X Directions

Keith Packard

X Window System Architect

keithp@keithp.com

2005–12–01

X Directions

- Dynamically adapt to environment
- Improve software distribution mechanism
- Leverage modern graphics hardware
- Shed legacy code

X.Org
FOUNDATION

Dynamic Environment

- "Everything can change"
- Hot-plug monitors
- Screen resize, rotate
- Hot-plug input
- Modify Xinerama layout

Modular Distribution

- Why
    - Exit build-system business
    - Decouple core/driver release schedules
    - Reduce security update impact
    - Enable forking of drivers, libraries, etc.
- How
    - Autotools
    - pkg-config
    - Dual 6.9/7.0 release
- Scheduled to ship Dec. 21, 2005

X on 3D graphics hardware

- Why
  - “3D” hardware occupies 90% of chip area
  - 2D performance no longer matters to vendors
  - Render extension maps to GL/Direct-X nicely
- EXA acceleration architecture
  - Traditional X architecture
  - Incremental driver change
  - Focus on video memory management
  - Accelerates Render extension
- X as GL application
  - Reduces priviledged code base
  - Reduces driver development effort
  - Allows interesting 3D immersive environments

Composited Window System

- 2D application as “paint”
- Arbitrary rendering of application results
- Input redirection still in progress

Cleaning up Messes

- Xlib internationalization support
  - Pre-Unicode model
  - No standard input method interface
  - Dependent on core font support
  - Implemented at wrong level (xlib instead of toolkit)
- Keyboard support
  - XKB is too complex
  - Limitations (4 layouts)
  - Poorly integrated with input method mechanisms
- Color management
  - Pre-sRGB model
  - Pixel-value centric
  - Again, wrong level – should be above xlib

Migration to XCB

- XCB is tiny (100K vs 1M)
- Proven correct threading model
- Expose all protocol capabilities
- Just protocol, no additional functionality

Kernel Graphics Drivers

- Current situation
  - fbdev
  - DRM
  - X
- Resources
  - PCI address space
  - Interrupt routing
  - Memory management
  - GPU scheduling
- Three drivers for one device
- Need one central resource manager
- Must include kernel piece

Utopian Dream for Graphics

- Kernel driver for hw management
- X is just a DRI application
- Standard video mode notification API
- Extensible video mode configuration API
- X is unprivileged
- DRI surfaces exposed to other DRI applications
- Cross-application memory management

Wrap-up

- Dynamic adaptation
  - Screens, input, layout
- Improve release process
  - Modularization
- Modern graphics hardware
  - EXA, GL-based X
- Escape our past
  - Xlib → XCB
  - Replace(?) XKB
  - Eliminate XCMS
  - Move i18n above Xlib, standardize im interface
  - Build real kernel driver infrastructure

X.Org
FOUNDATION