Goals

- Share FreeBSD’s long history
- What is FreeBSD and Why People Use It
- Why you should use and/or contribute to FreeBSD
- FreeBSD Feature Highlights
The FreeBSD World

FreeBSD is an open source Unix-like operating system descended from the Unix developed at the University of California, Berkeley in the 1970s.

The FreeBSD Project is an active open source community since 1993 with hundreds of committers and thousands of contributors around the world.

The FreeBSD Foundation is a 501(c)3 non-profit organization registered in Colorado, USA in 2000 dedicated to supporting the FreeBSD Project, its development and its community.
What is FreeBSD?

It’s not a Linux Distribution!
What is FreeBSD?

One of the oldest (1993), largest, and most successful open source projects in the world

Complete operating system including kernel, userland, documentation, and tools

Over 33,000 3rd Party Open Source Packages
What is FreeBSD?

Created and distributed by a community of highly technical and committed contributors (Over 400 active developers and thousands of contributors)

Works on Intel / AMD x86 32 and 64-bit, 32 and 64 bit ARM, RISC-V, PowerPC, MIPS, AWS, Azure, GCP, …

10s of millions of deployed systems
Abridged BSD Family Tree

- AT&T Unix
- BSD
  - FreeBSD
  - NetBSD
A Brief Look Back at the History of FreeBSD

UNIX
In 1969 Ken Thompson, Dennis Ritchie and others started working on a program
Evolution of Unix and Unix-like systems

By Eraserhead1, Infinity0, Sav_vas - Levenez Unix History Diagram, Information on the history of IBM's AIX on ibm.com, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=1801948
Most Likely You Use FreeBSD!

- iPhone or Apple computer
- Streaming Netflix
- Messaging someone over Facebook’s WhatsApp application
- Sony PlayStation 4
- Planning your next vacation
- Getting an awesome deal!
Why Use FreeBSD?

• Friendly and Approachable Community

• Excellent Documentation

• Good Tooling and Modern Compilers

• Consistent Development and Release Processes

• Wide Variety of Architectures Supported

• 2-clause BSD license - Does not restrict what you can do with your own code!

• Secure
How the Project Works

Independent of the FreeBSD Foundation

Developer elected 9-person core team

Mentorship for Commit Bit

One community with different functional teams developing system as a whole (core, release engineering, security, ports, documentation, …)

Collaborative Development Environment
FreeBSD Project Org Chart

FreeBSD Foundation

FreeBSD Project

Core Team

Security Team

Document Team

Cluster Admin

Release Engineering

Ports Management

Other Teams include:
- Ports secteam
- Security Officer
- Bugmeisters
- Ports Security Team
- Continuous Integration Testing Admins
- Postmaster Team
- Webmaster Team
- Phabricator Code Review Administration

Core Team - 9
Committers - ~400
Contributors - Thousands
FreeBSD core team

9-member elected management body
- Elections held every two years
- Active committers vote for core members
- Non-voting core team secretary is selected by the core team

Responsibilities
- Administrative (commit bits, hats, team charters)
- Strategic (project direction, coordination, cajoling)
- Rules, conflict resolution, enforcement

We have no “benevolent” dictators for life!
Who are the FreeBSD committers

Locations
◦ 34 countries
◦ 6 continents

Ages
◦ Oldest (documented) committer born in 1948
◦ Youngest (documented) committer born in 1997
◦ Average age 42
◦ Data from circa June 2019
FreeBSD Releases

POLA: Principle Of Least Astonishment
Don’t break things that work
Upgrades are generally painless
Even across major releases

Two types of releases:

**Major Release**
(Dot Release) – 12.0 - Around every two years (supported for 5 years)

**Point Release** – 11.3 Around every 9 months – ABI/API compatibility

Two types of branches:

**Current** – Head – All changes to base system committed here. Dot releases built from here.

**Stable** – After testing, most changes in current moved here. Point releases built from stable.
Weekly snapshots available for current and stable branches

Supported Releases
- Production: 12.0, 11.3, 11.2
- Upcoming: 12.1
- Support Lifecycle

Download FreeBSD
How to Contribute to FreeBSD

Many ways to get involved with the project, including contributing code, writing documentation, maintaining ports, and advocacy.

The size of the project allows for a greater chance for anyone to make a notable impact.

Easy to get started contributing.

Start by translating or improving our documentation

Pick one of the many ports to maintain or add

Go through the PR list and fix some bugs

deb@freebsdfoundation.org @dgoodkin
Why Companies Use FreeBSD?

• History of innovation
• High performance
• Great tools
• ABI stability within major releases – Remember POLA
• Mature release model
• Excellent documentation
• Business Friendly License
• ZFS
• Open community
• Smaller footprint than most operating systems

“We choose FreeBSD for many of our internal services and product service offerings because we know we can rely on its consistent reliability and performance. Its portability not only allows us to run it on almost any commodity or enterprise server, but allows for the possibility to move a hard drive from one server to another, boot, and get back to normal operation with minimal fuss.”
The Power to Connect

**Application**

Open Connect is the name of the global network that delivers Netflix TV shows and movies to members worldwide.

- The building blocks are purpose-built Open Connect Appliances (OCAs).
- FreeBSD was selected as the operating system for OCA because of its balance of stability and features, strong development community, and staff expertise.

**Results**

- **Delivers over 100 Tb/second at peak**
- FreeBSD is central to pushing this much content cost-effectively. By minimizing kernel to userspace copies, data stays in the kernel as long as possible.
- Async Sendfile, a Netflix and NGINX innovation, is available to all FreeBSD users
  - Web server tells kernel to send this chunk of this file out over this socket
  - Kernel returns to userspace so the web server can do other things
  - Kernel continues in background sending files to users
Where FreeBSD Stands Out

- Embedded Systems
- Video CDN/Streaming
- Security
- Research
- Storage
- Virtualization
- Networking
- High Performance
- Data Centers
- Servers
- ISPs
Kernel features

Multi-processing multi-threaded kernel

Support for many popular hardware architectures: Intel/AMD x86/64, 32- and 64-bit ARM, RISC-V, PowerPC, MIPS

UNIX, POSIX, BSD programming interfaces

Multi-protocol network stack
• IPv4, IPv6, IPX/SPX, AppleTalk, IPSEC, ATM, Bluetooth, IEEE 802.11, SCTP,…
• Reference implementation for many protocols

Unified, coherent build-system across components

Extensive documentation
Userland features

Complete, integrated Unix system
• Expected tools are in the base installation – no extra packages needed
• Build-time knobs to trim the system down for appliances

Kernel and userland maintained together
• Userland is always in sync with the kernel
• New kernel features are immediately available in userland

Strong focus on consistency
Other Features

• **Robust file systems** including UFS and ZFS (Active work happening on ZFS)

• **DTrace** - an advanced event-based performance analysis and troubleshooting tool. DTrace can help you identify and quantify the root cause of virtually any performance issue, in both user-level and kernel code. It can be executed using custom and powerful one-liners and scripts.

• **Jails** – Lightweight virtualization added to FreeBSD in the early 2000s.

• **bhyve** – Full-blown hypervisor. This hypervisor supports a number of guests, including FreeBSD, OpenBSD, Microsoft Windows, and many Linux distributions.

• **TCP/IP** was originally developed on BSD and FreeBSD remains the reference implementation for several network protocols.

• **Capsicum** – Capsicum is a lightweight OS capability and sandbox framework developed at the University of Cambridge Computer Laboratory. Capsicum extends the POSIX API, providing several new OS primitives to support object-capability security on UNIX-like operating systems.
What is CHERI?

- **CHERI** is an **architectural protection model**
  - Composes the capability-system model with hardware and software
  - Adds new security primitives to Instruction-Set Architectures (ISAs)
  - Implemented by microarchitectural extensions to the CPU/SoC
  - Enables new security behavior in software

- CHERI mitigates vulnerabilities in **C/C++ Trusted Computing Bases**
  - Hypervisors, operating systems, language runtimes, browsers, ....
  - Fine-grained memory protection, scalable compartmentalization
  - Directly impedes common exploit-chain tools used by attackers
  - Mitigates many vulnerability classes .. even unknown future classes
The Morello Board

- An Industrial Demonstrator of a Capability architecture
- Uses a prototype capability extension to the Arm Architecture
  - Prototype is a “superset” of what could be adopted into the Arm architecture
- Use of a superset of the architecture is very unusual
  - Also unrealistic as a commercial product – there will be some frequency effects
  - However, there are tight timescales so architecture is nearly complete now
- The superset of the architecture will allow a lot of software experimentation
  - Various different mechanisms for compartmentalisation
  - Collection of features for which the justification is unclear
  - Techniques for holding the capability tag bit
- Architecture will have formally proved security properties (with UoC and UoE)
- Morello Board will be the ONLY physical implementation of this prototype architecture
  - Learnings from these experiments will be adopted into a mainstream extension to the Arm architecture
  - NO COMMITMENT TO FULL BINARY COMPATIBILITY TO THE PROTOTYPE ARCHITECTURE
    - But successful concepts are expected to be carried forward into the architecture and can be reused there
The FreeBSD Foundation

Founded in March 2000

501(c)3 (non-profit public charity)

Based in Boulder, Colorado

100% Funded by donations

Separate from the FreeBSD Project

Support critical needs of Project
Why We Should Work Together?

May work on multiple operating systems during your employment

Learn from each other. We both have successes and failures.

Different coding methodologies and philosophies – Understanding the reasons for both.

FreeBSD’s smaller code base makes it a great reference platform.

“Using and learning FreeBSD made me a better Linux admin and systems engineer.”
Why Contribute to FreeBSD

• Be part of an inclusive and welcoming community with a strong mentoring culture

• Great way to learn systems programming and study operating systems.

• The size of the project allows for a greater chance for anyone to make a notable impact.

• Some of the most notable BSD and FreeBSD Founders are still involved in the Project – And, they are approachable!

• Democratically run open source project allowing committers to commit their changes directly to the source tree without having to go through hierarchy of lieutenant model.
Get your hands dirty!

FreeBSD images available from all major cloud providers
- Amazon AWS
- Microsoft Azure
- Digital Ocean
- Gandi
- Vagrant
- Etc…

Or install in VMware / VirtualBox / …

https://www.FreeBSD.org/where.html

Download FreeBSD
Resources

Mailing Lists
Forums, Mailing Lists, IRC and Events
(https://www.freebsd.org/community.html)

Contributing to FreeBSD
(https://www.freebsd.org/doc/en_US.ISO8859-1/articles/contributing/)

FreeBSD Handbook

History
https://www.mckusick.com/history/

Forums:
http://forums.freebsd.org/

https://www.freebsdfoundation.org/journal/