XCB and Xlib

http://xcb.freedesktop.org/

Jamey Sharp
Computer Science Department
Portland State University

Overview

- Why XCB?
  - Xlib issues
  - XCB solutions
- XCB status
  - Xlib integration status
- Roadmap
  - Porting libraries and applications
- To-do

Why XCB?

- No magic!
  - Simple descriptions → direct bindings
  - Close correspondence to X protocol
- Threaded applications
- Embedded applications

Xlib issues

- Implementation is
    - Large
    - Complex
    - Inflexible
- Unpredictable requests
- Threading
- Minimal compile-time checking

XCB solutions

- Code generation
  - Direct mapping between protocol and C API
- Narrow interfaces
- Minimal caching and prefetching
- Reply cookies
- Careful use of C's type system

XCB status

- Today: Fully functional implementation, API largely stable
- ~20 functions in public API, plus protocol bindings for 17 extensions (and counting)
- Small API extensions planned to help with common cases

XCB performance and correctness

- Some optimizations not yet implemented
- Minimal performance testing done so far
  - No testing since last performance fixes
  - Should be the same (at worst) modulo bugs
- Implementation believed correct:
  - Test suite currently under development
  - Formal methods validate threading support

Xlib integration status

- Initial prototype easy
  - Most apps worked after two days of coding
- Xlib API semantics are hard!
  - Understanding Xlib transport is especially hard
  - X Test Suite says still bugs here
- Need to pass or fix X Test Suite
  - XPROTO tests will never pass as written: they completely violate Xlib's API
- Stress testing under real workloads is ongoing

Xlib roadmap

- Status quo: Apps use Xlib API
- Incremental library and application porting
  - Must support legacy apps without maintaining dual parallel libraries
- Goal: XCB only; Apps don't use Xlib

Porting libraries and applications

1. Use new Xlib/XCB interface to port individual functions
2. Convert internal interfaces and structures from Xlib types to XCB
3. Libs
   a) Convert public interfaces to XCB
   b) Provide thin wrapper library offering old interface to let old applications use the newly ported library

XCB roadmap

- Execute Xlib roadmap
- Auto-generate documentation
- XCB-specific test suite
- Utility APIs

To-do: XCB test suite

- Work in progress by student in Open Source Software class at PSU
- General strategy: rewrite and augment XTS5 to test XCB

To-do: Utilities

- Image buffer manipulation (XImage)
- Core keyboard, XKB, and input methods
- Caches, as needed
  - Atoms
  - Graphics contexts

Acknowledgments

- Thanks to Keith Packard and Bart Massey for much of XCB's design and inspiration
- Thanks to the Computer Science Department at Portland State University for their continuing support, allowing me to attend this conference

Availability

- MIT/X licensed
- http://xcb.freedesktop.org