Medley Interlisp

2023 Project 18 March 2023

Interlisp.org

What have we been doing?

+ Adapting the system to current environments and standards

hardware: keyboard, mouse, display, CPU (64bit, little endian)

software: standard C, Unicode, Posix, UI Guidelines

+ Lower barriers to entry

+ Demonstrate unique and original features

+ Help with revival of other applications built using system

+ Improve maintainability for future users

+ Gather and update documentation

Who might use Medley Interlisp?

- Retrocomputing enthusiasts

- Researchers, IDE tool creators, looking into ideas

- Software and AI historians (present and future)

- Software archivists (SPN, technical infrastructure)

- Developers of new applications

Who is involved?

+ Some of original developers from Xerox PARC (in 70's)

+ Developers of s using Interlisp (recursive revivers)

- Open Source contributors

- Students of computer science and history

+ Friends, enthusiasts, and ... you?

```
(OPEN-URL "https://github.com/orgs/interlisp/people")
(MEDLEY-CONTRIB "medley")
(MEDLEY-CONTRIB "maiko")
(MEDLEY-CONRIB "online")
(MEDLEY-CONTRIB "Interlisp.github.io")
```

More on Modernizing Medley

Goals

- Adapt Medley UI to current hardware

- Make Medley work more like current applications for comfort

Mouse: Wheel Scroll, Window move on title, Window resize on corner. Warning: Exec windows. Consolation: you can do it all with menus. Reduction: some menus are invoked by "middle" mouse button which on some mice is hard to press.

Keyboard: Desired state: compatible keystrokes. Subtasks: decide what keystrokes do what; slash through umpteen layers of keyboard re-iterpretation.

Display: Color, high-resolution displays, modern fonts.

Reduce Barriers to Entry

- Running Online

```
OPEN-URL("https://www.youtube.com/watch?v=mI3Ga5LyIlI")

OPEN-URL("https://online.interlisp.org")
```

- Installation Instructions

```
OPEN-URL("https://interlisp.org/running/")
```

More notes on Running on Modern Systems

- about K&R C and C standards; type declarations, big-endian vs. little endian, 32 bit vs 64 eliminating errors, file-sytem changes, process handles, bugs left over from 24- to 28-bit address space

- docker, installers, CO/CI(?), two levels of virtualization

- online.interlisp.org: connect in seconds.

- Mysteries in the code we inherited, "software archeology"

- Robust "loadup process", all day vs seconds

- bytecode virtual machine aids portability.

```
fb {li}<maiko>bin/makefile-*.*-x
```

Linux, Mac, Windows (WSL1, WSL2, Cygwin)

Docker, FreeBSD, OpenBSD, SunOS5
x86_64, i386, arm7l, arm64, and older

Getting Involved

- Try things out, report new problems, or new reproducible cases

- Clean up old issues

```
(OPEN-URL "https://github.com/interlisp/medley/issues")
```

```
(FILESLOAD PICK)
```

```
PICK ISSUE
```

- Network emulation

-Donate? See our GitHub sponsor page