

# Tema II

## Introducción a X-Lib (R-1.0)

Programación en Entornos Interactivos.

14 de marzo de 2011

Dpto. Lenguajes y Sistemas Informáticos  
Universidad de Alicante



## Resumen

Presentación de X-Lib. Errores de ejecución en XLib. Ficheros de cabecera estándar en X-Lib. Valores y tipos genéricos. Convenios en los nombres y argumentos en XLib. Consideraciones a la hora de programar con Xlib. Juegos de caracteres y codificaciones. Distintas categorías de funciones en Xlib.



# Presentación de X-Lib (I)

- **Xlib** es una biblioteca de subrutinas en **C** que es utilizada por un 'cliente' como el interfaz con *X-Window System*, también descrito como:

*over-sized, over-featured, over-engineered and incredibly over-complicated. X11R6 released in May 1994.*

- Manual de referencia: "*Xlib - C Language X Interface*".
- Cada *toolkit* viene documentado en su propio manual.
- El comportamiento que presente **X** está definido en "*The X Window System Protocol*".

## Presentación de X-Lib (II)

- Casi todas las funciones de Xlib se ejecutan de manera asíncrona, son peticiones enviadas a un buffer de salida. Para trabajar de modo síncrono se ha de hacer uso de **XSync** tras cada función de Xlib que queramos ejecutar de modo síncrono.
- Muchas funciones de Xlib devuelven un valor entero: '*resource ID*', el cual se utiliza para referirnos a objetos que hay almacenados en el servidor X: *Window*, *Font*, *Pixmap*, *Colormap*, *Cursor* y *GContext*. Estos tipos están definidos en `<X11/X.h>`.

# Errores de ejecución en XLib

- Algunas funciones en Xlib devuelven un resultado entero que debe interpretarse como una indicación de error ( $0 = \text{error}$ ).
- Por defecto los errores son tratados por una función estándar de la biblioteca, la cual imprime el error producido, pero el programador puede proporcionar una propia: *error handler*.
- Debido al funcionamiento asíncrono entre *cliente* – “*servidor X*”, los errores pueden ser detectados mucho después de ocurrir, sin embargo se puede hacer que Xlib sea completamente síncrono para depurar programas.

## Ficheros de cabecera estándar en X-Lib (I)

- **<X11/Xlib.h>**: Cabecera principal. Contiene la declaración de la mayoría de los símbolos.
- **<X11/X.h>**: Declara constantes y tipos usados en el protocolo **X**. Se incluye en **<X11/Xlib.h>**.
- **<X11/Xcms.h>**: Contiene símbolos para trabajo con colores. Antes se debe incluir **<X11/Xlib.h>**.
- **<X11/Xutil.h>**: Declara funciones, tipos y símbolos para comunicación entre clientes. Antes se debe incluir **<X11/Xlib.h>**.

## Ficheros de cabecera estándar en X-Lib (II)

- **<X11/Xresource.h>**: Declara funciones, tipos y símbolos para trabajo con recursos. Antes se debe incluir **<X11/Xlib.h>**.
- **<X11/Xatom.h>**: Declara los átomos predefinidos, los símbolos que comienzan con **"XA\_"**.
- **<X11/cursorfont.h>**: Declara los cursores estándar del ratón. Todos comienzan con **"XC\_"**.

## Ficheros de cabecera estándar en X-Lib (III)

- **<X11/keysymdef.h>**: Declara los valores estándar *KeySym*, todos comienzan por “**XK\_**”.
- **<X11/keysym.h>**: Define algunos símbolos del preprocesador, y luego incluye a **<X11/keysymdef.h>**.
- **<X11/Xlibint.h>**: Declara más funciones, tipos y símbolos. Se incluye en **<X11/Xlib.h>**.



## Ficheros de cabecera estándar en X-Lib (IV)

- **<X11/Xproto.h>**: Declara más funciones, tipos y símbolos para el protocolo básico de **X**. Se incluye en **<X11/Xlibint.h>**.
- **<X11/Xprotostr.h>**: Declara más funciones, tipos y símbolos para el protocolo básico de **X**. Se incluye en **<X11/Xproto.h>**.
- **<X11/X10.h>**: Declara funciones, tipos y símbolos para compatibilidad con la versión **X10**.

# Valores y tipos genéricos

Xlib define los siguientes símbolos:

- El tipo **Bool** y los valores **True** y **False**.
- El identificador **None**, que representa el '*resource ID*' o átomo nulos.
- El tipo **XID** usado para '*resource IDs*' genéricos.
- El tipo **XPointer**, definido como '**char \***'. Se utiliza como un puntero *opaco* a la información.

# Convenios en los nombres y argumentos en XLib (I)

Xlib sigue una serie de convenios para dar nombre y representar la sintaxis de sus funciones. Los más importantes son:

- Los símbolos *externos* de **X** usan una mezcla de mayúsculas y minúsculas. Las *variables* siempre usan *minúsculas* y las *macros* sólo *mayúsculas*.
- Todas las funciones de Xlib comienzan con una **X** mayúscula.
- Todas las palabras que forman parte del nombre una función o un símbolo, siempre comienzan por una letra mayúscula.

## Convenios en los nombres y argumentos en XLib (II)

- Todas las estructuras de datos visibles para el usuario comienzan por una **X** mayúscula. Cualquier *cosa* a la que se pueda referir un usuario, siempre empieza por una **X** mayúscula.
- Las macros y otros símbolos no empiezan por una letra **X** mayúscula. Cada palabra de la macro comienza por una letra mayúscula.
- Una variable o un elemento de una estructura siempre se escribe en minúsculas. Los nombres compuestos de éstas se separan por un símbolo '\_'.

## Convenios en los nombres y argumentos en XLib (III)

- El argumento que representa el '*display*', cuando se utiliza, siempre es el primero de la lista de argumentos.
- Los argumentos que sean '*resource IDs*' van a continuación del '*display*'.
- Los argumentos de tipo '*graphic context*' siempre van detrás de los que sean de otro tipo de recurso.
- Los argumentos de *entrada* van delante de los de *salida*.
- El argumento **x** siempre precede al argumento **y**, así como el argumento **width** precede al **height**, y siempre que aparezcan los cuatro, el **x** y el **y** van delante de los otros dos.

## Consideraciones a la hora de programar con Xlib

- Las coordenadas y tamaños en X son cantidades de 16-bits.
- Los teclados son muy diferentes de un fabricante a otro. Hay que llevar un especial cuidado en este aspecto a la hora de escribir un programa portable.
- Muchos '*displays*' tienen un tamaño pequeño de memoria, en ellos se debe evitar el uso de *pixmap*s y el atributo *backing store*.
- Las aplicaciones no deberían realizar tareas específicas del manejador de ventanas.

# Juegos de caracteres y codificaciones (I)

Algunas funciones de Xlib hacen referencia a unos juegos de caracteres específicos, estos son:

- **X Portable Character Set:** 97 caracteres más el carácter **espacio**, **tab** y **newline**.
- **Host Portable Character Encoding:** Es la codificación del juego de caracteres anterior, la cual, no está definida, pero debe ser la misma en todas las *versiones locales* de Xlib en una máquina.
- **Latin-1:** El juego de caracteres definido por el estándar **ISO8859-1**.
- **STRING encoding:** Es el **Latin-1** junto con **tab** y **newline**.
- **POSIX Portable Filename Character Set:** Conjunto de 65 caracteres empleados para dar nombre a ficheros en un computador conforme a POSIX. Los caracteres son: **a..z A..Z 0..9 .\_-**

¿Qué es POSIX?

*Portable Operating System Interface for computer environments. A set of IEEE standards designed to provide application portability. IEEE1003.1 defines a Unix-like operating system interface, 1003.2 the shell and utilities, and 1003.4 real-time extensions.*



# Distintas categorías de funciones en Xlib

Una división por categorías de las funciones de Xlib puede ser ésta:

<b>Display</b> (Conexión con el X-server)	<b>Window</b> (Creación y manejo de ventanas)
<b>Window Information</b> (Leer/modificar propiedades, selecciones)	<b>Graphics Resource</b> (Trabajo con colores, pixmaps y GC)
<b>Graphics</b>	<b>Window Manager</b> (Funciones específicas del WM)
<b>Event Handling</b> (Comunicación del X-server con los clientes)	<b>Property</b> (Propiedades de una ventana)
<b>Utility</b>	