Cairo** et **Pango** :
API de plus haut niveau, plus puissantes.



Accélération : XAA et EXA

Utiliser les possibilités du matériel pour accélérer les tracés.

XAA X Acceleration Architecture

défini par XFree86 4.x pour des cartes essentiellement 2D.

implémente chaque tracé élémentaire sur le matériel si possible,

sinon repli sur les routines indépendantes du matériel. (**mi**).

EXA Excellent Acceleration Architecture

nouvelle conception plus récente, adaptée à Render & Co.

Accélère surtout les copies de/vers la mémoire vidéo.

Nécessite une gestion de la mémoire vidéo.



Affichage de séquences vidéos

Problème difficile...

- format compressés (MPEG)
- codage couleur spécifique (YUV...)
- redimensionnement
- support matériel dans certaines cartes

Gérés par extensions XVideo et XvMC.

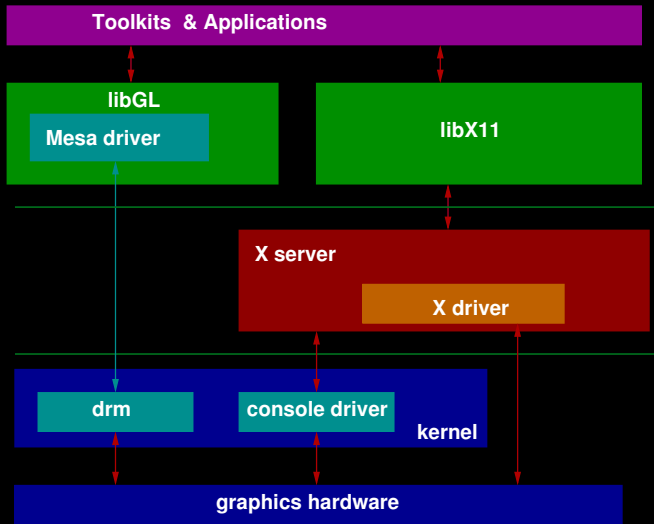


Accélération 3D : OpenGL & DRI

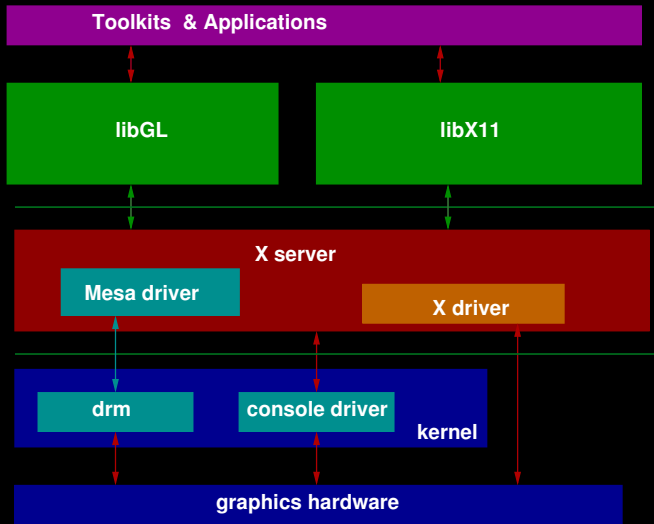
- Mesa3D : implémentation libre d'OpenGL
- DRI : Direct Rendering Infrastructure - permet l'accélération matérielle d'OpenGL sur certaines cartes par accès direct au matériel.
- GLX : Extension X qui permet de transporter des requêtes OpenGL
- AIGLX : Accelerated Indirect GLX



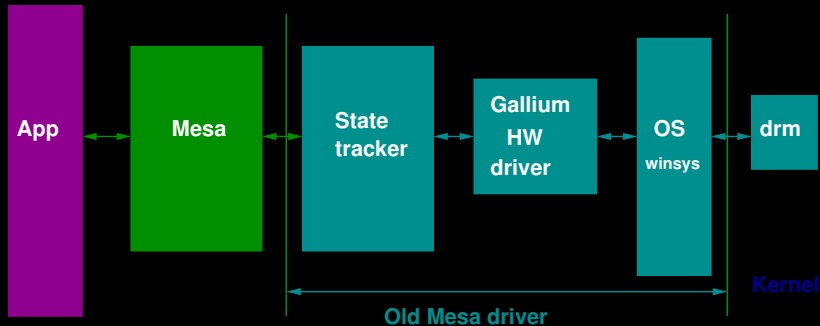
Direct rendering



Indirect rendering



Gallium



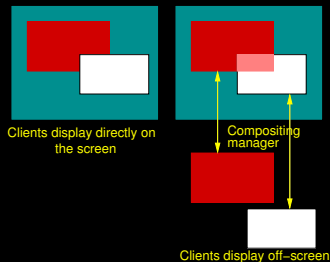
- Rendre DRI plus simple et plus proche du matériel actuel
- Les “state trackers” indépendants du matériel implémentent OpenGL.



Composite

Nouveau modèle pour l'affichage du bureau :

- Les fenêtres sont calculées d'abord hors de l'écran
- puis composées sur l'écran par le compositing manager.



Compiz : fenêtres 3D

Compiz est à la fois :

- un gestionnaire de fenêtres
- un gestionnaire de composition

Utilise OpenGL pour permettre de composer les fenêtres en 3D (cube), ou avec des effets (zoom, modification de couleurs, etc.).



Gestion des modes graphiques

Problème difficile dans l'architecture PC...

- caractéristiques des moniteur(s) (fréquences de balayage, résolution native, etc.)
- possibilités de la carte graphique et des drivers X.

⇒ trouver la liste ordonnée des modes à utiliser.

- fichier `xorg.conf`
- autoconfiguration
- configuration dynamique : **XRandR**



Autoconfiguration

Identificateur PCI de la carte graphique \Rightarrow section **Device** de la configuration.

DDC Display Data Channel
protocole VESA permettant de récupérer l'information EDID d'un moniteur via un bus I²C sur le connecteur VGA.

EDID Extended Display Identification Data
données décrivant un écran.

GTF Generalized Timing Formula
Formule fournie par VESA pour calculer un mode graphique.

\Rightarrow sections **Monitor** et **Screen** de la configuration.



X Resize and Rotate

- Redimensionner et orienter dynamiquement la fenêtre racine (le fond)
- Informer les applications des changements
- Gérer les sorties multiples d'une carte graphique (1.2)
- Gérer le « pan and scan » des anciens drivers (1.3)



Kernel Mode Setting

Programmation des modes graphiques dans le noyau au travers de DRI.

- moins d'accès direct du serveur X au matériel
- meilleure cohérence avec la console texte : disparition des flashes de l'écran lors du démarrage de X.
- possibilité de reprise après un crash du serveur X

Implémenté dans le noyau Linux 2.29 et dans certains drivers récents.



Gestion de la mémoire video

GEM - Graphical Execution manager

- module noyau
- gère les allocations dans la mémoire de la carte graphique
- caches de pixmaps, zones de travail pour GPU, etc.



Shatter

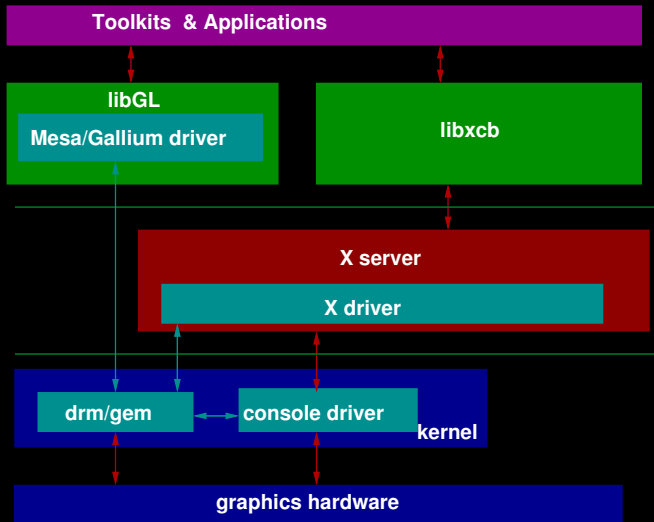
But : permettre le support de plusieurs cartes graphiques dans X moderne.

Principe : permettre de découper un pixmap en morceaux indépendants.

Implémentation en cours...



Architecture future DRI2 + Gallium + KMS



Agenda

- 1 Présentation générale
- 2 Architecture
- 3 Affichage
- 4 Entrée**
- 5 Développement de X



Périphériques d'entrée : l'autre partie du bureau...

- Histoire longue et complexe
- Composants :
 - Noyau,
 - Serveur X,
 - Bibliothèques,
 - Applications.
- Nouvelles génération de périphériques : touch-pads, écrans tactiles, multi-touch etc.

→ abstraction : input events : keys, button, valuator,....



Le monde du PC



Trois générations :

- contrôleur clavier PC/XT et AT. Souris séparée (RS/232 ou protocole propriétaire).
- interface type PS/2 : gère clavier et souris de façon commune. Pas connectable à chaud.
- USB : classe de périphérique spécifiques pour périphériques d'entrée. Connectable à chaud.

Les premières générations d'Unix pour PC n'avaient pas ou peu de fonction de gestion des périphériques d'entrées. Tout était implémenté dans XFree86.



Systèmes d'aujourd'hui

- Claviers et souris USB presque partout.
- D'avantage de traitement dans le noyau (driver evdev pour Linux).
- Connectable à chaud → configuration dynamique.



Nouveaux matériels

Écrans tactiles de plus en plus répandus.
Nouvelles technologies développent les possibilités des écrans tactiles :



Problématiques spécifiques des écrans tactiles

- écran et surface d'entrée physiquement liés.
- nécessité de calibration.
- pointeur : visible ou pas ?
- pas d'événement de déplacement seul : click ou drag uniquement.
- stylo ou doigt ?
- multi-touch...



Entrée : protocole X traditionnel

- Un clavier (avec pas trop de touches)
- Une souris, avec 5 boutons maximum.



Gestion du clavier

- DIX attend des événements « up / down » utilisant des keycodes pour représenter les touches.
- keycodes sont traduits en keysyms et envoyés aux clients.
- répétition automatique des touches est gérée soit par logiciel dans le serveur X, soit en laissant faire le matériel.
- Modifiers (Shift, Control, Alt, Meta, Super) génèrent des événements distincts qui modifient l'interprétation des événements par les clients.
- libX11 (coté client) traduit les séquences de keysyms en chaînes de caractères (ASCII, Latin-1, UTF-8 ou autre).



Souris, touchpad (coordonnées relatives) :

- PS/2, USB gérés par le noyau et les drivers evdev, mouse ou synaptics
- série gérés entièrement par le driver mouse.

Tablettest, écrans tactiles (coordonnées absolues) :

- soit gérées directement par le noyau (evdev)
- soit par un driver X spécifique.



Le driver Linux evdev

Les noyaux Linux récents implémentent un modèle d'événement générique, similaire à celui de Solaris ou de BSD.

Utilise HAL pour gérer la connexion à chaud des périphériques et leur configuration à la fois pour la console et pour X.



L'extension Xinput

But : supprimer les restrictions dans la gestion des périphériques d'entrée :

→ plusieurs claviers, souris, tablettes, etc. en parallèle.

- Xinput n'est plus optionnelle
Les drivers clavier/souris actuels dépendent de Xinput.
- Xinput est en cours de réécriture → Xinput2
- support pour la connexion à chaud, les propriétés attachées aux périphériques, MPX...



Propriétés pour les périphériques d'entrée

Données de type « clé / valeur » arbitraires attachées à un périphérique, modifiables à l'exécution via le protocole X.

Exemples d'application

- réglage de l'émulation du bouton du milieu,
- configuration de la stratégie d'accélération du pointeur,
- données de calibration pour les écrans tactiles,
- etc.

→ à partir du serveur version 1.6.

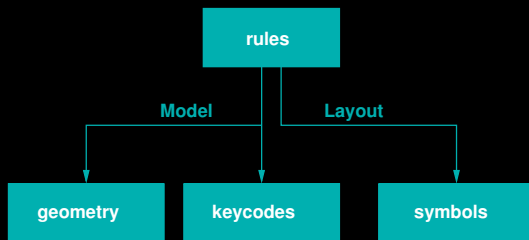


L'extension XKB

But : rendre la cartographie des touches du clavier configurable.

Système très complexe qui permet de gérer trop de choses.

Modèle RMLVO - Rules Model Layout Variants Options



XKB : Plein de bugs et de code inutile.

Travail en cours :

- nettoyage du code,
- fusionner la fonctionnalité de xkbcomp dans le serveur,
- supprimer tout le code pré-XKB.



Multi Pointer X

- Pointeurs & curseurs virtuels
 - attachés à un ou plusieurs dispositifs physiques
 - fournissent les événements aux applications
- Claviers virtuels - focus
 - attachés à zéro ou plus claviers physiques
 - fournissent les événements aux applications

→ prévu pour la prochaine version du serveur X.



Xaw/Motif/Gnome/KDE/...

- interprètent les événements X et les convertissent en actions
- gèrent les double-clicks, transformations d'événement de la molette en commandes défilement, etc.
- assurent la gestion de la sémantique des gestes en multi-touch (zoom, rotation, etc.)



Multi-head / multi-seat

Comment associer les périphériques d'entrée et de sortie...

- Configuration populaire là où il n'y a pas trop d'argent.
- Problèmes spécifiques de configuration, de gestion de ressources.

Il existe du code pour gérer ce type de configuration, mais c'est un peu du bricolage...



Agenda

- 1 Présentation générale
- 2 Architecture
- 3 Affichage
- 4 Entrée
- 5 Développement de X**



Modularisation

Depuis 2005, le développement de X est modulaire.

Plus de 300 modules classés en :

proto fichiers d'entête décrivant le protocole X et les extensions.

lib bibliothèques clientes.

app applications (demos et tests de base)

xserver les serveurs X (Xorg, Xnest, Xvfb, kdrive).

data données (bitmaps, définitions de clavier]

doc documentation



Développement des modules

- Chaque module a son propre numéro de version et son propre cycle de vie.
- Utilisation généralisée des autotools à la place de imake pour la compilation.
- Utilise Git pour la gestion de version.
- Hébergés sur freedesktop.org



Externalisation

X.Org ne gère plus les composants extérieurs qui ont leurs propres équipes de développement (mais relations et coopération) :

- Freetype2
- fontconfig
- Mesa
- pciaccess
- pixman
- xterm
- XKB data



L'an dernier : X.Org 7.4

Septembre 2008 - Basé sur xserver 1.5.

- Utilisation de la bibliothèque **pciaccess** pour les accès au bus PCI
- XACE remplace l'extension X Security
- Generalisation de XRandr 1.2 dans les drivers
- Driver **evdev** pour les périphériques d'entrée sous Linux.
- Driver **radeonhd** (basé sur les docs diffusées par AMD/ATI)



Xserver 1.6

Février 2009 - Release d'un serveur intermédiaire, avant X.Org 7.5

- Input device properties
- DRI2
- XRandR 1.3
- nouvelle accélération du pointeur



Prochaine version : X.Org 7.5

Basée sur xserver 1.7. Prévues pour juillet 2009.

- MPX
- Mesa 7.5 avec Gallium
- Xinput2
- support E-EDID
- suppression d'extensions obsolètes (Xprint)
- Selinux/Solaris Trusted Extensions basées sur XACE
- Répare le support multi-cartes
- ...



Et après...

- continuer à nettoyer le code existant,
- support matériel (radeon, nouveau,...)
- implémenter Shatter,
- Xrandr : transformations géométriques 3D sur l'entrée,
- gestion des profils colorimétriques sérieuse,
- Supprimer complètement le besoin des privilèges root,
- support GPGPU
- ...



La Fondation X.Org

Fondation à but non lucratif de l'état du Delaware (USA).

Buts : supporter et favoriser le développement du système X windows et des standards associés.

Membres : individus intéressés par le système X. Adhésion gratuite.

Conseil d'administration

Eric Anholt

Matthieu Herrb

Daniel Stone

Donnie Berkholz

Adam Jackson

Carl Worth

Alan Coopersmith

Bart Massey



Projets de la fondation

- Refonte de la structure légale - presque fini
- Google Summer of Code - vacation of code
- Conférences :
 - XDC - Portland State University, septembre 2009
 - XDS - autour de FOSDEM 2010
- Recruter plus de membres parmi les contributeurs réguliers
- Améliorer la communication



Quelques Difficultés

- Support des fabricants
 - en particulier nVidia...
- Multi-plateformes... tendance à devenir Linux-centrique
- Tiraillements entre distributeurs Linux (ex : radeonhd)
- Pas assez de contributeurs réguliers
- Finances de la fondation en baisse (crise ?)



Listes de diffusion :

- xorg@lists.freedesktop.org liste générale de discussion autour de X.
- xorg-announce@lists.freedesktop.org annonces de nouvelles versions de modules.

Sites web :

- <http://wiki.x.org/> site principal de X.Org
- <http://planet.x.org/> agrégateur de blogs de développeurs X.Org
- <http://phoronix.com/> site d'infos qui suit X.Org de près (M. Larabel)



Participer

- Pages Developpers du wiki :
<http://wiki.x.org/wiki/DeveloperStart>
- Accès git :
 - git clone
`git://anongit.freedesktop.org/git/xorg/...`
 - <http://cgit.freedesktop.org/>
- liste de diffusion : `xorg-devel@lists.x.org` dédiée au développement : soumission de patches, discussion sur ceux-ci.
- bugzilla : <https://bugs.freedesktop.org/> produit **xorg**.
- IRC : canaux `\#xorg` et `\#xorg-devel` sur `freenode.net`
- site des membres de la fondation
<http://members.x.org/>



Questions ?