

---

# Systemes d'Exploitation

## Utilisation « avancée »

Cours de Licence Informatique  
Olivier Dalle

## Plan du Cours

---

### Introduction

- I. Interface Graphique (Xwindow)
- II. Interface Textuelle (zsh)
- III. a - Environnement Réseau
- III. b - Introduction à Windows NT
- IV. Outils de traitement de données
- V. Installation de Linux

## Introduction

- ◆ Objectifs du Cours
- ◆ Organisation Pratique
- ◆ Programme des TPs

- I. Interface Graphique (Xwindows)
- II. Interface Textuelle (zsh)
- III. a – Environnement Réseau
- III. b - Introduction à Windows NT
- IV. Outils de traitement de données
- V. Installation de Linux

### ● Philosophie

#### ◆ Approfondir ses connaissances

- Concepts des SE multi-utilisateurs et multi-processus
- Application à UNIX et Windows NT
- Approche synthétique : entités en présence, organisation, etc.

#### ◆ Orienté vers **l'utilisation** des systèmes

### ◆ Concrètement

- ◆ Comprendre ce qui se passe pour ne pas être perdu devant le système
  - Interfaces de commande : utilisation et configuration
  - Outils systèmes : manipulation de ressources, automatisation de tâches (filtres, scripts, etc)
- ◆ Ce cours ne couvre pas ces aspects :
  - Programmation système
  - Administration système

## Organisation Pratique

---

- 5x2h de Cours
- 6x2h de TD sur machine
- Contrôle des connaissances
  - ◆ CC : 30% (interrogations)
  - ◆ Examen (2h)

## Programme des TD sur machine

---

1. Intro wiki + Révisions, Rappels MI2  
<http://deptinfo.unice.fr/> -> Wiki
2. XWindow
  - utilisation
  - configuration
3. Shell ZSH
4. Environnement Réseau & Intro NT
5. Outils de filtres, expressions régulières
6. Installation de Linux

## I. Interface Graphique (XWindow)

---

### I. Interface Graphique (Xwindow)

1. Présentation générale
2. Architecture de XWindow
3. Fonctionnement du serveur
4. Configuration des clients

### II. Interface Textuelle (zsh)

### III. a - Environnement Réseau

### III. b - Introduction à Windows NT

### IV. Outils de traitement de données

### V. Installation Linux



# I. Interface Graphique (XWindow)

## 1. Présentation Générale

---

### • Objectifs du système XWindow

- ◆ Mécanisme pour construire des Interfaces Homme/Machine
- Aspect Communication et Graphisme

### • Sources gratuits (mais pas GPL)

- ◆ Version actuelle : version 11, release 6 (X11R6)
  - Dernière version : X11R6.6 (25/04/01)
- ◆ Nombreuses « contribs » (clients)
- ◆ Nombreuses adaptations
  - Gratuites : X386, XFree86 (version améliorée de X386)
  - Commerciales : Metro-Link (intel/multi-OS), SCO (ODT), OpenWindows (Solaris)

### Historique Rapide

- Développé au MIT dès 1984 (projet Athena)
- Création du *X-Consortium* en 1988
  - ◆ En association avec des industriels
  - ◆ Garantie de la pérennité

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

<http://www.x11.org>
- Rapproché d'OSF en 1997 : « The Open Group »

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

# Le point sur XFree86 (le serveur X11 gratuit de Linux)

---

- Version améliorée de X386
  - ◆ X386 = portage de X11R6 pour intel/Multi-OS
- Nouvelles fonctionnalités
- Améliorations de performances
- Récemment : adaptation à d'autres architectures (MIPS, Alpha, ...)
- Dernière version (Sept 2002) : XFree 4.2.1 (basée sur X11R6.6)
- <http://www.xfree86.org/>

## I.1 Présentation Générale

# Applications X particulières : L'interface Utilisateur / Système

---

- Par défaut X11 ne fait rien pour faciliter l'utilisation de l'environnement
  - ◆ Fenêtres « nues », sans menu, ni « poignées » de déplacement, ...
- Solution : confier l'habillage à un ou plusieurs clients X (« Window Manager »)
  - ◆ Environnements simples :
    - xterm + window manager + quelques utilitaires choisis
    - Exemples : twm, mwm, tvwm, **fvwm**, ...
  - ◆ Environnements de type « bureau » (cf Windows)
    - window manager + utilitaires spécifiques
    - Exemples : **KDE**, **GNOME**, CDE, eXode
- Avantage : liberté de choisir son « look & feel »

### Quelques pages de Manuel

---

- X : les principes de base
- Xserver, xinit, xdm, XFree86 (linux) : lancement du serveur
- Xsecurity, xauth, xhost : sécurité
- xset, xrdb, editres, ... : config

# I. Interface Graphique (XWindow)

## 2. Architecture du Système XWindow

---

- Basé sur un modèle N clients / 1 serveur
  - ◆ Les clients peuvent s'exécuter à distance, sur d'autres machines
- Le serveur X est chargé de s'occuper du matériel
  - ◆ Ecran, clavier, souris, tablette, ...
  - ◆ S'exécute forcément en local
    - Sur le Terminal X
    - Sur la station « propriétaire » du matériel
- Le serveur X communique avec les clients
  - ◆ Exécute les instructions graphiques reçues
  - ◆ Transmet des informations d'état (interactions souris, clavier, ...)

### Tâches du Serveur X

#### ● Permettre l'accès à(aux) l'écran(s) par les clients X

##### ◆ Désignation d'un écran (« display »)

`machine_serveurX:num_console.num_écran`

##### ◆ Exemple :

- Serveur sur machine  $\neq$  machine client

`export DISPLAY=toto.unice.fr:0.0`

- Clients et serveur sur la même machine

`export DISPLAY=:0.0`

- Xon : Configuration automatique pour un client donné :

`xon machine.unice.fr client`

*Attention : xon ne marche pas toujours ...*

### Tâches du Serveur X

---

- Interpréter et exécuter les messages (requêtes) envoyés par les clients X
- Transmettre aux clients X les évènements liés au clavier et à la souris
  - ◆ Pour se rendre compte, essayer la commande `xev`
- Les échanges se font selon le *protocole X11* (format des messages : type, données, ...)
  - ◆ Protocole complètement portable
    - Exemple : au dessus de TCP/IP

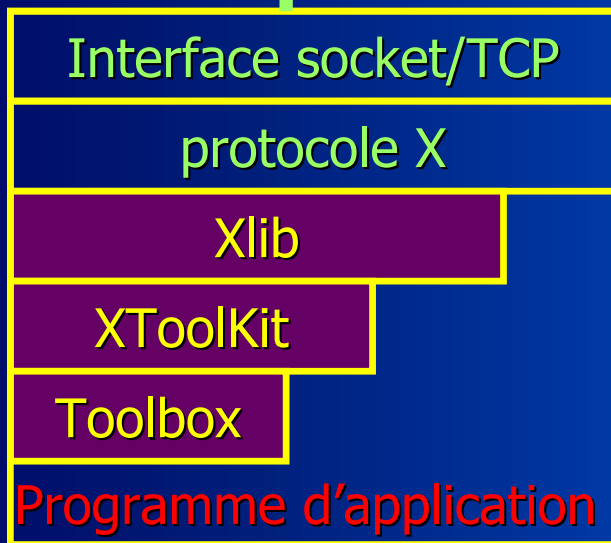


## I.2 Architecture du Système XWindow

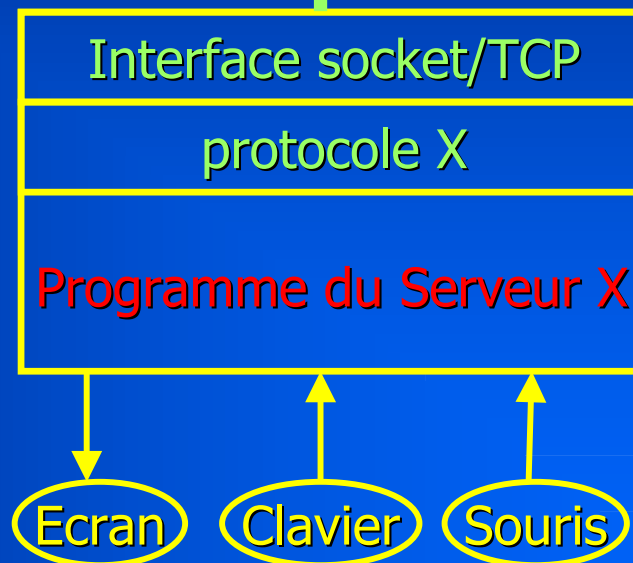
# Architecture d'un Serveur X11

Réseau

Machine du client



Machine du serveur



# Architecture d'un Client X

- Programme utilisant données et fonctions de l'API
  - ◆ La Xlib
  - ◆ La XToolkit (intrinsics)
  - ◆ Une Toolbox = Boîte à outils graphique de haut niveau proposant un « Look & Feel »
    - Motif, OpenLook, Athena, ...
- Interface d'un Client constituée d'une hiérarchie de « widgets » (Windows Gadgets)
  - ◆ Un widget fenêtre principale peut contenir d'autres widget fenêtres, qui peuvent contenir des ascenceurs, des listes, des boutons, des bitmaps, etc.

### Ressources d'un Client X

---

- ◆ Certains paramètres de l'application ou des widgets peuvent être personnalisés
  - ◆ Apparence, géométrie, couleurs, texte, comportement, etc.
  - ◆ Un tel paramètre est appelé « **ressource** »
    - Certaines ressources sont **standard** = communes à toutes les applications
    - D'autres sont **spécifiques** à l'application
  - ◆ Les valeurs des ressources sont obtenues au démarrage du client

# Configuration des Ressources

- Les valeurs des ressources sont obtenues à partir de différents endroits
  - ◆ Gestionnaire de Ressources de X
  - ◆ Ligne de commande
  - ◆ Fichier(s) de configuration commun
  - ◆ Fichier(s) de configuration de l'utilisateur
- La recherche de la valeur définitive se fait selon **un ordre précis** (et complexe) de consultation de ces endroits
  - ◆ Une même ressource peut changer plusieurs fois de valeur avant de prendre sa valeur définitive !
  - ◆ Règle générale : l'utilisateur a toujours le dernier mot ...

### ● Xlib

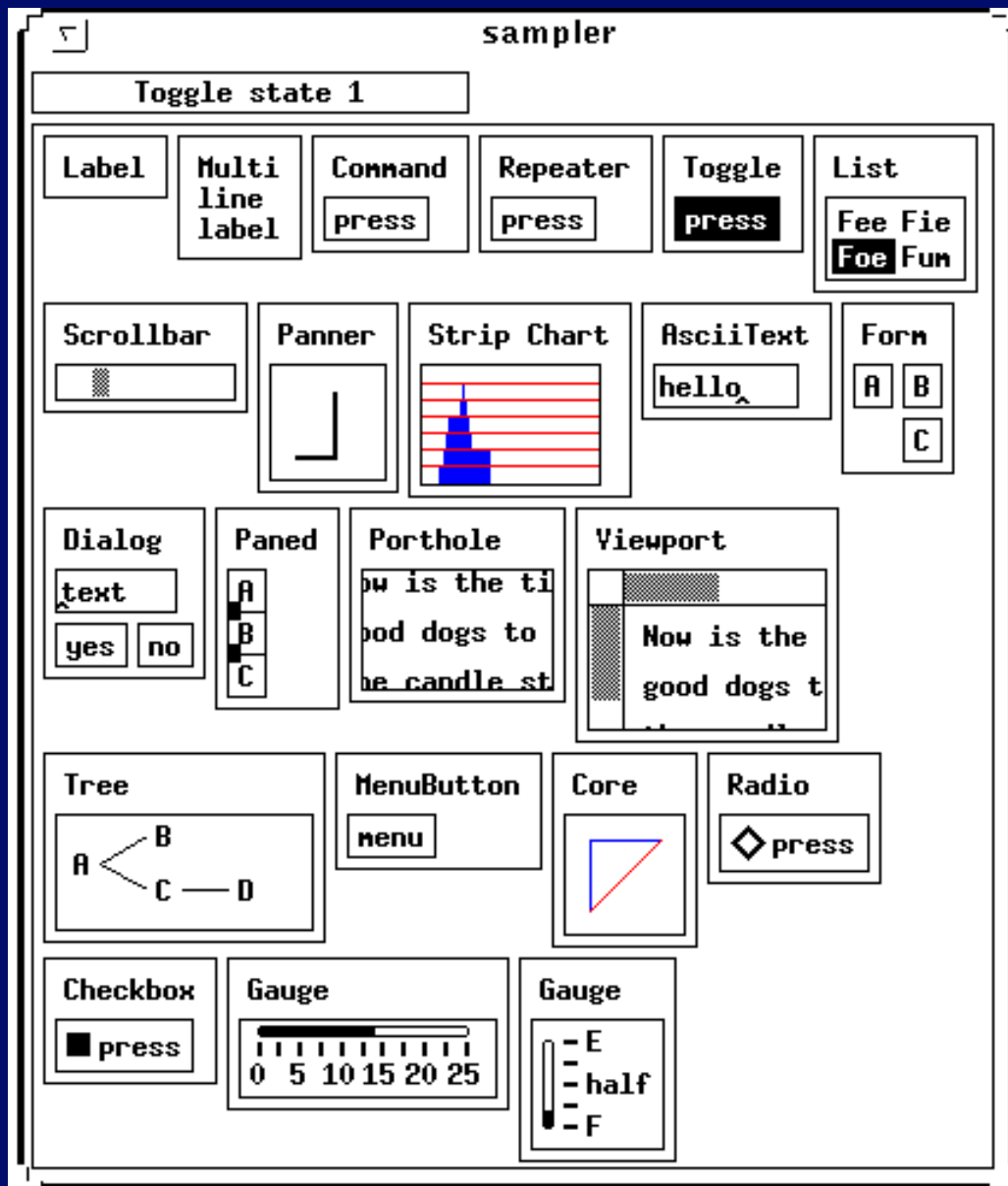
- ◆ Une interface en « C » pour programmer avec le protocole X (libX11.a)
  - XChangeKeyboardMapping
  - XDrawText, XDrawPoint, XtDrawRectangle ...

### ● XToolkit Intrinsic (libXt.a)

- ◆ Définit une hiérarchie abstraite de Widgets et les ressources standard
  - XtResizeWidget
  - XtMenuPopup, XtMenuPopDown
  - XtResolvePathName

### ● Toolbox : implémentation d'un modèle d'assemblage de widgets spécialisés (look & feel)

# Exemples de widget Athena

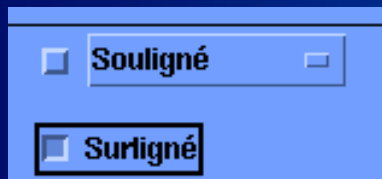


## I.2 Architecture du Système XWindow

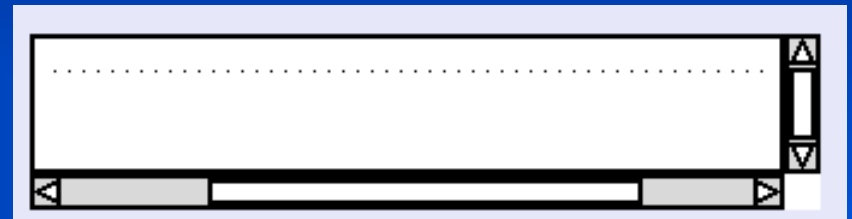
# Exemples de Widgets Motif

---

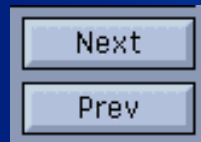
### Cases à cocher, liste



### Ascenceurs



### Boutons



## 3. Fonctionnement et Configuration du Serveur X

---

### ● Lancement d'une session X

- ◆ Avec démarrage « manuel » du serveur
  - scripts startx ou xinit
- ◆ A partir d'un serveur déjà lancé
  - Bannière de login graphique
    - Xdm
    - Gdm (GNOME)
    - Kdm (KDE)



# Démarrage manuel du Serveur

---

### • Lancement par xinit

```
xinit [clientargs [-- serverargs]]
```

#### ◆ Exemples :

- Désignation explicite du premier client :

```
xinit xterm
```

```
xinit emacs -geometry +1+1
```

```
xinit /usr/local
```

- Désignation implicite du premier client

```
xinit ⇔ xinit $HOME/.xinitrc
```

### • startx : script qui appelle xinit

# Lancement de la Session par Xdm

- Xdm est lancé au boot de la machine
  - ◆ Peut proposer une liste de machines (Terminal X)
  - ◆ Propose une bannière de login graphique
  - ◆ Evolutions plus conviviales de Xdm : Gdm, Kdm, ...
    - Choix et configuration de l'environnement de travail
- Déroulement de la connexion
  1. Interprète script Xstartup (compte root)
  2. Lance un shell pour le compte de l'utilisateur
    - ◆ Interprète le script Xsession
    - ◆ Xsession invoque `$HOME/.xsession`
    - ◆ Fin de XSession = fin de session, déconnexion, retour de la bannière

# Protection et Autorisations d'Accès

---

- L'accès au serveur se fait « à la tête du client »
  - ◆ Protection indispensable :
    - Un client espion peut récupérer des mots de passe lors de leur saisie, copier et transmettre le contenu des fenêtres, etc.
- 2 types de protection :
  - ◆ Selon la machine hôte du client
    - Seules certaines machines sont autorisées
  - ◆ Selon un mécanisme de clef secrète
    - Le client doit présenter la clef secrète pour prouver qu'il a bien la permission

## I.3 Fonctionnement et Configuration du Serveur X

# Protection selon la machine hôte du client

- Le serveur maintient une liste de machines autorisées
- L'utilisateur peut modifier la liste des machines autorisées à connecter des clients : commande `xhost`
  - ◆ `xhost` : affiche la liste
  - ◆ `xhost -` : aucune machine autorisée (sauf locale)
  - ◆ `xhost +` : toutes les machines sont autorisées (danger !)
  - ◆ `xhost +gelas.unice.fr` : ajoute la machine gelas à la liste
  - ◆ `xhost -gelas.unice.fr` : retire la machine gelas de la liste

## I.3 Fonctionnement et Configuration du Serveur X

# Protection par clef secrète (MIT-MAGIC-COOKIE-1)

---

### ● Idée :

- ◆ Le serveur connaît un secret (clef = code)
- ◆ Pour se connecter un client doit prouver qu'il connaît aussi le secret

### ● Comment le client connaît-il le secret du serveur ?

- ◆ Le secret est stocké au démarrage de la session dans un fichier **protégé en lecture** (`.xauthority`) sur le compte de l'utilisateur « propriétaire » du serveur
- ◆ Les clients lancés par l'utilisateur « propriétaire » ont le droit de lire le fichier, mais pas les clients lancés par d'autres utilisateurs

## I.3 Fonctionnement et Configuration du Serveur X

# Protection par clef secrète (MIT-MAGIC-COOKIE-1)

- Pour permettre à un autre utilisateur d'utiliser le serveur X, il suffit que le « propriétaire » lui donne sa clef

- ◆ par mail, téléphone, rsh, etc

- Comment extraire la clef du fichier `.Xauthority` ?

- ◆ Commande `xauth` :

- `xauth list` : donne la liste des clefs

- `xauth extract - display` : extrait une clef

- `xauth add/merge` : ajoute une clef

- Exemple :

- ```
xauth extract - $DISPLAY | \  
rsh station -l login xauth merge -
```

- ◆ Remarques

- Il arrive qu'on ait besoin de se donner la clef secrète à soi-même ...
  - L'autorisation par **xhost est prioritaire** sur l'autorisation par clef secrète.

## I.3 Fonctionnement et Configuration du Serveur X

# Configuration du Serveur X

### ● Information sur la gestion du matériel :

- ◆ `xset q`

### ● Clavier :

- ◆ `xset b`, `xset c`, `xset r`,

- ◆ `xkeycaps`, `xmodmap`

### ● Souris : `xset m`

### ● Ecran :

- ◆ `xset s`

- ◆ fenêtre de fond : `xsetroot`, `xpmroot`

### ● Polices de caractères :

- ◆ `xset fp`

- ◆ `xlsfont`, `xfontsel`

# Gestion des couleurs

### 2 façons de désigner les couleurs

#### ◆ nom logique :

- red, green, blue, yellow, darkseagreen, ...
- /usr/lib/X11/rgb.txt

#### ◆ Combinaison de 3 couleurs primaires : Rouge, Vert, Bleu (RGB)

- X reconnaît des niveaux d'intensité pour chaque couleur primaire
- un niveau d'intensité est donné par un nombre hexa à 1, 2 3 ou 4 digits (0= mini, f, ff, fff ou ffff = maxi)
- syntaxe : rgb:rr/gg/bb
- exemple :

```
xclock -bgcolor `rgb:ff/00/00`
```



## 4. Configuration des Clients X

---

- Fenêtre d’Affichage
- Ressources
  - ◆ Définition
  - ◆ Nommage
  - ◆ Détermination des Valeurs

# Fenêtre d’Affichage

- L’interface d’un client apparaît dans une fenêtre (généralement rectangulaire)
- Le Window Manager
  - ◆ Ajoute une décoration : bords, titre, poignées, menus, boutons
  - ◆ Se charge des modifications
    - Déplacement, taille, aspect, ...
  - ◆ La décoration est (en général) personnalisable
    - Fichier `$HOME/.wmrc`  
`.twmrc`, `.fvwmrc`, `.tvtwmrc`, ...
  - ◆ Remarque : tuer le WM ne tue pas le client (mais peut terminer la session)

# Géométrie d'une Fenêtre

● **[L x H] [ {+/-} X {+/-} Y ]**

◆ L,H : en pixels ou en caractères, selon client

◆ +/- X : gauche/droite

◆ +/- Y : haut/bas

● **Exemples :**

◆ 80x25

◆ 80x25+10-20

◆ +500+1

◆ -1-1

# Ressources des Clients

---

- Ressource = paramètre de l'application ou d'un widget de sa hiérarchie
  - ◆ peut modifier apparence
  - ◆ peut modifier comportement
- Valeur par défaut peut être modifiée
  - ◆ (man client !)
  - ◆ Modification en ligne de commande
  - ◆ Modification dans un fichier (lancement plus simple)

# Types de Ressources

---

### • Ressources Standard

- ◆ Tout client développé au dessus de la Xlib les possède
  - géométrie
  - background (bg), foreground (fg)
  - borderwidth (bw)
  - title ...

### • Ressources spécifiques

- ◆ Propres à l'application
  - Pas toujours modifiables en ligne de commande

### Nommage des Ressources

---

- Un widget (l'appli aussi) possède des champs élémentaires : ressources
- Un widget est une occurrence (instance) d'un type (une classe)
  - ◆ « Une sorte de »
- Chaque ressource peut être désignée
  - ◆ Par son nom (nom d'instance)
  - ◆ Par son type (nom de classe)

# Convention de Nommage des Ressources

---

### • Convention :

- ◆ nominstance
- ◆ NomClasse

### • Exemples

- ◆ scrollbar/ScrollBar
- ◆ appli/Appli
  - emacs / Emacs
- ◆ xclient /XClient
  - xterm/XTerm

• Rem : Le nom d'instance peut être redéfini par l'utilisateur, mais pas la classe

# Schéma de Nommage Hiérarchique

- Les widgets sont organisés hiérarchiquement
  - ◆ Hiérarchie = arbre
  - ◆ Désignation de la ressource = chemin dans l'arbre

:

```
[[racine]{ .widget }].ressource
```

Avec :

- racine = classe / instance
- widget = classe / instance
- ◆ Remarque :
  - Si racine est omis : tous les clients sont désignés



# Exemple de Désignation de Ressources

### Chemins d'instance

- ◆ `xc.calculator.pave_numérique.zero.labelString : 0`
- ◆ `xc.calculator.pave_numerique.un.labelString : 1`

### Chemins de classe

- ◆ `XC.XmForm.XmForm.XmPushButton.XmString : 0`
- ◆ `XC.XmForm.XmForm.XmPushButton.XmString : 1`

**=> Bouton « 0 » étiqueté « 1 » !!**

### Chemin mixte

- ◆ `XC.XmForm.XmForm.zero.labelString : 0`
- ◆ `XC.XmForm.XmForm.un.XmString : 1`

## I.4 Configuration des Clients X

### Utilisation de Raccourcis dans les chemins

---

- \* peut remplacer tout ou partie d'un chemin
- ? peut remplacer un élément du chemin
- Exemples :
  - ◆ `XC.*.zero.labelString : 0`
  - ◆ `XC.?.?.un.XmString : 1`
  - ◆ `*.deux.XmString : 2`

## I.4 Configuration des Clients X

# Résolution des Ambiguïtés de Désignation

- Pour chaque composant de chemin en partant de la racine : priorité au plus spécifique

1. \*
2. ?
3. classe
4. instance

- Exemple :

```
*.labelString : 5  
XC.*.un.labelString : 3  
xc.*.labelString : 1
```

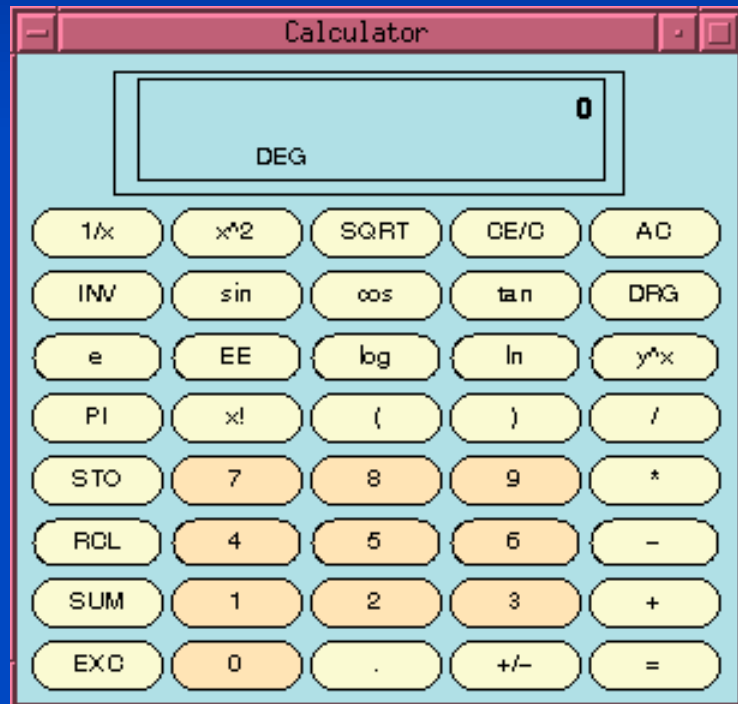
Tous les boutons contiennent « 1 »  
car xc plus spécifique que XC

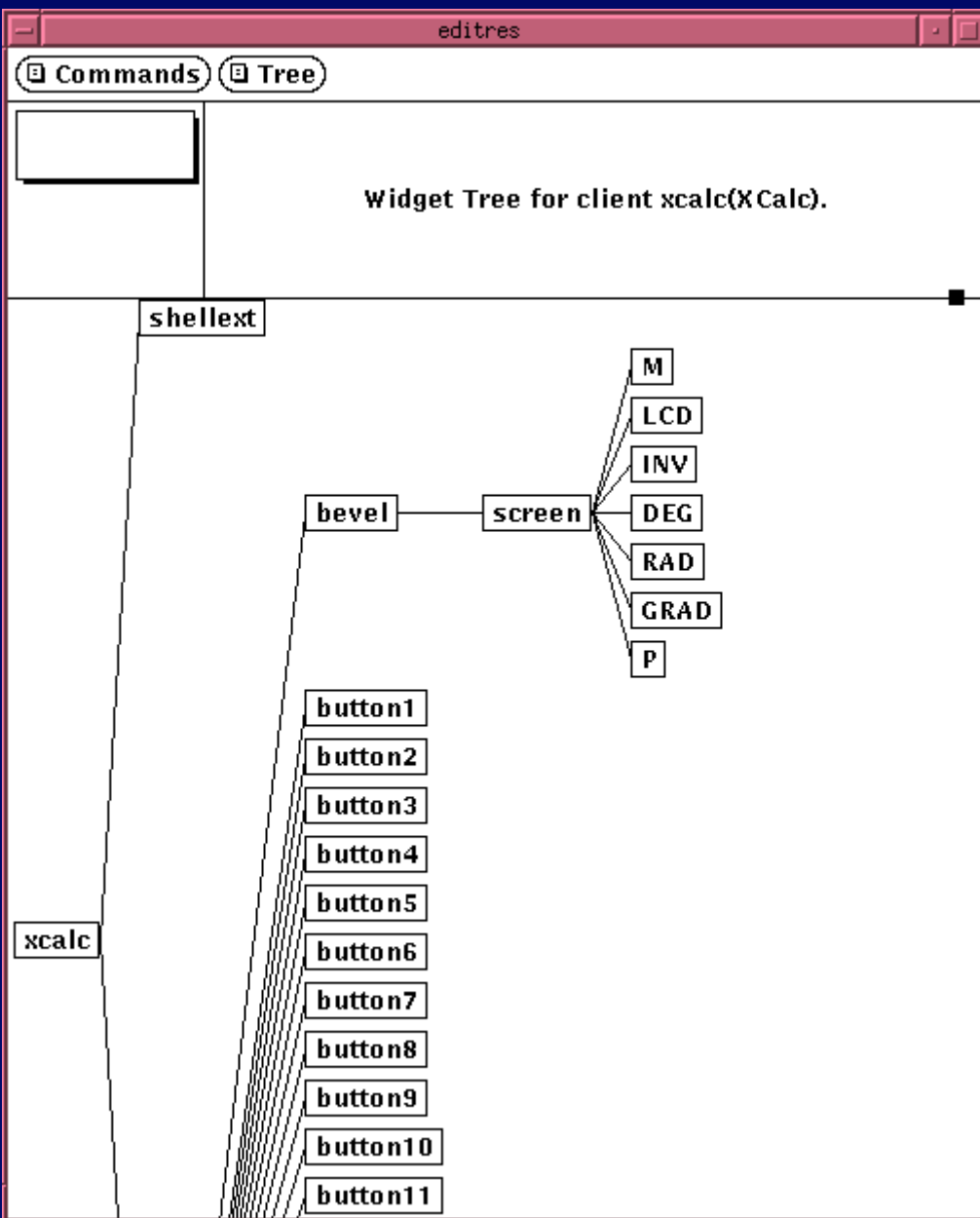
- En cas d'égalité (multiples définitions) : la dernière l'emporte

## I.4 Configuration des Clients X

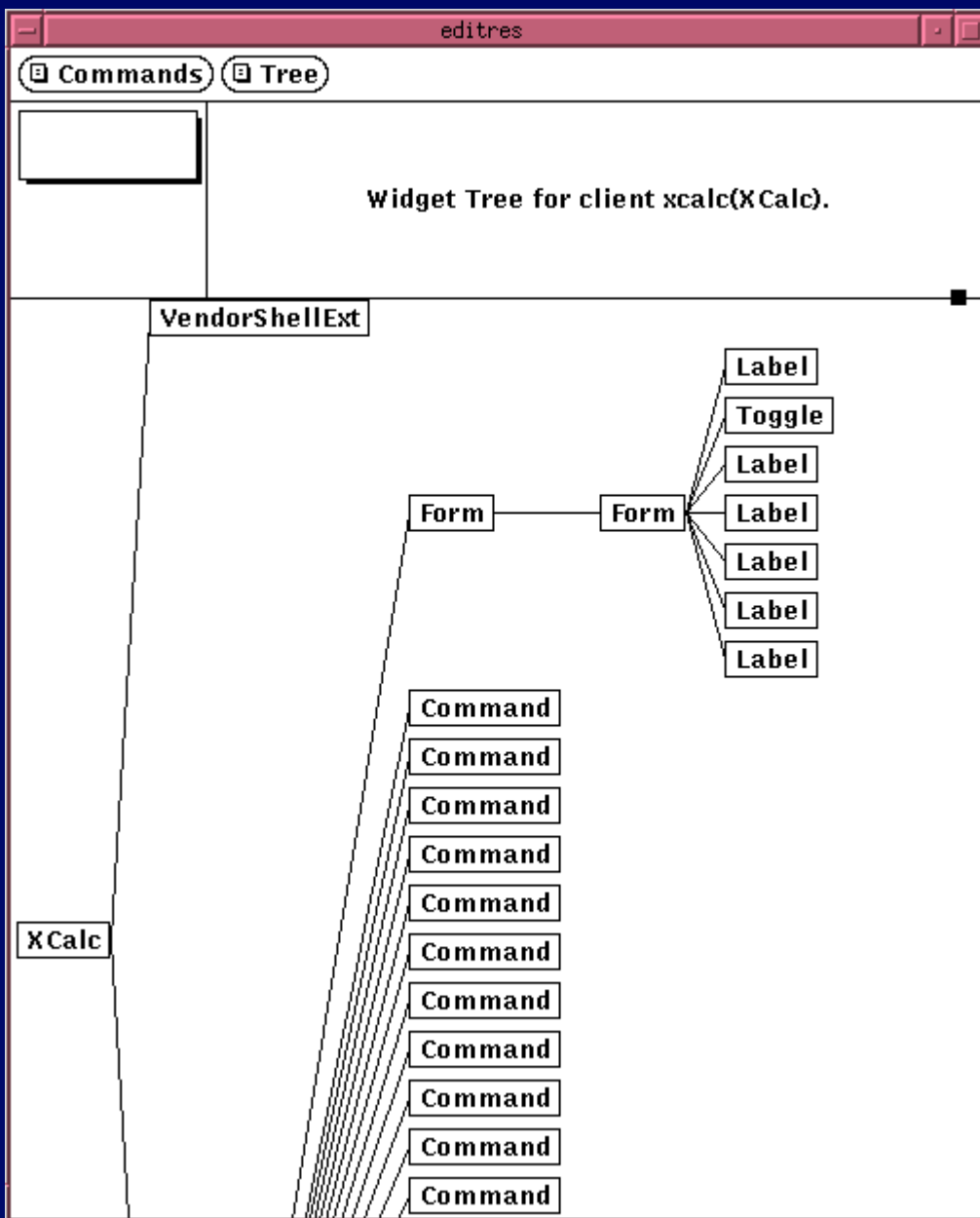
# Outils pour Consulter les Définitions de Ressources

- `viewres` pour la toolbox Athena
- `editres` (lorsque l'appli est compilée pour)
- Exemple sur `xcalc`





# Hiérarchie des Widgets de xcalc



# Classes des Widgets de xcalc

## .xcalc.ti.button1.background:

| .xcalc           | .ti              | .button1         |
|------------------|------------------|------------------|
| * XCalc *        | * Form *         | * Command *      |
| Any Widget       | Any Widget       | Any Widget       |
| Any Widget Chain | Any Widget Chain | Any Widget Chain |

### Normal Resources: mb2 gets a value

|                    |                        |
|--------------------|------------------------|
| accelerators       | highlightThickness     |
| ancestorSensitive  | insensitiveBorder      |
| <b>background</b>  | internalHeight         |
| backgroundPixmap   | internalWidth          |
| bitmap             | international          |
| borderColor        | justify                |
| borderPixmap       | label                  |
| borderWidth        | leftBitmap             |
| callback           | mappedWhenManaged      |
| colormap           | pointerColor           |
| cornerRoundPercent | pointerColorBackground |
| cursor             | resize                 |
| cursorName         | screen                 |
| depth              | sensitive              |
| destroyCallback    | shapeStyle             |
| encoding           | translations           |
| font               | width                  |
| fontSet            | x                      |
| foreground         | y                      |
| height             |                        |

### Constraint Resources

|           |               |              |
|-----------|---------------|--------------|
| bottom    | horizDistance | right        |
| fromHoriz | left          | top          |
| fromVert  | resizable     | vertDistance |

red

Enter Resource Value:

Set Save File Save Apply Save and Apply

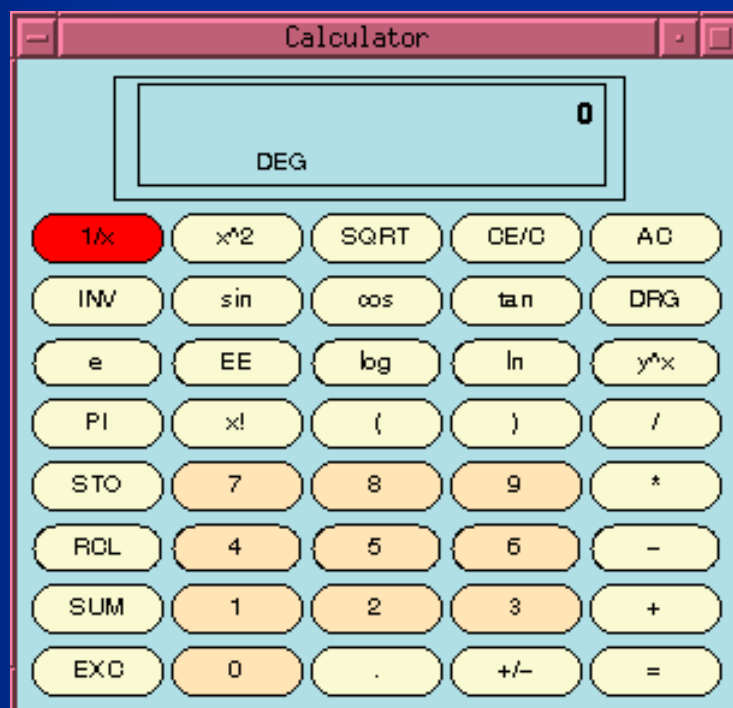
Popdown Resource Box

# Ressources d'un Widget

Possibilité de modifier dynamiquement une ressource avec editres

## I.4 Configuration des Clients X

# Résultat de la Modification





# Où sont Définies les valeurs des Ressources ?

---

- Gestionnaire des Ressources de X (Xrm):
  - ◆ Décide au moment du lancement du client
  - ◆ Décide quelle valeur affecter à chaque ressource
    - Plusieurs sources existent
    - Consultation dans un ordre précis, avec écrasement : en cas d'égalité, c'est la dernière spécification qui est gardée

## I.4 Configuration des Clients X

# Sources de définition des Ressources

1. Valeurs par défaut pour tout le site
  - `/usr/lib/X11/app-defaults/ClassAppli`
2. Valeurs par défaut pour l'application par l'utilisateur
  - `$XAPPLRESDIR/ClassAppli`  
`$HOME/app-defaults/ClassAppli`
  - ou `$HOME/ClassAppli`
3. Valeur par défaut par utilisateur
  - Chargées par `xrdb (.resources)`
  - `$HOME/.Xdefaults`
4. Valeur par défaut par utilisateur pour la machine où le client est lancé
  - `$HOME/.Xdefaults-nom-machine`
5. Paramètres (option ou `-xrm`) de la ligne de commande