

# RetroBSD and LiteBSD: Meet the Smallest BSDs

Brian Callahan <[bcallah@devio.us](mailto:bcallah@devio.us)>

New York City \*BSD User Group July 2016 meeting

# First thing's first...

- Interrupt me if you have questions.

# About me

- George continues to think me giving talks at NYC\*BUG is a good idea...
- My OpenBSD developer account is locked currently...
- Contributing to RetroBSD since 2013
- One of two or three people who have commit rights to LiteBSD
- Soon to be ABD in the Science & Technology Studies Ph.D. program at RPI
  - I teach with \*BSD
- Admin at Devio.us, a large OpenBSD free shell provider
- One-time speaker at AsiaBSDCon
- Two-time BSD Now and BSD Talk interviewee
- Award winning \*BSD conference attendee
- I've been around the \*BSD scene a while now...

# A Presentation in Four Parts

- Brief overview of the projects and community, getting set up
- RetroBSD: an OS from the past in 128k RAM
- LiteBSD: a micro-reimagining of today's BSDs
- Live demonstration and (not so mini?) installfest

# RetroBSD and LiteBSD: an overview

- Begun by Serge Vakulenko around 2009.
- Targets embedded MIPS CPUs in the PIC32 series.
  - RetroBSD: PIC32MX
  - LiteBSD: PIC32MZ
- Targets specific hardware.
  - <http://retrobsd.org/wiki/doku.php/board/index>

# Try before you buy

- RetroBSD tree comes with a virtualmips simulator.
  - Building RetroBSD builds the simulator, so it's just two commands from zero to running RetroBSD.
- LiteBSD can be run in QEMU.

# The community

- All code is on GitHub.
- The community is centered around a forum and a wiki.
- OpenBSD-style marketing team
- Few talks, little general information
  - <https://www.youtube.com/watch?v=w4tmvfkRpyU> – BSD Day 2011
  - <https://www.youtube.com/watch?v=TSz4yn63YNw> – 3.5 minutes of BSD Now coverage
  - I spoke for 5 minutes about it at AsiaBSDCon 2016...

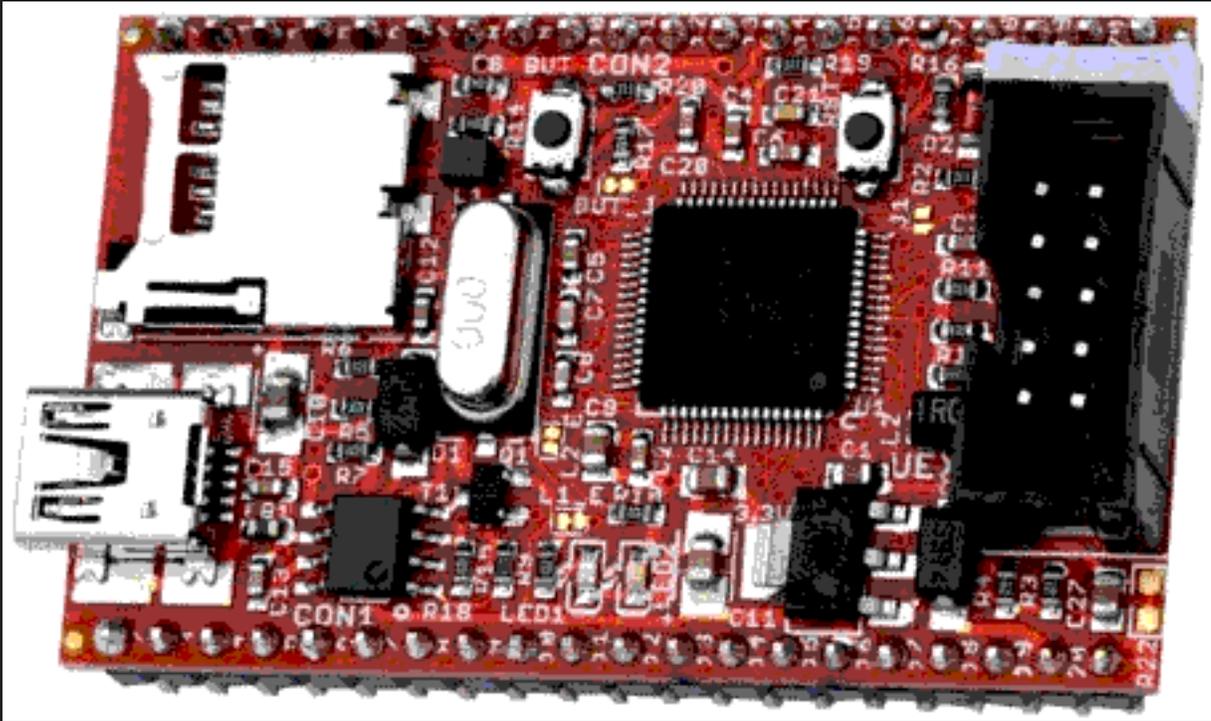
# The community

- We try to make it as easy as possible to get involved.
- Don't want to compile anything? No problem—we autobuild after every commit.
- Want to develop? We have toolchains available for FreeBSD, Linux, and Mac OS X.
- We take just about any code you want to give us.

# RetroBSD: an OS from the Past in 128k RAM

- Main features:
  - Small resource requirements. RetroBSD requires only 128 kbytes of RAM to be up and running user applications.
  - Memory protection. Kernel memory is fully protected from user application using hardware mechanisms.
  - Open functionality. Usually, user application is fixed in Flash memory - but in case of RetroBSD, any number of applications could be placed into SD card, and run as required.
  - Real multitasking. Standard POSIX API is implemented (fork, exec, wait4 etc).
  - Development system on-board. It is possible to have C compiler in the system, and to recompile the user application (or the whole operating system) when needed.

# Hardware



# One-stop shopping

- All the code that has been tested on RetroBSD lives in the RetroBSD tree.

# OK, but what do you get?

- Complete \*BSD userland
  - sh(1), csh(1), text editors, games, and more
- Basic networking capabilities (via a userland library)
- File transfer via X/Y/ZMODEM

# What can you do in 128k RAM anyway?

- First, you need 32k for the kernel.
- So really, what can you do with 96k of RAM?
  - (Audience: take guesses!)
- Oh, and you have no MMU, no virtual memory, and no paging

# What can you do in 96k RAM anyway?

- Edit text!
  - Six different text editors from ed to microemacs
- Play games!
- Learn computer programming!
  - Assembler
  - BASIC interpreter
  - 5 C compilers
  - 3 Forth interpreters
  - Linker
  - Make
  - Scheme interpreter
  - Shell scripting
  - Tcl shell
- Access GPIO pins
- Explore \*BSD history
- More programs are added as fast as the community can write them.

# Future work...

- More applications?
- More hardware?
- More users and developers!
- Bug fixing (as always)

# LiteBSD: a mini modern \*BSD

- Much newer project, started in 2014.
- Targets the newer PIC32MZ chips
  - 512k RAM!
- Began with the PMAX port of 4.4BSD-Lite2

# What's in the box?

- Another complete \*BSD userland
- Utilities you already use!
  - Sqlite3, ftp(1), vi(1), mg(1), gzip/gunzip, netcat
- Lots of useful libraries
  - Ncurses 6.0, libarchive, zlib, openssl (libressl not an option yet...)
- Lua (but not in the kernel...)
- Real networking
- File transfer via Kermit
- Package utility

# A modern \*BSD in 512k

- Most of the userland is from the current BSDs
- Examples:
  - csh(1), dc(1), ksh(1), mg(1), nc(1), pax(1), rm(1), sed(1), sort(1), spell(1) from OpenBSD
  - awk(1), cron(1), nl(1) from NetBSD
  - bc(1), libarchive from FreeBSD
  - Library functions from everywhere

# Adventures in making a modern \*BSD

- What's missing from 4.4BSD-Lite2 that we might take for granted these days?

# Future work...

- My wishlist:
  - Working toolchain
  - /dev/random
  - More users and developers!
  - Bug fixing
  - ssh?
  - tmux?

# Ports... you knew it had to happen

- LiteBSD has a ports tree
  - Approximately 90 ports
- Have you used a ports tree before? Then you know how to use this one.
  - I used my experience with the OpenBSD ports tree to create the LiteBSD ports tree.
- `opkg(1)` utility for installation/removal/etc.

# Ports adventures

- Building ports is simple:
  - Build LiteBSD
  - Checkout ports tree
  - ``cd`` to desired port
  - ``make package`` (or ``bmake package`` if on Linux or Mac OS X)
- Or you can download our pre-built packages

# Notable ports

- gmake
- ee editor (from FreeBSD)
- nano editor (for Allan Jude)
- A variety of games
- A variety of languages (BASIC, Bf, FORTRAN, Forth, Pascal, TinyScheme)
- irc clients (ii, naim, sic)
- file(1)
- less(1)
- Links+ browser
- Several small web servers (thttpd)

# Future hopes

- Be able to use gmake/other makes to build ports
- More than just C and scripting languages
- A BSD-licensed reimplementation of the `opkg(1)` utility
- More ports!
  - And more people making ports...

# Live demo time!

- Ask questions if you have any.
- This is also the time I'll be available for help installing on your machines.
- Thanks for listening!

# Thanks and shout-outs

- Serge, for helping the project and his great skill and infinite patience
- The RetroBSD community, none of this happens without all of you
- Microchip Technologies, for sending me a free WiFIRE board
- Olimex Ltd., for sending me a free EMZ64 and PIC32-RETROBSD
- And you, for listening

# If you want to learn more...

- RetroBSD/LiteBSD:
  - Website & Forums: <http://retrobsd.org/>
  - GitHub:
    - <https://github.com/RetroBSD/retrobsd>
    - <https://github.com/sergev/LiteBSD>
    - <https://github.com/ibara/LiteBSD-Ports>
- Me:
  - Email: [bcallah@devio.us](mailto:bcallah@devio.us)
  - Twitter: @\_\_briancallahan