Unix GUI

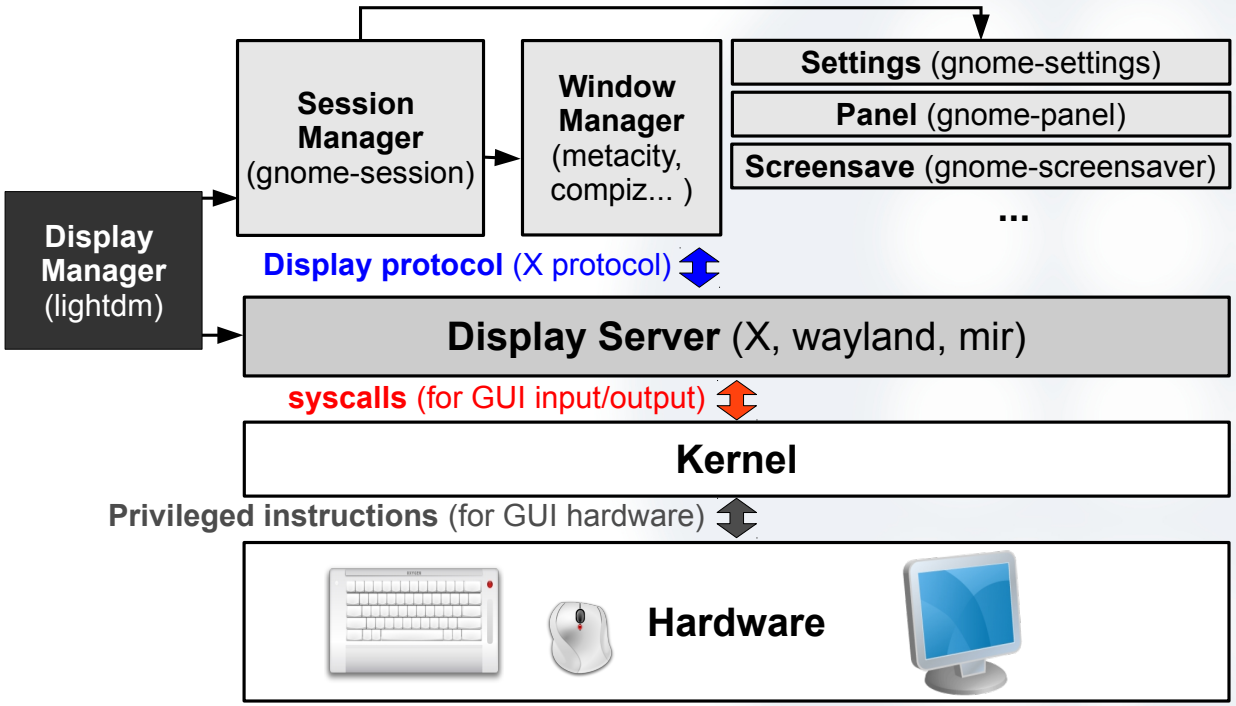
Jose L. Muñoz, Oscar Esparza, Juanjo Alins, Jorge Mata
Telematics Engineering
Universitat Politècnica de Catalunya (UPC)

Outline

Outline

GUI Organization

- The GUI is achieved through the co-operation of separate components.
- Complex but flexible design (user space and client/server).

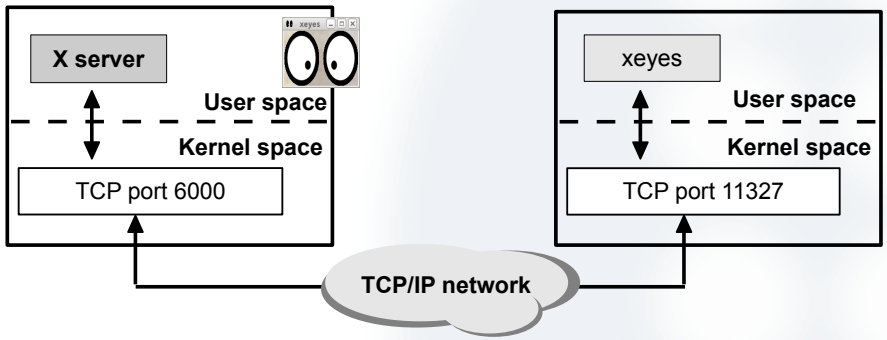


Display Server

- A display server is a program whose primary tasks are:
  - Coordinate the input and output of its clients to and from the hardware related to GUI.
  - Coordinate the communication between its clients.
- Uses GUI syscalls with the kernel and a display protocol with the clients.
- Clients need not be concerned with the details of talking to the actual graphics display device.
- Display protocol: “draw a line from here to here”.
- Not just a graphics library but a communications protocol that can be used over a network.
- This is called a “network transparent protocol”.
- Most widely known display server implementation is the X server.

X's

- The X protocol is network transparent.
- X was designed as a client-server architecture.
- X provides a library called Xlib, which handles all low-level client-server communication tasks.
- The X server listens by default on port TCP 6000.
- The applications themselves are the clients.



- Terms seem reversed, user's computer runs the server and remote applications are the clients.
- But the display server is a "server" because it provides graphic I/O services for its clients.

Window Manager I

- Clients should not be concerned about what other clients have displayed on the screen (manage overlaps).
- This is smartly solved using rectangle areas are called windows.
- Clients use a window and then call the functions to perform the graphic actions inside the window like:
  - “draw a line from here to there”
  - “tell me whether the user is moving the mouse in my screen area”.
- The display server does not do the job of managing windows.

Window Manager II

- This is the job of the “window manager”.
- The window manager decides things like:
  - Where to place windows.
  - Gives mechanisms for users to control the windows’ appearance, position and size.
  - Provides “decorations” like window titles, frames, buttons, etc.
  - Most window managers provide additional facilities like some way of launching applications.
- Examples: metacity, compiz, kwin, icewm, fvwm, etc.

Desktop System I

- Libraries provided by display servers are very low level libraries.
- For example, with Xlib (the X's lib) is terribly complicated doing things like putting buttons on screen, text, or nice controls like scrollbars, menus, etc.
- Luckily, these controls (known as widgets) have been programmed and are accessible through a "widget library".
- As it happens with window managers, there are many widget libraries or toolkits.
- Most widely known being Gtk and Qt.

Desktop System II

- Due to the fact that clients can be programmed using several possible different toolkits and we can use different window managers:
  - Each window manager has a different approach to managing the clients (decorations, etc).
  - The Toolkit selected also impacts in the look and behavior of applications.
  - Using different toolkits also increases resource usage (dynamic shared libraries).
- There are features that one expects from a GUI environment that our current GUI components do not cover.
- For example, a control panel for configuring the system, a screen-saver, etc.

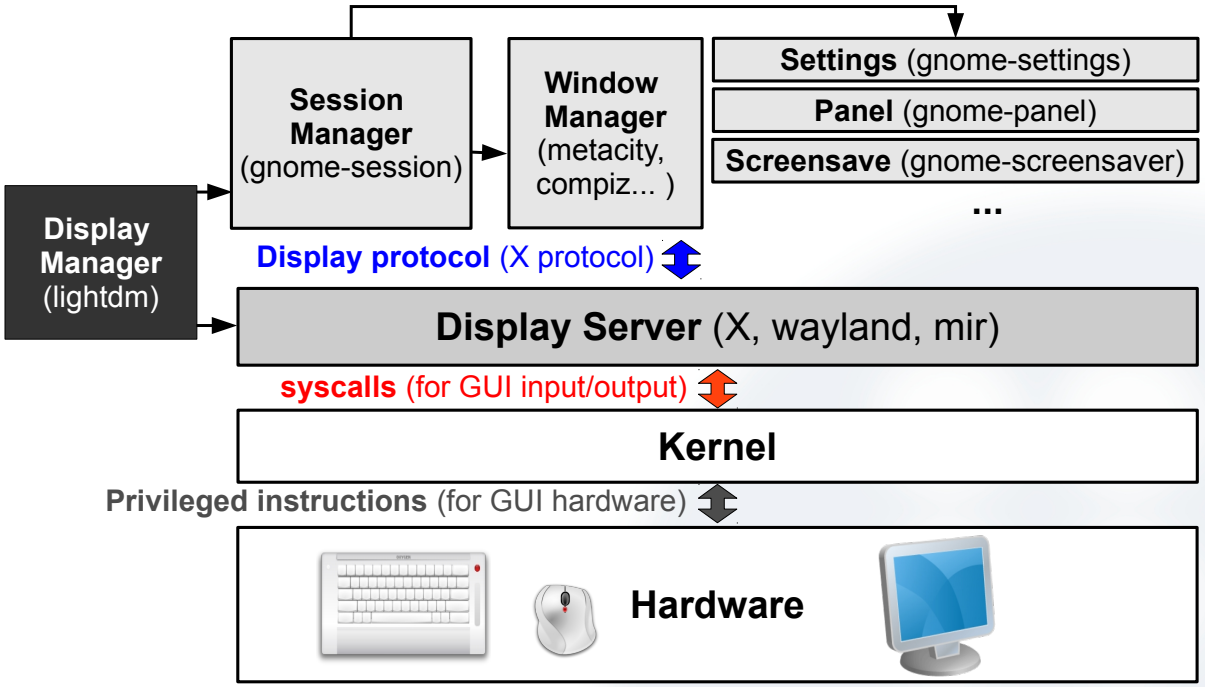
Desktop System III

- A desktop environment aims to provide a complete interface to the operating system, and supply its own range of integrated utilities and applications.
- Under Linux, the two most popular desktop environments are KDE and GNOME:
  - KDE uses the Qt toolkit and kwin as window manager.
  - GNOME
    - Has always tried to be window manager-agnostic.
    - Today it uses mainly compiz and metacity.
    - It uses the Gtk toolkit.
    - Provides a set of higher-level functions and has its own set of programming guidelines in order to guarantee a consistent behavior between compliant applications.

Session Manager

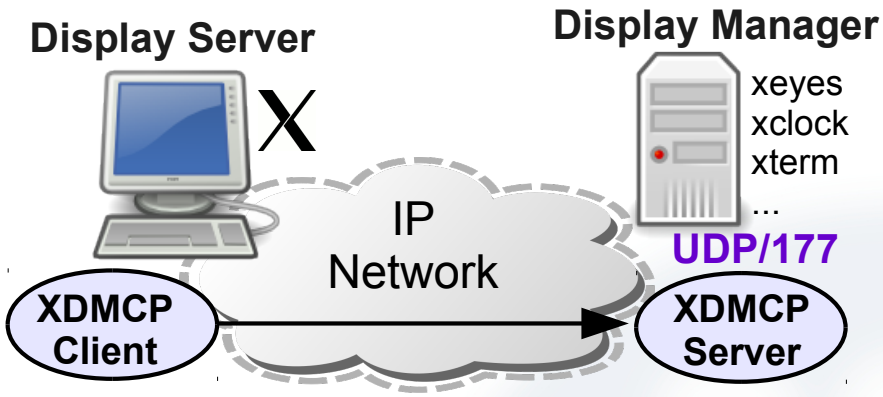
- A session is the collection of applications, settings, and resources (like open files) present on the user's desktop.
- A session manager is responsible for:
  - Starting sessions on a desktop environment.
  - Providing users a possibility to save and restore their sessions.
- The X Window System includes a default session manager called xsm.
- Other session managers have been developed for specific desktop systems: ksmserver (KDE) and gnome-session (GNOME).

Display Manager I



- The last important component is the display or login manager.
- The display manager starts the display server before presenting the user the login screen.
- The login screen prompts for a username and password.

Display Manager II



- The display manager and the display server can also **run on different systems**:
  - The display server runs on the user's computer.
  - The display server is configured to be connected to a remote display manager.
  - The X Display Manager Control Protocol (XDMCP) is used.
  - XDMCP uses the client/server model with the well-known port 177/UDP for the server.

Display Manager III

- The **display server is the XDMCP client**.
- The **display manager is the XDMCP server**.
- The display server requests the display manager to start a session in the remote desktop.
- After that, we can execute GUI applications in this session.

- There are many display managers: XDM (X standard), GDM (GNOME), KDM (KDE) and LightDM (by Canonical Ltd, the creators of Ubuntu).
- On many Linux distributions, the display manager is selected in the file /etc/X11/default-display-manager.

Outline

Transport Architecture I

- X communications can use different transport architectures.
- Main are Unix sockets and TCP/IP sockets.
- Transport to be used is determined by a ”`displayname`“.
- `displaynames` are in the form: `hostname:displaynumber.screennumber`.
  - `hostname` is the name or IP address of the host in which is running the X server.
  - `displaynumber` identifies the X server within the host since there can be different X servers running at a time in the same host.
  - `screennumber` specifies the screen to be used on that server. Multiple screens can be controlled by a single X server.

Transport Architecture II

- When `displayname` does not contain a hostname, e.g. ":0.0", then Unix sockets are used.
- When `displayname` does contain a hostname, e.g. "localhost:0.0", then TCP/IP sockets are used.
- The `displayname` can be found in the DISPLAY environment variable.
- Example:

```
$ export DISPLAY="192.168.0.1:1.0"
$ xeyes &
```

- In the previous example, we are specifying with DISPLAY the address of the X server in which the `xeyes` are going to be displayed.

Transport Architecture III

- This display server is:
  - In a host that can be reached with IP address 192.168.0.1.
  - Our target X server is 1 (port 6000+1=6001)
  - Screen is 0.
- Another way to set the `displayname` is by a command line option that all X applications have: ”-display [displayname]“.
- Example:

```
$ xeyes -display 192.168.0.3:0.0
```

Activate TCP/IP

- In many modern Linux distributions, the X server is not configured to use the TCP/IP transport.
- If desired, we have to enable this feature in the display manager.
- In new Ubuntu distributions (>= 11.10) the default display manager is LightDM.
- Open the file /etc/lightdm/lightdm.conf (you must be root) and add the following content:

```
[Seat:0]
xserver-allow-tcp=true
```

- Reboot lightdm (or the system):

```
# /etc/init.d/lightdm restart
```

X authentication

- X has an authentication mechanism based on a list of allowed client hosts.
- The `xhost` command is used for this purpose.

`xhost`	shows the X server access control list.
`xhost +hostname`	Adds hostname to X server access control list.
`xhost -hostname`	Removes hostname from X server access control list.
`xhost +`	Turns off access control (all remote hosts will have access to X server).
`xhost -`	Turns access control back on.

- Executing `xhost` with the options that affect access control is only allowed on the host in which the display server is running.