OldWeb.today
Keeping Retro Web Sites Available on the Modern Web (including Flash!)

Ilya Kreymer, Webrecorder Lead Developer
The Webrecorder project is focused on advancing open source software development and research in the following key areas:

- FOSS web archiving tools to create and view web archives
- Highest-fidelity capture and replay
- Integrate with existing archival systems
- Exploring intersection of web archiving and software emulation
- Empower anyone to create, use and share web archives
- Making web archiving more accessible via decentralized and p2p technologies
About OldWeb.Today

- Browse old web sites with old browsers!
- Original version launched in 2015 with emulators running in Docker containers still available (classic.oldweb.today) but requires infrastructure and maintenance (eg. was built using a now-obsolete version of Docker)
- Latest OldWeb.today created in early 2021
- Client-side emulation - all emulators run in the browser itself
- Should improve as WebAssembly and JS emulators improve
What is web archiving?

“What is web archiving is the process of collecting portions of the World Wide Web, preserving the collections in an archival format, and then serving the archives for access and use.” International Internet Preservation Consortium
Most well known:

[Internet Archive logo]

[Wayback Machine logo]
Public Web Archives

Most well known:

But also many others out there:

Many institutions run their own wayback machines…
Many web archives make the original web content (without banner, any changes available via) a de-facto convention, using the “id_” modifier: https://<archive-path-prefix>/<timestamp>id_/<url>

Examples:

Given a web archive that supports this, possible to load archived web pages dynamically given a URL and Timestamp.

(Note: Still need a Cross-Origin-Resource-Sharing (CORS) proxy - as most web archives do not support CORS)
Emulators used in Oldweb.today

- **v86** - popular x86 Emulator written in JS, and WebAssembly (Used for Window and Linux Environments)
  [https://copy.sh/v86/](https://copy.sh/v86/)

- **Basilisk II** - 68k Mac Emulation, then ported to JS and WebAssembly (Used for MacOS Environments)

- **Ruffle** - Flash-only emulator written in Rust, running WebAssembly
  [https://ruffle.rs/](https://ruffle.rs/)
Each environment was prepared in v86 or Basilisk II.

For Basilisk II, disk images created using desktop version.

For v86, disk images exported via web version after installing browser. (also restoring emulator state for faster loading)

Supported Environments

- NCSA Mosaic 2
- NCSA Mosaic 3
- MacLynx 2
- Navigator 3
  - SHOCKWAVE 3
  - JAVA 1.0
- Navigator 4
  - SHOCKWAVE 3
- IE 4
  - SHOCKWAVE 3
  - JAVA 1.0
- Navigator 4
- IE 5
  - FLASH 8
  - JAVA 1.1.5
- IE 6
  - FLASH 9
  - JAVA 5.0
- Firefox 10
  - FLASH 32
- Opera 12
  - FLASH 32
- Ruffle (Flash)
  - FLASH
JS/WebAssembly Emulators

Great Support for:

- Video
- Audio
- Networking
Great Support for:

- Video
- Audio
- Networking

Need a JS/WASM based TCP/IP stack to support networking!
How to connect emulators to web archives?

- Web Archives need to be accessed via HTTP, over TCP/IP
- V86 supports sending Ethernet frames over a WebSocket
- Basilisk II port in JS had no networking support, but Basilisk II does, and can get Ethernet frames.
- Want to avoid relying on external infrastructure as much as possible.

Ethernet Packet — ??? — HTTP Web Archive
How to connect emulators to web archives?
TCP Stack in the Browser

- **Solution:** Extending on excellent work from eaas-proxy: https://gitlab.com/emulation-as-a-service/eaas-proxy/-/blob/master/webnetwork.js
- PicoTCP compiled via Emscripten to WASM + wrappers for sockets over native JS web streams (full TCP stack)
- Can terminate the HTTP connection from emulator in the users’ browser!
**Custom Networking Setup**

**Emulator Web Worker**: Run emulator WASM (v86 or Basilisk)

**PicoTCP Worker**: Run PicoTCP Stack

Emulator Worker sends Ethernet frames to PicoTCP web worker over a broadcast channel, which processes them as TCP packet, and passes them to HTTP Server, which then proxies to the live web / web archive… and then sends the response back!
How to dynamically set the URL and Timestamp in the emulated environment?

Solution: Browser in emulator uses an HTTP proxy pointing to, 10.0.2.2:6082, redirects to the actual URL.

When timestamp is changed, user is asked to reload to get new home page.
What about Flash (and Java?)

✅ Prepared environments include browsers with various versions of Flash and Java installed!

✅ Many Flash projects do work!

✅ Can support latest Flash (32)

❓ But only older browsers (upto Firefox 10 / Opera 12)

❓ Many Flash/Java not fully archived in the 90s / early 00s (.swf, .jar files missing)

❓ Perf: Loading can be slow for Flash (and Java)
Ruffle Emulator

✅ Also supported in oldweb.today!

✅ Web page loaded directly in native browser, emulator only loaded for Flash content

✅ Much faster than a full-OS emulation (no need for custom networking)

❓ Not all of Flash is supported, yet (no AS3)

❓ Re-implementation of Flash, may be subtle differences.
## Other Approaches to Supporting Flash in Web Archives

<table>
<thead>
<tr>
<th></th>
<th>Supports All Flash (include AS3)?</th>
<th>Runs at full fidelity? (Video + Audio + Speed)</th>
<th>No Infrastructure CPU Cost Per User</th>
<th>Available in Tools:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ruffle</td>
<td>Not Yet</td>
<td>Yes</td>
<td>Yes</td>
<td>oldweb.today, archiveweb.page, replayweb.page</td>
</tr>
<tr>
<td>OldWeb.today JS Emulators</td>
<td>Yes</td>
<td>Not Yet</td>
<td>Yes</td>
<td>oldweb.today</td>
</tr>
<tr>
<td>Remote Browsers</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>pywb-remote-browsers, conifer, classic.oldweb.today</td>
</tr>
</tbody>
</table>
Blog Post about oldweb.today: https://webrecorder.net/2020/12/23/new-oldweb-today.html

Main Repo: https://github.com/oldweb-today/oldweb-today

Basilisk II Fork: https://github.com/oldweb-today/macemu

V86 Fork: https://github.com/oldweb-today/v86

EaaS JS Network Based + PicoTCP Emscripten Build: https://gitlab.com/emulation-as-a-service/eaas-proxy

Emulators running server-side in Docker containers: https://github.com/webrecorder/pywb-remote-browsers

ArchiveWeb.page Extension - allows archiving Flash via Ruffle emulator in the browser https://archiveweb.page (injects Ruffle)
Future Work for oldweb.today

- Optimization!
- Adding more emulation environments to support newer browsers + newer OSs
- Make networking more robust / debug occasional errors
- Rewrite https URLs for older browsers
- Support for adding different besides Internet Archive
- Ability to upload an archive directly + create archive while browsing (integration with other Webrecorder tools)