UNIX System Programming
Lecture 23: X Windows

- Outline
  - Introduction
  - Window Managers
  - Remote X
- Reference
  - BLP: Chapter 16
Lecture 23: X Windows

Introduction

- The X Window System (aka X or X11) was released by MIT in 1984. The X.Org Foundation currently leads X development. X is available as free software.

- X has features that are still not found on other display systems.

- X is not an integral part of the OS. It is an application layer on top of the kernel.
Lecture 23: X Windows
Introduction

- X was designed to be used over a network.
- On most windowing systems, the application draws directly on the screen. In X, client and server processes are used to draw images. The server takes care of drawing and managing screen contents. The client (application) sends graphics requests (draw window here, etc.) to the X server.
Lecture 23: X Windows

Introduction

- The client (application) and (display) server communicate over a network and may be running on different machines. You have the ability to run programs on one machine while displaying graphics on a different one.

- The client and server processes can even be running on different OSes. X servers (and clients) are available for UNIX, Windows and the Mac.
Lecture 23: X Windows

Introduction

- The *X server* is responsible for displaying graphics. A server is usually running on the user's machine. (This is not a requirement, but displaying graphics on a remote machine that you can’t see is not that useful.)

- The X server accepts requests for graphical output from clients and sends back user input (from keyboard, mouse or touchscreen) to clients.
Lecture 23: X Windows

Introduction

- Window, Session Managers
- High Level X Toolkit
- Low level API (Xlib)
- X Network Protocol
- Base System (Server)
- Application
The base X system provides *mechanism, not policy*. The base system can draw overlapped, tiled, or pop-up windows. Whether and how these should be displayed is human interface policy and is left up to the window manager. The network protocol is the *only* interface to the base system.
Programmers typically do not program directly in Xlib. Instead a higher order toolkit is used. There are many toolkits available. Some popular ones are:

<table>
<thead>
<tr>
<th>Toolkit</th>
<th>Language</th>
<th>Shipped With</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tk</td>
<td>Tcl</td>
<td>Tcl, Python</td>
</tr>
<tr>
<td>GTK+</td>
<td>C</td>
<td>Gnome</td>
</tr>
<tr>
<td>Qt</td>
<td>C++</td>
<td>KDE</td>
</tr>
<tr>
<td>FLTK</td>
<td>C++</td>
<td>—</td>
</tr>
<tr>
<td>wxWidgets</td>
<td>C++</td>
<td>—</td>
</tr>
</tbody>
</table>
Lecture 23: X Windows Toolkits

- Bindings for C, C++, Perl, Tcl, and Python are available for all of the toolkits. They are also available for Windows and the Mac.
- Tk is the oldest. It was originally developed for Tcl. It is now included with Python.
- GTK+ was developed to support the GIMP, but it is now the standard GNOME toolkit.
- Qt is the toolkit used by KDE.
- FLTK and wxWidgets were designed to be cross-platform toolkits.
Lecture 23: X Windows
Window Managers

- The window manager is a special client that acts as interface between user and X server.
- The window manager provides window decorations (borders), maintains window focus, window iconification, and menus (that can be used to launch applications).
- There are many, many window managers available: twm, mwm, FVWM, AfterStep, Blackbox, KWin (KDE), Metacity (Gnome) ...
• A window manager is not required, but when one is running, some server – client interaction is redirected through the window manager. In particular, a request for a new window is redirected to the window manager and the manager decides on the initial window position. A window manager will usually re-parent a window by adding a frame and title bar.
Lecture 23: X Windows

No Window Manager
Lecture 23: X Windows
twm Window Manager
Lecture 23: X Windows
FVWM Window Manager
Lecture 23: X Windows

Cygwin/X (win) LXDE Manager
Lecture 23: X Windows
Cygwin/X (full) LXDE Manager
Lecture 23: X Windows

Cygwin/X Integrated Manager
Lecture 23: X Windows

Xming Integrated Manager
Lecture 23: X Windows
Lubuntu (18.04) LXDE Manager
Lecture 23: X Windows
Ubuntu (18.04) Gnome Manager
Lecture 23: X Windows
FreeBSD (11.2) Gnome-Session
Lecture 23: X Windows
Remote Clients

• By default only clients running on the same machine as the server are allowed access to the server. Access to remote clients must be specifically granted.

• X supports several access methods (see the Xsecurity man page). Only host based access control is discussed here.

• On the local (X server) machine:

  xhost +oreo.ibm.com      # allow oreo
  xhost +                  # allow all
Lecture 23: X Windows
Remote Clients

- On the remote machine we must tell the clients where the server is located. This is done either with the "-display" argument to the client or by setting the DISPLAY environment variable:

  xclock -display pc10.ibm.com:0.0

  or

  export DISPLAY pc10.ibm.com:0.0
  xterm
The name of an X server is of the form: hostname:displaynumber.screennumber.

The *hostname* may be either a name or IP address. The *displaynumber* and *screennumber* are usually 0 (unless you are running multiple X servers or screens).

An X server does not need to be run on the machine running the clients. The server and clients may be running on different OSes!
Lecture 23: X Windows
Remote Clients

- Instead of host based access control, use ssh with “-Y” to connect to the remote machine.
- You don't need to use “xhost” on the server machine or set the DISPLAY variable on the remote machine.
- ssh automatically sets the DISPLAY variable to a dummy server on the remote machine and then tunnels the X display traffic back to the local machine.
Lecture 23: X Windows Remote Clients

- The geometry option allows you to specify not only the size but also the position. The format is “XSIZExYSIZE+XOFF+YOFF”:
  
  \[
  \begin{align*}
  &\text{xclock } -\text{geometry } 100x100+50+50 \\
  &\text{xeyes } -\text{geometry } 50x50-0-100
  \end{align*}
  \]

- Here are a few other standard options:
  
  \[
  \begin{align*}
  &\text{xterm } -\text{fg cyan } -\text{bg black} \\
  &\text{xterm } -\text{title } \text{“ssh to cserver”} \\
  &\text{xterm } -\text{iconic}
  \end{align*}
  \]
KDE and GNOME are complete desktop environments. They include standard sets of applications for home, office, education, development and administration.

Each provides configurable “look and feel” (themes), unified access to documentation, drag and drop protocols, IPC protocols (CORBA - GNOME and DCOP-KDE).

You can run KDE applications under GNOME and GNOME apps under KDE.
Lecture 23: X Windows
Introduction to GTK

- GTK+ is a multi-platform toolkit for creating graphical user interfaces. It was initially developed for and used by the gnome (the GNU Image Manipulation Program), but is now used by several applications. It is the toolkit used by the GNOME desktop.

- You can write GTK+ apps under X on Linux, Windows (Cygwin) and Mac. There is also a native Windows version available.
GTK+ consists (primarily) of 3 libraries:

- GLib is the low-level core library that forms the basis of GTK+ and GNOME.
- Pango is a library for layout and rendering of text, with an emphasis on internationalization.
- The ATK library provides a set of interfaces for accessibility. By supporting the ATK interfaces, an application or toolkit can be used with such tools as screen readers, magnifiers, and alternative input devices.
Under X Windows GLib depends upon the GDK (GTK+ Drawing Kit) library which is a wrapper around corresponding Xlib routines.

GTK+ is an object-oriented library written in C. There are bindings for many other languages including: C++, Perl, Python, Guile, Java, and C#. Only the C interface will be discussed.