

Tema I

Introducción a X-Window System (16 de febrero de 2011)

Programación en Entornos Interactivos.

16 de febrero de 2011

Dpto. Lenguajes y Sistemas Informáticos
Universidad de Alicante

Resumen

Visión general de X-Window System. Uso de X-Window System. Configuración a varios niveles de X-Window System. Ejecución de aplicaciones en X-Window System.

Preliminares.

- Desarrollado en el MIT con ayuda de DEC.
- Las versiones más importantes: **X10** y **X11**
- Adoptado como estándar por una serie de vendedores Hardware y Software.
- Disponible en la mayoría de sistemas Unix.
- No define un estilo de interfaz estándar.

Preliminares.

- Desarrollado en el MIT con ayuda de DEC.
- Las versiones más importantes: **X10** y **X11**
- Adoptado como estándar por una serie de vendedores Hardware y Software.
- Disponible en la mayoría de sistemas Unix.
- No define un estilo de interfaz estándar.

Preliminares.

- Desarrollado en el MIT con ayuda de DEC.
- Las versiones más importantes: **X10** y **X11**
- Adoptado como estándar por una serie de vendedores Hardware y Software.
- Disponible en la mayoría de sistemas Unix.
- No define un estilo de interfaz estándar.

Preliminares.

- Desarrollado en el MIT con ayuda de DEC.
- Las versiones más importantes: **X10** y **X11**
- Adoptado como estándar por una serie de vendedores Hardware y Software.
- Disponible en la mayoría de sistemas Unix.
- No define un estilo de interfaz estándar.

Preliminares.

- Desarrollado en el MIT con ayuda de DEC.
- Las versiones más importantes: **X10** y **X11**
- Adoptado como estándar por una serie de vendedores Hardware y Software.
- Disponible en la mayoría de sistemas Unix.
- No define un estilo de interfaz estándar.

Modelo Cliente-Servidor

- La arquitectura de X-Window System se basa en el modelo *Cliente-Servidor*.
- Existe un único proceso *servidor* que se encarga de gestionar todos los dispositivos de entrada y salida.
- Cualquier aplicación que use las facilidades proporcionadas por el servidor X, se denomina *cliente*.
- Los clientes se comunican con el servidor de manera asíncrona a través de una red. Esta comunicación puede ser N a N.
- Esto permite que el servidor y un cliente puedan estar ejecutándose en distintas máquinas conectadas en red.
- Lo primero que debe hacer un cliente es abrir una conexión con el servidor. Desde ese momento puede usar cualquier *screen* que sea controlada por el servidor.

Modelo Cliente-Servidor

- La arquitectura de X-Window System se basa en el modelo *Cliente-Servidor*.
- Existe un único proceso *servidor* que se encarga de gestionar todos los dispositivos de entrada y salida.
- Cualquier aplicación que use las facilidades proporcionadas por el servidor X, se denomina *cliente*.
- Los clientes se comunican con el servidor de manera asíncrona a través de una red. Esta comunicación puede ser N a N.
- Esto permite que el servidor y un cliente puedan estar ejecutándose en distintas máquinas conectadas en red.
- Lo primero que debe hacer un cliente es abrir una conexión con el servidor. Desde ese momento puede usar cualquier *screen* que sea controlada por el servidor.

Modelo Cliente-Servidor

- La arquitectura de X-Window System se basa en el modelo *Cliente-Servidor*.
- Existe un único proceso *servidor* que se encarga de gestionar todos los dispositivos de entrada y salida.
- Cualquier aplicación que use las facilidades proporcionadas por el servidor X, se denomina *cliente*.
- Los clientes se comunican con el servidor de manera asíncrona a través de una red. Esta comunicación puede ser N a N.
- Esto permite que el servidor y un cliente puedan estar ejecutándose en distintas máquinas conectadas en red.
- Lo primero que debe hacer un cliente es abrir una conexión con el servidor. Desde ese momento puede usar cualquier *screen* que sea controlada por el servidor.

Modelo Cliente-Servidor

- La arquitectura de X-Window System se basa en el modelo *Cliente-Servidor*.
- Existe un único proceso *servidor* que se encarga de gestionar todos los dispositivos de entrada y salida.
- Cualquier aplicación que use las facilidades proporcionadas por el servidor X, se denomina *cliente*.
- Los clientes se comunican con el servidor de manera asíncrona a través de una red. Esta comunicación puede ser N a N.
- Esto permite que el servidor y un cliente puedan estar ejecutándose en distintas máquinas conectadas en red.
- Lo primero que debe hacer un cliente es abrir una conexión con el servidor. Desde ese momento puede usar cualquier *screen* que sea controlada por el servidor.

Modelo Cliente-Servidor

- La arquitectura de X-Window System se basa en el modelo *Cliente-Servidor*.
- Existe un único proceso *servidor* que se encarga de gestionar todos los dispositivos de entrada y salida.
- Cualquier aplicación que use las facilidades proporcionadas por el servidor X, se denomina *cliente*.
- Los clientes se comunican con el servidor de manera asíncrona a través de una red. Esta comunicación puede ser N a N.
- Esto permite que el servidor y un cliente puedan estar ejecutándose en distintas máquinas conectadas en red.
- Lo primero que debe hacer un cliente es abrir una conexión con el servidor. Desde ese momento puede usar cualquier *screen* que sea controlada por el servidor.

Modelo Cliente-Servidor

- La arquitectura de X-Window System se basa en el modelo *Cliente-Servidor*.
- Existe un único proceso *servidor* que se encarga de gestionar todos los dispositivos de entrada y salida.
- Cualquier aplicación que use las facilidades proporcionadas por el servidor X, se denomina *cliente*.
- Los clientes se comunican con el servidor de manera asíncrona a través de una red. Esta comunicación puede ser N a N.
- Esto permite que el servidor y un cliente puedan estar ejecutándose en distintas máquinas conectadas en red.
- Lo primero que debe hacer un cliente es abrir una conexión con el servidor. Desde ese momento puede usar cualquier *screen* que sea controlada por el servidor.

Conceptos Display y Screen

- Denominaremos **display** a un único proceso servidor de **X**.
- Este display atiende, normalmente, a un ratón, un teclado y un monitor.
- Denominaremos **screen** al monitor en el que se realiza la salida gráfica.
- Un *display X* puede soportar diversos *screens*.
- Antiguamente se solía usar los términos de *display* y *screen* intercambiadamente.

Conceptos Display y Screen

- Denominaremos **display** a un único proceso servidor de **X**.
- Este display atiende, normalmente, a un ratón, un teclado y un monitor.
- Denominaremos **screen** al monitor en el que se realiza la salida gráfica.
- Un *display X* puede soportar diversos *screens*.
- Antiguamente se solía usar los términos de *display* y *screen* intercambiadamente.

Conceptos Display y Screen

- Denominaremos **display** a un único proceso servidor de **X**.
- Este display atiende, normalmente, a un ratón, un teclado y un monitor.
- Denominaremos **screen** al monitor en el que se realiza la salida gráfica.
- Un *display X* puede soportar diversos *screens*.
- Antiguamente se solía usar los términos de *display* y *screen* intercambiadamente.

Conceptos Display y Screen

- Denominaremos **display** a un único proceso servidor de **X**.
- Este display atiende, normalmente, a un ratón, un teclado y un monitor.
- Denominaremos **screen** al monitor en el que se realiza la salida gráfica.
- Un *display* X puede soportar diversos *screens*.
- Antiguamente se solía usar los términos de *display* y *screen* intercambiadamente.

Conceptos Display y Screen

- Denominaremos **display** a un único proceso servidor de **X**.
- Este display atiende, normalmente, a un ratón, un teclado y un monitor.
- Denominaremos **screen** al monitor en el que se realiza la salida gráfica.
- Un *display X* puede soportar diversos *screens*.
- Antiguamente se solía usar los términos de *display* y *screen* intercambiadamente.

Peticiones -Requests-

- Cualquier petición de servicio que quiera llevar a cabo un cliente es solicitada por éste al servidor -*crear y destruir ventanas, realizar salida gráfica, etc. . .*-.
 - El Servidor X se ejecuta de manera asíncrona* respecto a los clientes, y éstos respecto al servidor.
 - Las peticiones no tienen porqué ser atendidas por el servidor en el orden en el que llegaron, tampoco de manera inmediata. Estas peticiones van a parar a una cola.

Peticiones -Requests-

- Cualquier petición de servicio que quiera llevar a cabo un cliente es solicitada por éste al servidor -*crear y destruir ventanas, realizar salida gráfica, etc. . .*-.
 - El Servidor X se ejecuta de manera asíncrona* respecto a los clientes, y éstos respecto al servidor.
 - Las peticiones no tienen porqué ser atendidas por el servidor en el orden en el que llegaron, tampoco de manera inmediata. Estas peticiones van a parar a una cola.

Peticiones -Requests-

- Cualquier petición de servicio que quiera llevar a cabo un cliente es solicitada por éste al servidor -*crear y destruir ventanas, realizar salida gráfica, etc. . .* -.
- El Servidor X se ejecuta de manera asíncrona* respecto a los clientes, y éstos respecto al servidor.
- Las peticiones no tienen porqué ser atendidas por el servidor en el orden en el que llegaron, tampoco de manera inmediata. Estas peticiones van a parar a una cola.

Ventanas (I)

- Constituyen uno de los conceptos clave en X. Son un área rectangular de la pantalla.
- No tienen título, barras de desplazamiento (*scroll*) u otros adornos, solo un borde delimitador.
- Se pueden combinar varias ventanas para crear títulos, barras de *scroll*, etc. . .
- Una ventana se crea a petición de un cliente, pero puede ser manipulada por peticiones de cualquier otro.
- Las ventanas se organizan jerárquicamente (toda ventana tiene un ascendente), siendo la ventana principal la que se denomina ventana raíz (*root window*), la cual no tiene ascendente.

Ventanas (I)

- Constituyen uno de los conceptos clave en X. Son un área rectangular de la pantalla.
- No tienen título, barras de desplazamiento (*scroll*) u otros adornos, solo un borde delimitador.
- Se pueden combinar varias ventanas para crear títulos, barras de *scroll*, etc. . .
- Una ventana se crea a petición de un cliente, pero puede ser manipulada por peticiones de cualquier otro.
- Las ventanas se organizan jerárquicamente (toda ventana tiene un ascendente), siendo la ventana principal la que se denomina ventana raíz (**root window**), la cual no tiene ascendente.

Ventanas (I)

- Constituyen uno de los conceptos clave en X. Son un área rectangular de la pantalla.
- No tienen título, barras de desplazamiento (*scroll*) u otros adornos, solo un borde delimitador.
- Se pueden combinar varias ventanas para crear títulos, barras de *scroll*, etc. . .
- Una ventana se crea a petición de un cliente, pero puede ser manipulada por peticiones de cualquier otro.
- Las ventanas se organizan jerárquicamente (toda ventana tiene un ascendente), siendo la ventana principal la que se denomina ventana raíz (**root window**), la cual no tiene ascendente.

Ventanas (I)

- Constituyen uno de los conceptos clave en X. Son un área rectangular de la pantalla.
- No tienen título, barras de desplazamiento (*scroll*) u otros adornos, solo un borde delimitador.
- Se pueden combinar varias ventanas para crear títulos, barras de *scroll*, etc. . .
- Una ventana se crea a petición de un cliente, pero puede ser manipulada por peticiones de cualquier otro.
- Las ventanas se organizan jerárquicamente (toda ventana tiene un ascendente), siendo la ventana principal la que se denomina ventana raíz (**root window**), la cual no tiene ascendente.

Ventanas (I)

- Constituyen uno de los conceptos clave en X. Son un área rectangular de la pantalla.
- No tienen título, barras de desplazamiento (*scroll*) u otros adornos, solo un borde delimitador.
- Se pueden combinar varias ventanas para crear títulos, barras de *scroll*, etc. . .
- Una ventana se crea a petición de un cliente, pero puede ser manipulada por peticiones de cualquier otro.
- Las ventanas se organizan jerárquicamente (toda ventana tiene un ascendente), siendo la ventana principal la que se denomina ventana raíz (**root window**), la cual no tiene ascendente.

Ventanas (II)

- El servidor X crea una ventana raíz por cada *screen* que controla. Esta ventana:
 - Ocupa toda la pantalla.
 - No puede ser movida o cambiar de tamaño
- Las ventanas pueden solaparse.
- Cada ventana tiene su propio sistema de coordenadas que comienza en la esquina superior izquierda.

Ventanas (II)

- El servidor X crea una ventana raíz por cada *screen* que controla. Esta ventana:
 - Ocupa toda la pantalla.
 - No puede ser movida o cambiar de tamaño
- Las ventanas pueden solaparse.
- Cada ventana tiene su propio sistema de coordenadas que comienza en la esquina superior izquierda.

Ventanas (II)

- El servidor X crea una ventana raíz por cada *screen* que controla. Esta ventana:
 - Ocupa toda la pantalla.
 - No puede ser movida o cambiar de tamaño
- Las ventanas pueden solaparse.
- Cada ventana tiene su propio sistema de coordenadas que comienza en la esquina superior izquierda.

Ventanas (II)

- El servidor X crea una ventana raíz por cada *screen* que controla. Esta ventana:
 - Ocupa toda la pantalla.
 - No puede ser movida o cambiar de tamaño
- Las ventanas pueden solaparse.
- Cada ventana tiene su propio sistema de coordenadas que comienza en la esquina superior izquierda.

Ventanas (II)

- El servidor X crea una ventana raíz por cada *screen* que controla. Esta ventana:
 - Ocupa toda la pantalla.
 - No puede ser movida o cambiar de tamaño
- Las ventanas pueden solaparse.
- Cada ventana tiene su propio sistema de coordenadas que comienza en la esquina superior izquierda.

Contenido de una Ventana

- Una ventana no es visible, y tampoco sus contenidos, hasta que después de ser creada por el servidor, es 'map-eada'.
- Una ventana puede estar 'mapeada' y seguir sin ser visible por:
 - Estar oculta completamente por otras ventanas.
 - Algún ascendente no está mapeado.
 - La ventana está totalmente "desajustada" por un ascendente.
- El mantenimiento del contenido de una ventana puede ser responsabilidad del cliente o del servidor (expose, retained-raster, save-under).

Contenido de una Ventana

- Una ventana no es visible, y tampoco sus contenidos, hasta que después de ser creada por el servidor, es 'map-eada'.
- Una ventana puede estar 'mapeada' y seguir sin ser visible por:
 - Estar oculta completamente por otras ventanas.
 - Algún ascendente no está mapeado.
 - La ventana está totalmente "recortada" por un ascendente.
- El mantenimiento del contenido de una ventana puede ser responsabilidad del cliente o del servidor (expose, retained-raster, save-under).

Contenido de una Ventana

- Una ventana no es visible, y tampoco sus contenidos, hasta que después de ser creada por el servidor, es 'map-eada'.
- Una ventana puede estar 'mapeada' y seguir sin ser visible por:
 - Estar oculta completamente por otras ventanas.
 - Algún ascendente no está mapeado.
 - La ventana está totalmente "recortada" por un ascendente.
- El mantenimiento del contenido de una ventana puede ser responsabilidad del cliente o del servidor (expose, retained-raster, save-under).

Contenido de una Ventana

- Una ventana no es visible, y tampoco sus contenidos, hasta que después de ser creada por el servidor, es ‘*map*-eada’.
- Una ventana puede estar ‘mapeada’ y seguir sin ser visible por:
 - Estar oculta completamente por otras ventanas.
 - Algún ascendente no está mapeado.
 - La ventana está totalmente “recortada” por un ascendente.
- El mantenimiento del contenido de una ventana puede ser responsabilidad del cliente o del servidor (expose, retained-raster, save-under).

Contenido de una Ventana

- Una ventana no es visible, y tampoco sus contenidos, hasta que después de ser creada por el servidor, es ‘*map*-eada’.
- Una ventana puede estar ‘mapeada’ y seguir sin ser visible por:
 - Estar oculta completamente por otras ventanas.
 - Algún ascendente no está mapeado.
 - La ventana está totalmente “recortada” por un ascendente.
- El mantenimiento del contenido de una ventana puede ser responsabilidad del cliente o del servidor (expose, retained-raster, save-under).

Contenido de una Ventana

- Una ventana no es visible, y tampoco sus contenidos, hasta que después de ser creada por el servidor, es '*map*-eada'.
- Una ventana puede estar 'mapeada' y seguir sin ser visible por:
 - Estar oculta completamente por otras ventanas.
 - Algún ascendente no está mapeado.
 - La ventana está totalmente "recortada" por un ascendente.
- El mantenimiento del contenido de una ventana puede ser responsabilidad del cliente o del servidor (expose, retained-raster, save-under).

Comunicación Cliente - Servidor

- Ya hemos visto como el cliente se comunica con el servidor: enviándole peticiones.
- El servidor se comunica con los clientes por medio de **eventos**, bien predefinidos, bien definidos por el propio cliente.
- Los eventos se envían a los clientes colocándolos en una cola.
- La mayoría de aplicaciones X son dirigidas por eventos *—event-driven—*.

Comunicación Cliente - Servidor

- Ya hemos visto como el cliente se comunica con el servidor: enviándole peticiones.
- El servidor se comunica con los clientes por medio de **eventos**, bien predefinidos, bien definidos por el propio cliente.
- Los eventos se envían a los clientes colocándolos en una cola.
- La mayoría de aplicaciones X son dirigidas por eventos *–event-driven–*.

Comunicación Cliente - Servidor

- Ya hemos visto como el cliente se comunica con el servidor: enviándole peticiones.
- El servidor se comunica con los clientes por medio de **eventos**, bien predefinidos, bien definidos por el propio cliente.
- Los eventos se envían a los clientes colocándolos en una cola.
- La mayoría de aplicaciones X son dirigidas por eventos *-event-driven-*.

Comunicación Cliente - Servidor

- Ya hemos visto como el cliente se comunica con el servidor: enviándole peticiones.
- El servidor se comunica con los clientes por medio de **eventos**, bien predefinidos, bien definidos por el propio cliente.
- Los eventos se envían a los clientes colocándolos en una cola.
- La mayoría de aplicaciones X son dirigidas por eventos –*event-driven*–.

Gestores de Ventanas

- X no proporciona ningún gestor de ventanas –*Window-manager*– de forma estándar, pero sí facilidades para que el usuario utilice el que quiera.
- Un gestor de ventanas es una aplicación “cliente X” como otro cualquiera.
- Hay disponible una gran variedad de gestores de ventanas para X:
 - *twm* (The Window Manager)
 - *fvwm* (The Flat-looked Virtual Window Manager)
 - *icwm* (The Ice Window Manager)
 - *lwm* (The Linux Window Manager)
 - *metacity* (The Metacity Window Manager)
 - *openbox* (The Openbox Window Manager)
 - *xfwm* (The Xfce Window Manager)
 - *xmms* (The X Window Manager)
 - *xwm* (The X Window Manager)
 - *xwmwm* (The X Window Manager)
 - *xwmwm* (The X Window Manager)
 - *xwmwm* (The X Window Manager)
- Un “cliente X” debería poder funcionar de manera apropiada independientemente del gestor de ventanas usado, e incluso sin ninguno de ellos.

Gestores de Ventanas

- X no proporciona ningún gestor de ventanas –*Window-manager*– de forma estándar, pero sí facilidades para que el usuario utilice el que quiera.
- Un gestor de ventanas es una aplicación “cliente X” como otro cualquiera.
- Hay disponible una gran variedad de gestores de ventanas para X:
 - MWM, TWM, VTWM
 - OLYWM, FVWM, FVWM2, FVWM95
 - <http://www.gnome.org/>
 - <http://www.kde.org/>
 - <http://www.xfce.org/>
 - <http://www.cinnamon-project.com/>
 - <http://www.lxde.org/>
 - <http://www.ubuntu.com/>
- Un “cliente X” debería poder funcionar de manera apropiada independientemente del gestor de ventanas usado, e incluso sin ninguno de ellos.

Gestores de Ventanas

- X no proporciona ningún gestor de ventanas –*Window-manager*– de forma estándar, pero sí facilidades para que el usuario utilice el que quiera.
- Un gestor de ventanas es una aplicación “cliente X” como otro cualquiera.
- Hay disponible una gran variedad de gestores de ventanas para X:
 - MWM, TWM, VTWM
 - OLVWM, FVWM, FVWM2, FVWM95
 - AWM, AfterStep, WindowMaker
 - Kwm, sawfish, metacity (mutter), enlightenment, compiz, etc.
- Un “cliente X” debería poder funcionar de manera apropiada independientemente del gestor de ventanas usado, e incluso sin ninguno de ellos.

Gestores de Ventanas

- X no proporciona ningún gestor de ventanas –*Window-manager*– de forma estándar, pero sí facilidades para que el usuario utilice el que quiera.
- Un gestor de ventanas es una aplicación “cliente X” como otro cualquiera.
- Hay disponible una gran variedad de gestores de ventanas para X:
 - MWM, TWM, VTWM
 - OLVWM, FVWM, FVWM2, FVWM95
 - AWM, AfterStep, WindowMaker
 - Kwm, sawfish, metacity (mutter), enlightenment , compiz , etc. ...
- Un “cliente X” debería poder funcionar de manera apropiada independientemente del gestor de ventanas usado, e incluso sin ninguno de ellos.

Gestores de Ventanas

- X no proporciona ningún gestor de ventanas –*Window-manager*– de forma estándar, pero sí facilidades para que el usuario utilice el que quiera.
- Un gestor de ventanas es una aplicación “cliente X” como otro cualquiera.
- Hay disponible una gran variedad de gestores de ventanas para X:
 - MWM, TWM, VTWM
 - OLVWM, FVWM, FVWM2, FVWM95
 - AWM, AfterStep, WindowMaker
 - Kwm, sawfish, metacity (mutter), *enlightenment* , *compiz* , etc. ...
- Un “cliente X” debería poder funcionar de manera apropiada independientemente del gestor de ventanas usado, e incluso sin ninguno de ellos.

Gestores de Ventanas

- X no proporciona ningún gestor de ventanas –*Window-manager*– de forma estándar, pero sí facilidades para que el usuario utilice el que quiera.
- Un gestor de ventanas es una aplicación “cliente X” como otro cualquiera.
- Hay disponible una gran variedad de gestores de ventanas para X:
 - MWM, TWM, VTWM
 - OLVWM, FVWM, FVWM2, FVWM95
 - AWM, AfterStep, WindowMaker
 - Kwm, sawfish, metacity (mutter), [enlightenment](#) , [compiz](#) , etc. ...
- Un “cliente X” debería poder funcionar de manera apropiada independientemente del gestor de ventanas usado, e incluso sin ninguno de ellos.

Gestores de Ventanas

- X no proporciona ningún gestor de ventanas –*Window-manager*– de forma estándar, pero sí facilidades para que el usuario utilice el que quiera.
- Un gestor de ventanas es una aplicación “cliente X” como otro cualquiera.
- Hay disponible una gran variedad de gestores de ventanas para X:
 - MWM, TWM, VTWM
 - OLVWM, FVWM, FVWM2, FVWM95
 - AWM, AfterStep, WindowMaker
 - Kwm, sawfish, metacity (mutter), **enlightenment** , **compiz** , etc. . .
- Un “cliente X” debería poder funcionar de manera apropiada independientemente del gestor de ventanas usado, e incluso sin ninguno de ellos.

Gestores de Ventanas

- X no proporciona ningún gestor de ventanas –*Window-manager*– de forma estándar, pero sí facilidades para que el usuario utilice el que quiera.
- Un gestor de ventanas es una aplicación “cliente X” como otro cualquiera.
- Hay disponible una gran variedad de gestores de ventanas para X:
 - MWM, TWM, VTWM
 - OLVWM, FVWM, FVWM2, FVWM95
 - AWM, AfterStep, WindowMaker
 - Kwm, sawfish, metacity (mutter), **enlightenment** , **compiz** , etc. . .
- Un “cliente X” debería poder funcionar de manera apropiada independientemente del gestor de ventanas usado, e incluso sin ninguno de ellos.

- El interface con X de más bajo nivel lo proporciona **Xlib**¹, y está disponible en **C**.
- Por encima de Xlib/xcb existen los llamados **Toolkits**, más sencillos de utilizar (InterViews, Andrew, XForms, Qt, Gtk/Gtkmm, etc. . .).
- Existe un toolkit estándar: **X-Toolkit**, que consta de dos capas, **Xt Intrinsic**s y una serie de componentes de *interfaz de usuario* que genéricamente se llaman **Widgets**:

¹Xcb: xcb.freedesktop.org.

- El interface con X de más bajo nivel lo proporciona **Xlib**¹, y está disponible en **C**.
- Por encima de Xlib/xcb existen los llamados **Toolkits**, más sencillos de utilizar (InterViews, Andrew, XForms, Qt, Gtk/Gtkmm, etc. . .).
- Existe un toolkit estándar: **X-Toolkit**, que consta de dos capas, **Xt Intrinsic**s y una serie de componentes de *interfaz de usuario* que genéricamente se llaman **Widgets**:

¹Xcb: xcb.freedesktop.org.

- El interface con X de más bajo nivel lo proporciona **Xlib**¹, y está disponible en **C**.
- Por encima de Xlib/xcb existen los llamados **Toolkits**, más sencillos de utilizar (InterViews, Andrew, XForms, Qt, Gtk/Gtkmm, etc. . .).
- Existe un toolkit estándar: **X-Toolkit**, que consta de dos capas, **Xt Intrinsic**s y una serie de componentes de *interfaz de usuario* que genéricamente se llaman **Widgets**:

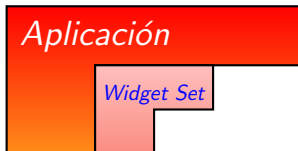
¹Xcb: xcb.freedesktop.org.

- El interface con X de más bajo nivel lo proporciona **Xlib**¹, y está disponible en **C**.
- Por encima de Xlib/xcb existen los llamados **Toolkits**, más sencillos de utilizar (InterViews, Andrew, XForms, Qt, Gtk/Gtkmm, etc. . .).
- Existe un toolkit estándar: **X-Toolkit**, que consta de dos capas, **Xt Intrinsic**s y una serie de componentes de *interfaz de usuario* que genéricamente se llaman **Widgets**:



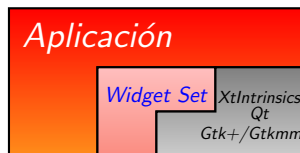
¹Xcb: xcb.freedesktop.org.

- El interface con X de más bajo nivel lo proporciona **Xlib**¹, y está disponible en **C**.
- Por encima de Xlib/xcb existen los llamados **Toolkits**, más sencillos de utilizar (InterViews, Andrew, XForms, Qt, Gtk/Gtkmm, etc. . .).
- Existe un toolkit estándar: **X-Toolkit**, que consta de dos capas, **Xt Intrinsic** y una serie de componentes de *interfaz de usuario* que genéricamente se llaman **Widgets**:



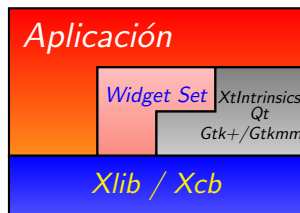
¹Xcb: xcb.freedesktop.org.

- El interface con X de más bajo nivel lo proporciona **Xlib**¹, y está disponible en **C**.
- Por encima de Xlib/xcb existen los llamados **Toolkits**, más sencillos de utilizar (InterViews, Andrew, XForms, Qt, Gtk/Gtkmm, etc. . .).
- Existe un toolkit estándar: **X-Toolkit**, que consta de dos capas, **Xt Intrinsics** y una serie de componentes de *interfaz de usuario* que genéricamente se llaman **Widgets**:



¹Xcb: xcb.freedesktop.org.

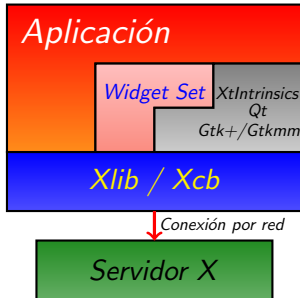
- El interface con X de más bajo nivel lo proporciona **Xlib**¹, y está disponible en **C**.
- Por encima de Xlib/xcb existen los llamados **Toolkits**, más sencillos de utilizar (InterViews, Andrew, XForms, Qt, Gtk/Gtkmm, etc. . .).
- Existe un toolkit estándar: **X-Toolkit**, que consta de dos capas, **Xt Intrinsics** y una serie de componentes de *interfaz de usuario* que genéricamente se llaman **Widgets**:



¹Xcb: xcb.freedesktop.org.

El API de X

- El interface con X de más bajo nivel lo proporciona **Xlib**¹, y está disponible en **C**.
- Por encima de Xlib/xcb existen los llamados **Toolkits**, más sencillos de utilizar (InterViews, Andrew, XForms, Qt, Gtk/Gtkmm, etc. . .).
- Existe un toolkit estándar: **X-Toolkit**, que consta de dos capas, **Xt Intrinsic**s y una serie de componentes de *interfaz de usuario* que genéricamente se llaman **Widgets**:



¹Xcb: xcb.freedesktop.org.

Uso de X-Window (I)

- Debemos iniciar la sesión *-poner en marcha el servidor X-* con: **startx**, o **openwin**, o **xinit**. Incluso podríamos probar con **X [-probeonly 2>err]**.
- Ejecución remota: variable `export DISPLAY = host:display.screen`. También disponible como opción “**-display host:display.screen**” o también con la opción **-X** de `ssh`.
- Para gráficos remotos es necesario dar permisos a las máquinas que mostrarán gráficos en la nuestra: **xhost + ó -**.
- Las siguientes combinaciones de teclas son *especiales* para el servidor **X**:

Uso de X-Window (I)

- Debemos iniciar la sesión *-poner en marcha el servidor X-* con: **startx**, o **openwin**, o **xinit**. Incluso podríamos probar con **X [-probeonly 2>err]**.
- Ejecución remota: variable `export DISPLAY = host:display.screen`. También disponible como opción “**-display host:display.screen**” o también con la opción **-X** de `ssh`.
- Para gráficos remotos es necesario dar permisos a las máquinas que mostrarán gráficos en la nuestra: **xhost + ó -**.
- Las siguientes combinaciones de teclas son *especiales* para el servidor **X**:

Uso de X-Window (I)

- Debemos iniciar la sesión *-poner en marcha el servidor X-* con: **startx**, o **openwin**, o **xinit**. Incluso podríamos probar con **X [-probeonly 2>err]**.
- Ejecución remota: variable `export DISPLAY = host:display.screen`. También disponible como opción “**-display host:display.screen**” o también con la opción **-X** de `ssh`.
- Para gráficos remotos es necesario dar permisos a las máquinas que mostrarán gráficos en la nuestra: **xhost + ó -**.
- Las siguientes combinaciones de teclas son *especiales* para el servidor X:

• Control+Alt++

• Control+Alt+-

Uso de X-Window (I)

- Debemos iniciar la sesión *-poner en marcha el servidor X-* con: **startx**, o **openwin**, o **xinit**. Incluso podríamos probar con **X [-probeonly 2>err]**.
- Ejecución remota: variable `export DISPLAY = host:display.screen`. También disponible como opción “**-display host:display.screen**” o también con la opción **-X** de `ssh`.
- Para gráficos remotos es necesario dar permisos a las máquinas que mostrarán gráficos en la nuestra: **xhost + ó -**.
- Las siguientes combinaciones de teclas son *especiales* para el servidor **X**:
 - **Control+Alt++**
 - **Control+Alt+-**
 - **Control+Alt+BackSpace**

Uso de X-Window (I)

- Debemos iniciar la sesión *-poner en marcha el servidor X-* con: **startx**, o **openwin**, o **xinit**. Incluso podríamos probar con **X [-probeonly 2>err]**.
- Ejecución remota: variable `export DISPLAY = host:display.screen`. También disponible como opción “**-display host:display.screen**” o también con la opción **-X** de `ssh`.
- Para gráficos remotos es necesario dar permisos a las máquinas que mostrarán gráficos en la nuestra: **xhost + ó -**.
- Las siguientes combinaciones de teclas son *especiales* para el servidor **X**:
 - **Control+Alt++**
 - Control+Alt+-
 - Control+Alt+BackSpace

Uso de X-Window (I)

- Debemos iniciar la sesión *-poner en marcha el servidor X-* con: **startx**, o **openwin**, o **xinit**. Incluso podríamos probar con **X [-probeonly 2>err]**.
- Ejecución remota: variable `export DISPLAY = host:display.screen`. También disponible como opción “**-display host:display.screen**” o también con la opción **-X** de `ssh`.
- Para gráficos remotos es necesario dar permisos a las máquinas que mostrarán gráficos en la nuestra: **xhost + ó -**.
- Las siguientes combinaciones de teclas son *especiales* para el servidor **X**:
 - **Control+Alt++**
 - **Control+Alt+-**
 - **Control+Alt+BackSpace**

Uso de X-Window (I)

- Debemos iniciar la sesión *-poner en marcha el servidor X-* con: **startx**, o **openwin**, o **xinit**. Incluso podríamos probar con **X [-probeonly 2>err]**.
- Ejecución remota: variable `export DISPLAY = host:display.screen`. También disponible como opción “**-display host:display.screen**” o también con la opción **-X** de `ssh`.
- Para gráficos remotos es necesario dar permisos a las máquinas que mostrarán gráficos en la nuestra: **xhost + ó -**.
- Las siguientes combinaciones de teclas son *especiales* para el servidor **X**:
 - **Control+Alt++**
 - **Control+Alt+-**
 - **Control+Alt+BackSpace**

Uso de X-Window (II)

- Debemos elegir el gestor de ventanas —**Window Manager**— que queramos ejecutar en cada ocasión, ya que **X** no proporciona ninguno: `fvwm`, `windowmaker`, `sawfish`, `twm`, `kwm`, `metacity`, etc...
- Existen los llamados *gestores de ventanas virtuales*, los cuales proporcionan un *‘escritorio virtual.’*

Uso de X-Window (II)

- Debemos elegir el gestor de ventanas —**Window Manager**— que queramos ejecutar en cada ocasión, ya que **X** no proporciona ninguno: `fvwm`, `windowmaker`, `sawfish`, `twm`, `kwm`, `metacity`, etc...
- Existen los llamados *gestores de ventanas virtuales*, los cuales proporcionan un '*escritorio virtual*.'

Trabajo con Ventanas (I)

- Existe una ventana, antecesora de todas, llamada ventana raíz —**Root Window**—.
- La apariencia de una ventana (bordes, título, botones, comportamiento, etc. . .) depende del *gestor de ventanas* utilizado.
- Gestores de ventanas basados en 'embaldosado' (*tiling*).

Trabajo con Ventanas (I)

- Existe una ventana, antecesora de todas, llamada ventana raíz —**Root Window**—.
- La apariencia de una ventana (bordes, título, botones, comportamiento, etc. . .) depende del *gestor de ventanas* utilizado.
- Gestores de ventanas basados en 'embaldosado' (*tiling*).

Trabajo con Ventanas (I)

- Existe una ventana, antecesora de todas, llamada ventana raíz —**Root Window**—.
- La apariencia de una ventana (bordes, título, botones, comportamiento, etc. . .) depende del *gestor de ventanas* utilizado.
- Gestores de ventanas basados en 'embaldosado' (*tiling*).

Trabajo con Ventanas (II)

- Tenemos asociados menús a las ventanas, incluida la ventana **root**, la cual puede tener un menú asociado a cada botón del ratón.
- Los **xterm** presentan 3 menús pulsando **Control**+botón del ratón:
 - Control+botón 1: *main options*
 - Control+botón 2: *VT options*
 - Control+botón 3: *VT Font*
- Existen otros emuladores de terminal como *kterm* y *gnome-terminal* o *ROXTerm*.

Trabajo con Ventanas (II)

- Tenemos asociados menús a las ventanas, incluida la ventana **root**, la cual puede tener un menú asociado a cada botón del ratón.
- Los **xterm** presentan 3 menús pulsando **Control**+botón del ratón:
 - **Control**+botón 1: *main optios*
 - **Control**+botón 2: *VT options*
 - **Control**+botón 3: *VT Fonts*.
- Existen otros emuladores de terminal como *kterm* y *gnome-terminal* o *ROXTerm*.

Trabajo con Ventanas (II)

- Tenemos asociados menús a las ventanas, incluida la ventana **root**, la cual puede tener un menú asociado a cada botón del ratón.
- Los **xterm** presentan 3 menús pulsando **Control**+botón del ratón:
 - **Control**+botón 1: *main optios*
 - **Control**+botón 2: *VT options*
 - **Control**+botón 3: *VT Fonts*.
- Existen otros emuladores de terminal como *kterm* y *gnome-terminal* o *ROXTerm*.

Trabajo con Ventanas (II)

- Tenemos asociados menús a las ventanas, incluida la ventana **root**, la cual puede tener un menú asociado a cada botón del ratón.
- Los **xterm** presentan 3 menús pulsando **Control**+botón del ratón:
 - **Control**+botón 1: *main optios*
 - **Control**+botón 2: *VT options*
 - **Control**+botón 3: *VT Fonts*.
- Existen otros emuladores de terminal como *kterm* y *gnome-terminal* o *ROXTerm*.

Trabajo con Ventanas (II)

- Tenemos asociados menús a las ventanas, incluida la ventana **root**, la cual puede tener un menú asociado a cada botón del ratón.
- Los **xterm** presentan 3 menús pulsando **Control**+botón del ratón:
 - **Control**+botón 1: *main optios*
 - **Control**+botón 2: *VT options*
 - **Control**+botón 3: *VT Fonts*.
- Existen otros emuladores de terminal como *kterm* y *gnome-terminal* o *ROXTerm*.

Trabajo con Ventanas (II)

- Tenemos asociados menús a las ventanas, incluida la ventana **root**, la cual puede tener un menú asociado a cada botón del ratón.
- Los **xterm** presentan 3 menús pulsando **Control**+botón del ratón:
 - **Control**+botón 1: *main optios*
 - **Control**+botón 2: *VT options*
 - **Control**+botón 3: *VT Fonts*.
- Existen otros emuladores de terminal como *kterm* y *gnome-terminal* o *ROXTerm*.

Configuración de X-Window System (I)

- La configuración más básica, relaciona hardware con software, la realizamos en el fichero **xorg.conf**², el cual suele estar en '/etc/X11'.
- La configuración de distintas opciones de las aplicaciones se realiza para cada usuario en el fichero **.Xdefaults** o **.Xresources**, y globalmente, en un fichero, por cada aplicación, situado en /usr/lib/X11/app-defaults.
- El fichero **.Xdefaults** o **.Xresources** se lee por el servidor **X** al arrancar éste, y se puede releer con el comando "xrdb [-merge] ~/.Xdefaults" o "xrdb [-merge] ~/.Xresources".

²Llamado XF86Config en la implementación de XFree86: 

Configuración de X-Window System (I)

- La configuración más básica, relaciona hardware con software, la realizamos en el fichero **xorg.conf**², el cual suele estar en '/etc/X11'.
- La configuración de distintas opciones de las aplicaciones se realiza para cada usuario en el fichero **.Xdefaults** o **.Xresources**, y globalmente, en un fichero, por cada aplicación, situado en /usr/lib/X11/app-defaults.
- El fichero **.Xdefaults** o **.Xresources** se lee por el servidor **X** al arrancar éste, y se puede releer con el comando "xrdb [-merge] ~/.Xdefaults" o "xrdb [-merge] ~/.Xresources".

²Llamado XF86Config en la implementación de XFree86.

Configuración de X-Window System (I)

- La configuración más básica, relaciona hardware con software, la realizamos en el fichero **xorg.conf**², el cual suele estar en '/etc/X11'.
- La configuración de distintas opciones de las aplicaciones se realiza para cada usuario en el fichero **.Xdefaults** o **.Xresources**, y globalmente, en un fichero, por cada aplicación, situado en /usr/lib/X11/app-defaults.
- El fichero **.Xdefaults** o **.Xresources** se lee por el servidor **X** al arrancar éste, y se puede releer con el comando “**xrdb [-merge] ~/.Xdefaults**” o “**xrdb [-merge] ~/.Xresources**”.

²Llamado XF86Config en la implementación de XFree86:

Configuración de X-Window System (II)

- En **.xinitrc** o **.xsession** indicamos lo que queremos ejecutar al arrancar el servidor **X**, incluido el *manejador de ventanas*. La versión general de éste fichero está en `/usr/X11R6/lib/X11/xinit/xinitrc`.
- Fichero específico del *manejador de ventanas* elegido `-.fvwmrc`, `.fvwm2rc`, `.fvwm2rc95`, etc...-.
- Algunas órdenes de configuración: `xset`, `xsetroot`, `xmodmap`, `xev`, `xlsfonts`, `xfontsel`.

Configuración de X-Window System (II)

- En **.xinitrc** o **.xsession** indicamos lo que queremos ejecutar al arrancar el servidor **X**, incluido el *manejador de ventanas*. La versión general de éste fichero está en `/usr/X11R6/lib/X11/xinit/xinitrc`.
- Fichero específico del *manejador de ventanas* elegido - `.fvwmrc`, `.fvwm2rc`, `.fvwm2rc95`, etc...-.
- Algunas órdenes de configuración: `xset`, `xsetroot`, `xmodmap`, `xev`, `xlsfonts`, `xfontsel`.

Configuración de X-Window System (II)

- En **.xinitrc** o **.xsession** indicamos lo que queremos ejecutar al arrancar el servidor **X**, incluido el *manejador de ventanas*. La versión general de éste fichero está en `/usr/X11R6/lib/X11/xinit/xinitrc`.
- Fichero específico del *manejador de ventanas* elegido `-.fvwmrc`, `.fvwm2rc`, `.fvwm2rc95`, etc...-.
- Algunas órdenes de configuración: `xset`, `xsetroot`, `xmodmap`, `xev`, `xlsfonts`, `xfontsel`.

Ejecución de Aplicaciones en X-Window (I)

- Existen dos tipos de aplicaciones: **clientes** *-window smart-* y **no clientes** *-ejecutadas en un emulador de terminal-*.
- Las aplicaciones se pueden ejecutar desde el intérprete de órdenes o eligiendo una opción desde un menú.
- Algunos clientes: `xterm`, `rxvt`, `xclock`, `xload`, `xcalc`, `xkill`, etc...
- La mayoría de clientes aceptan una serie de opciones estándar al iniciarlos: `-bg color`, `-fg color`, `-fn font (/usr/lib/X11/fonts)`.
- La ejecución de **no clientes** debe hacerse desde un `xterm`: `"xterm -e top"`.

Ejecución de Aplicaciones en X-Window (I)

- Existen dos tipos de aplicaciones: **clientes** *-window smart-* y **no clientes** *-ejecutadas en un emulador de terminal-*.
- Las aplicaciones se pueden ejecutar desde el intérprete de órdenes o eligiendo una opción desde un menú.
- Algunos clientes: `xterm`, `rxvt`, `xclock`, `xload`, `xcalc`, `xkill`, etc...
- La mayoría de clientes aceptan una serie de opciones estándar al iniciarlos: `-bg color`, `-fg color`, `-fn font (/usr/lib/X11/fonts)`.
- La ejecución de **no clientes** debe hacerse desde un `xterm` : `"xterm -e top"`.

Ejecución de Aplicaciones en X-Window (I)

- Existen dos tipos de aplicaciones: **clientes** *-window smart-* y **no clientes** *-ejecutadas en un emulador de terminal-*.
- Las aplicaciones se pueden ejecutar desde el intérprete de órdenes o eligiendo una opción desde un menú.
- Algunos clientes: `xterm`, `rxvt`, `xclock`, `xload`, `xcalc`, `xkill`, etc...
- La mayoría de clientes aceptan una serie de opciones estándar al iniciarlos: `-bg color`, `-fg color`, `-fn font (/usr/lib/X11/fonts)`.
- La ejecución de **no clientes** debe hacerse desde un `xterm` : `"xterm -e top"`.

Ejecución de Aplicaciones en X-Window (I)

- Existen dos tipos de aplicaciones: **clientes** *-window smart-* y **no clientes** *-ejecutadas en un emulador de terminal-*.
- Las aplicaciones se pueden ejecutar desde el intérprete de órdenes o eligiendo una opción desde un menú.
- Algunos clientes: `xterm`, `rxvt`, `xclock`, `xload`, `xcalc`, `xkill`, etc...
- La mayoría de clientes aceptan una serie de opciones estándar al iniciarlos: `-bg color`, `-fg color`, `-fn font (/usr/lib/X11/fonts)`.
- La ejecución de **no clientes** debe hacerse desde un `xterm` : `"xterm -e top"`.

Ejecución de Aplicaciones en X-Window (I)

- Existen dos tipos de aplicaciones: **clientes** *-window smart-* y **no clientes** *-ejecutadas en un emulador de terminal-*.
- Las aplicaciones se pueden ejecutar desde el intérprete de órdenes o eligiendo una opción desde un menú.
- Algunos clientes: `xterm`, `rxvt`, `xclock`, `xload`, `xcalc`, `xkill`, etc...
- La mayoría de clientes aceptan una serie de opciones estándar al iniciarlos: `-bg color`, `-fg color`, `-fn font (/usr/lib/X11/fonts)`.
- La ejecución de **no clientes** debe hacerse desde un **xterm** : `"xterm -e top"`.

Ejecución de Aplicaciones en X-Window (II)

- Paro 'apropiado' de **clientes**, a veces, mediante opción '*close*' del menú local, o de la opción que tengan para ello.
- Paro 'apropiado' de **no clientes**, de la opción que tengan para ello.
- Otros modos de 'paro': **Control+c**, **Control+d**.
- Paso a 2º plano: **Control+z**, y uso de los comandos: `jobs ([1]+ Running xterm &)`, `bg%num-trabajo`, `fg%num-trabajo`.
- Una mayoría de aplicaciones de **X11** permiten la copia de información con el ratón. Pinchando y arrastrando con el botón 1 se copia a un portapapeles y con el 2 se *pega*, en la misma aplicación o en otra.

Ejecución de Aplicaciones en X-Window (II)

- Paro 'apropiado' de **clientes**, a veces, mediante opción '*close*' del menú local, o de la opción que tengan para ello.
- Paro 'apropiado' de **no clientes**, de la opción que tengan para ello.
- Otros modos de 'paro': **Control+c**, **Control+d**.
- Paso a 2º plano: **Control+z**, y uso de los comandos: `jobs ([1]+ Running xterm &)`, `bg%num-trabajo`, `fg%num-trabajo`.
- Una mayoría de aplicaciones de **X11** permiten la copia de información con el ratón. Pinchando y arrastrando con el botón 1 se copia a un portapapeles y con el 2 se *pega*, en la misma aplicación o en otra.

Ejecución de Aplicaciones en X-Window (II)

- Paro 'apropiado' de **clientes**, a veces, mediante opción '*close*' del menú local, o de la opción que tengan para ello.
- Paro 'apropiado' de **no clientes**, de la opción que tengan para ello.
- Otros modos de 'paro': **Control+c**, **Control+d**.
- Paso a 2º plano: **Control+z**, y uso de los comandos: `jobs ([1]+ Running xterm &)`, `bg%num-trabajo`, `fg%num-trabajo`.
- Una mayoría de aplicaciones de **X11** permiten la copia de información con el ratón. Pinchando y arrastrando con el botón 1 se copia a un portapapeles y con el 2 se *pega*, en la misma aplicación o en otra.

Ejecución de Aplicaciones en X-Window (II)

- Paro 'apropiado' de **clientes**, a veces, mediante opción '*close*' del menú local, o de la opción que tengan para ello.
- Paro 'apropiado' de **no clientes**, de la opción que tengan para ello.
- Otros modos de 'paro': **Control+c**, **Control+d**.
- Paso a 2º plano: **Control+z**, y uso de los comandos: `jobs ([1]+Running xterm &)`, `bg%num-trabajo`, `fg%num-trabajo`.
- Una mayoría de aplicaciones de **X11** permiten la copia de información con el ratón. Pinchando y arrastrando con el botón 1 se copia a un portapapeles y con el 2 se *pega*, en la misma aplicación o en otra.

Ejecución de Aplicaciones en X-Window (II)

- Paro 'apropiado' de **clientes**, a veces, mediante opción '*close*' del menú local, o de la opción que tengan para ello.
- Paro 'apropiado' de **no clientes**, de la opción que tengan para ello.
- Otros modos de 'paro': **Control+c**, **Control+d**.
- Paso a 2º plano: **Control+z**, y uso de los comandos: `jobs ([1]+ Running xterm &)`, `bg%num-trabajo`, `fg%num-trabajo`.
- Una mayoría de aplicaciones de **X11** permiten la copia de información con el ratón. Pinchando y arrastrando con el botón 1 se copia a un portapapeles y con el 2 se *pega*, en la misma aplicación o en otra.