Ladybird
Independent web browser
Introduction

Andreas Kling
@awesomekling

My browser history

- 2006: KHTML (KDE)
- 2009-2011: QtWebKit (Nokia)
- 2011-2017: Safari (Apple)
- 2019-Present: Ladybird
Ladybird: Overview

- Full-stack browser & engine
- Everything from scratch
- Permissively licensed (2BSD)
- Written in modern C++20
- Part of the greater SerenityOS project
- Also runs on Linux, macOS, other *nix, but not Windows (yet)
Ladybird project goals

- Render the live web with acceptable stability, security and performance.
- Advance the web platform by filing spec bugs, writing tests, etc.
- Add engine diversity to the browser market.
- Have fun!
Simple HTML widget for rich text display

Start of JavaScript engine development

Cross-platform version with a Qt GUI

Jun 2019

Oct 2019

Mar 2020

Jun 2020

Jul 2022

Browser application for SerenityOS

Multi-process mode
State of Ladybird
Acid tests almost perfect
Making progress on modern sites
But many bugs remain :^)
Ladybird architecture
Ladybird application architecture

SerenityOS app

Qt app

Process boundary

WebContent

LibWeb
LibJS
LibRegex

LibCore
LibGfx
LibWasm

LibXML
LibUnicode
LibWebSocket
LibJS

- Implements the ECMAScript language & runtime
- Garbage collector
- Simple AST tree-walk interpreter
- Almost entirely unoptimized
LibWeb

- Implements most of the web platform specs:
  - DOM, HTML, CSS, SVG, Fetch, XHR, etc.
- Layout
- Painting
- Hit testing
Other libraries

- LibCore: Event loops, shared memory
- LibIPC: Inter-process communication
- LibGfx: 2D graphics, OpenType rasterizer
- LibRegex: Regular expression bytecode VM
- LibUnicode: Unicode support
- LibXML: XML parsing
- LibWasm: Wasm interpreter
- ... and others
LibWeb DOM node class hierarchy

- JS::Cell
- JS::Object
- Web::Bindings::PlatformObject
- Web::DOM::EventTarget
- Web::DOM::Node
LibWeb rendering pipeline

HTML DOM

CSSOM

Per-Element Computed Style

Layout Tree

Paint Tree
Layout implementation

- Organized around CSS spec concepts:
  - FormattingContext (with subclasses for BFC, FFC, GFC, …)
  - AvailableSpace

- Two layout modes:
  - Normal
  - Intrinsic sizing
Ladybird testing

- **LibJS:**
  - test-js: In-house regression test suite
  - libjs-test262: test262 runner

- **LibWasm:**
  - test-wasm: Wasm spec tests

- **LibWeb:**
  - Layout tests
  - Text tests
  - WPT runner under construction (actively)
Upcoming work

- Performance
- Memory safety
- Implement specs
- Fix bugs
<table>
<thead>
<tr>
<th>Name</th>
<th>Name</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adam Hodgen</td>
<td>Gunnar Beutner</td>
<td>Matthew Olsson</td>
</tr>
<tr>
<td>Aliaksandr Kalenik</td>
<td>Idan Horowitz</td>
<td>Max Wipfli</td>
</tr>
<tr>
<td>Ali Mohammad Pur</td>
<td>Igor Pissolati</td>
<td>Moustafa Raafat</td>
</tr>
<tr>
<td>Andi Gallo</td>
<td>Jamie Mansfield</td>
<td>networkException</td>
</tr>
<tr>
<td>Andreas Kling</td>
<td>Jelle Raaijmakers</td>
<td>Obinna Ikeh</td>
</tr>
<tr>
<td>Andrew Kaster</td>
<td>Jonah</td>
<td>Sam Atkins</td>
</tr>
<tr>
<td>Ben Wiederhake</td>
<td>Karol Kosek</td>
<td>Simon Wanner</td>
</tr>
<tr>
<td>Brian Gianforcaro</td>
<td>Kenneth Myhrai</td>
<td>sin-ack</td>
</tr>
<tr>
<td>Daniel Bertalan</td>
<td>Leon Albrecht</td>
<td>Srikavin Ramkumar</td>
</tr>
<tr>
<td>David Tuin</td>
<td>Linus Groh</td>
<td>stelar7</td>
</tr>
<tr>
<td>DexesTTP</td>
<td>Luke Wilde</td>
<td>Tim Schumacher</td>
</tr>
<tr>
<td>Egor Ananyin</td>
<td>Martin Falisse</td>
<td>Tobias Christiansen</td>
</tr>
</tbody>
</table>
Thank you!

Q&A
Ask me anything!