

Becoming a FreeBSD developer

- hands-on workshop

Li-Wen Hsu (許立文) <lwhsu@FreeBSD.org>
<https://lwhsu.org>

Slides



Who am I

- Li-Wen Hsu <lwhsu@FreeBSD.org>
- <https://lwhsu.org>

Feel free to contact me for help in the future!

Agenda

- Preface
- Overview, VCS
- src
- ports
- doc
- Before submitting
- Submit
- After submitting
- Become a developer
- More Git
- Practice

Plan

- 9:00 - 9:15
 - Introduction
 - Plan
- 9:15 - 10:00
 - Lecture
- 10:15 - 11:00
 - Lecture
- 11:15 - 12:00
 - Hands-on exercises

Preface

What is an Operating System?



A set of lego blocks

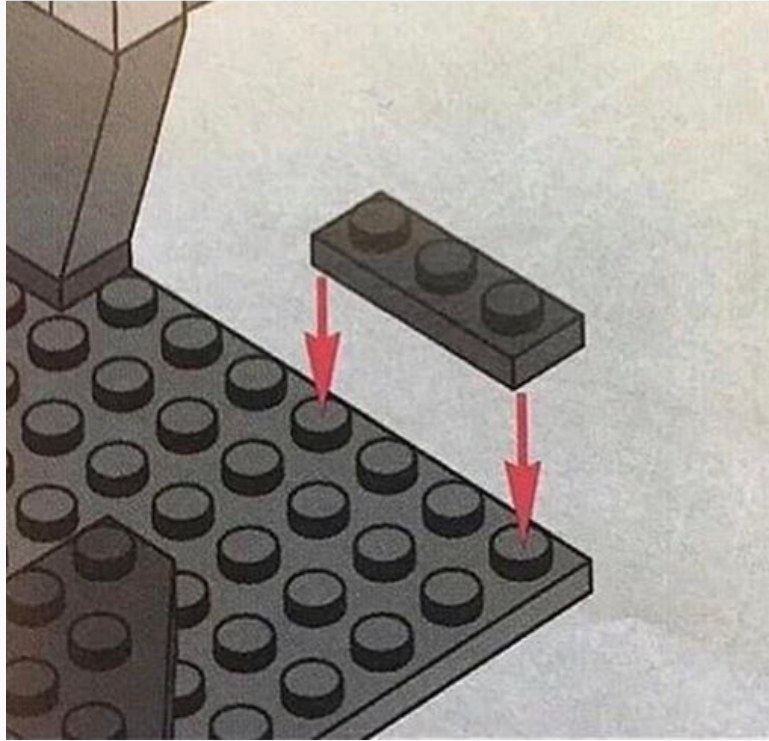
What do OS designers like to build?



Things they think are cool

Senior Developer: Just read the documentation!

The documentation:



This tutorial is to...

- Build a big picture of the FreeBSD development environment and process
 - For the most important 3 repositories
- Help you avoid the stupid things which may waste your time
 - Save energy and the Earth
- BTW, our documents (man pages and handbooks) are not bad in fact
 - We write codes and both documents (and you should do so as well)
 - Let us know if there is any issue or unclear part
 - Opportunity to contribute!
 - <https://docs.freebsd.org>

This tutorial is not...

- Going through the data structure, algorithm nor source code
- Instead, check the books, videos and other resources in reference
 - And ask in the communities!

main()

Environment

- Bare metal, virtual machine and cloud instance are all fine
- FreeBSD, latest release (13.2 currently, 14.0 should also work)
 - <https://www.freebsd.org/where/>
 - FreeBSD -CURRENT **MUST** be able to be built on the latest -STABLE
 - Latest -RELEASE should work mostly
 - You don't have to install "src" component on installing
- -CURRENT could be best, but on your own risk
- (Almost) everything needed for developing comes with default installation
 - Editing text (vi, ee)
 - Building (clang)
 - ~~Getting latest source code~~ (git)



bear^Wbare metal is
always the best!

Customize Environment

- "latest" pkg branch, if you like to stay on the bleeding edge
 - `vi /etc/pkg/FreeBSD.conf`
 - `url: "pkg+http://pkg.FreeBSD.org/${ABI}/latest"`
 - quarterly (RELEASE default) -> latest
- `pkg update; pkg install [-y] <...>`
 - Suggested: `pkg install -y git sudo tmux vim zsh`
 - And anything you like
- Setup your `~/.*` files and others

SCM

- <https://www.freebsd.org/developers/cvs/>
- We finally migrated to GIT in 2020 and 2021
 - It's just the first phase!
 - We're working on the second phase for a better git workflow
- If it's a headache, it's committers', not yours (yet)

“We should not debate on what is the best SCM.

We should let everyone can contribute to FreeBSD with their favorite SCM”

FreeBSD's Git repositories

- Repositories & (selected) branches
 - doc
 - main
 - ports
 - main, branches/yyyyQn
 - src
 - main, stable/N, releng/X.Y
- Focus on **main** first
 - `git clone -o freebsd https://git.freebsd.org/doc.git freebd-doc`
 - `git clone -o freebsd https://git.freebsd.org/ports.git freebd-ports`
 - `git clone -o freebsd https://git.freebsd.org/src.git freebd-src`

FreeBSD's Git infra

- **git.FreeBSD.org**
 - Geographically distributed read-only mirrors
 - (https) <https://git.freebsd.org/src.git>
 - (anonymous ssh) [anongit@git.freebsd.org:src.git](ssh://anongit@git.freebsd.org:src.git)
- **cgit.FreeBSD.org**
 - Cgit web repository browser
- **External mirrors**
 - GitHub, GitLab, Codeberg, etc.
 - Officially maintained
- **gitrepo.FreeBSD.org**
 - Centric developer-only write repository
 - Source of truth

GitHub

- We don't (directly) accept pull requests and issues on github currently
 - `src/CONTRIBUTING.md`
 - Problem report goes to <https://bugs.freebsd.org>
 - Patch goes to <https://reviews.freebsd.org>
 - Not really git friendly
 - We're working on a new system
- Different policies in each repository
 - We're working on fixing it

Workflow

- Get the code
 - Hack
 - Test
 - Submit
 - Refine
 - Merge
-
- Repeat
 - Fun & Profit(?)

src

src - the Base System

- Everything of an operating system in a single repository
 - Kernel
 - Boot loader
 - Utilities
 - Libraries
 - Manual pages

Kernel and World

- Kernel -> **/boot/kernel**
 - Kernel and the dynamic loadable modules
 - Under **src/sys**
- World -> *any* other things except **/usr/local**
 - Under **src/bin, src/usr/sbin, src/lib** ...
 - 3rd party software goes to **/usr/local/**
- make kernel & make world
 - Too dangerous
- build* & install*
 - **make -DNO_CLEAN buildworld buildkernel installkernel installworld**
 - (kernel only) **make -DWITHOUT_CLEAN=yes kernel**
 - Don't rebuild things you don't have to build
 - **-DWITHOUT_TOOLCHAIN**

Hack

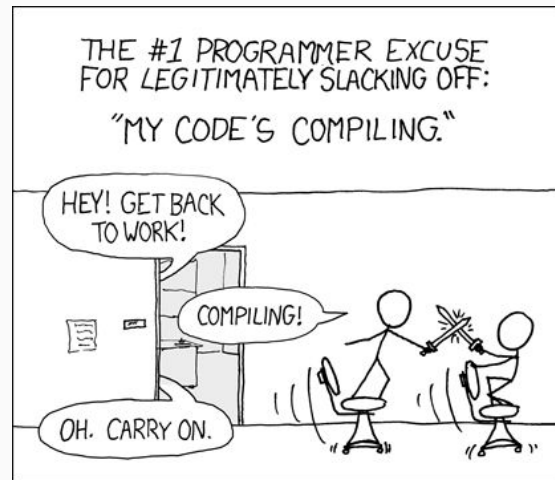
- You can checkout/clone the source to anywhere you like
- For beginning, **/usr/src** is safer
- (optional) use `mount_nullfs(8)`:
 - `sudo mount -t nullfs ~/freebsd-src /usr/src`

```
cd /path/to/src; hack; hack; hack
```

Build

```
cd /usr/src make -DNO_CLEAN -jN buildworld buildkernel
```

- N = `sysctl -n hw.ncpu`
- Object files go to `/usr/obj`
 - Set env `MAKEOBJDIRPREFIX` to change
- `/etc/make.conf`
 - `make.conf(5)`
 - Also affects ports building
 - `make __MAKE_CONF=~/.src/make.conf ...`
- `/etc/src.conf` & `/etc/src-env.conf`
 - `src.conf(5)`



Install & reboot & PRAY

- `cd /usr/src`
- `sudo make installkernel installworld`
- `sudo etcupdate`
- `sudo reboot`



If something goes wrong

- System boot menu -> “Escape to a loader prompt” ->

boot kernel.old

- **Before building the new kernel, backup the good one!**
- Handbook:
 - <https://docs.freebsd.org/en/books/handbook/kernelconfig/#kernelconfig-trouble>

Package and create VM image

- Like how we build at ci.FreeBSD.org
 - Available at <https://artifact.ci.freebsd.org/snapshot/>
- **cd /usr/src/release**
- **sudo make clean; sudo make base.txz kernel.txz**
- **This is not the optimal method**
 - Can be done by **make DESTDIR=foo installkernel installworld distribution** and some small extra steps.
 - See these targets in **release/Makefile** and *.txz targets in **release/Makefile**
 - You should use those after you get familiar with the build system

Package and create VM image (c.)

- `mkdir -p ufs`
- `sudo tar Jxf base.txz -C ufs`
- `sudo tar Jxf kernel.txz -C ufs`
- `sudo tar Jxf tests.txz -C ufs`
- `dd if=/dev/random of=ufs/boot/entropy bs=4k count=1`
- `sudo makefs -d 6144 -t ffs -f 200000 -s 8g -o version=2,bsize=32768,fspace=4096 -Z ufs.img ufs`
- `mking -s gpt -f raw \
-b ufs/boot/pmbr \
-p freebsd-boot/bootfs:=ufs/boot/gptboot \
-p freebsd-swap/swapfs::1G \
-p freebsd-ufs/rootfs:=ufs.img \
-o disk-test.img`
- Install 3rd party packages into image:
 - https://github.com/freebsd/freebsd-ci/blob/master/scripts/build/build-test_image-head.sh#L65

Emulators

- Bhyve
 - Create disk image and run
 - `sh /usr/share/examples/bhyve/vmrun.sh -m 4096m -c 2 -d disk-test.img <test-vm-name>`
- QEMU
 - <https://wiki.freebsd.org/QemuRecipes>
 - `src/.cirrus.yml`
 - `tools/boot/ci-qemu-test.sh`
 - `build-test_image-head.sh`
- VirtualBox, VMWare, etc.

tests(7)

- ATF/Kyua by Julio Merino <jmmv@{NetBSD,FreeBSD}.org>
 - C/C++/sh unit test framework and test driver
 - <https://github.com/freebsd/atf> (framework, imported as contrib/atf)
 - <https://github.com/freebsd/kyua> (driver, imported as contrib/kyua)
 - <https://github.com/jmmv/kyua/wiki/Quickstart>
 - **cd /usr/tests; kyua test**
- <https://ci.FreeBSD.org>
 - <https://github.com/freebsd/freebsd-ci>
- <https://wiki.freebsd.org/HostedCI>

Continuous Integration

- <https://ci.FreeBSD.org>
- <https://github.com/freebsd/freebsd-ci>
 - Merging to src repository

Userspace only changes

- `cd src/bin/cat (and src/bin/cat/tests)`
 - (actual test codes are in `src/contrib/netbsd-tests/bin/cat`)
- `hack; hack; hack`
- `make depend all`
- `make install`
- `cd /usr/tests/bin/cat`
- `kyua test`

Cross Compiling

- TARGET/TARGET_ARCH
 - `make targets`
- Build env
 - `make TARGET=riscv TARGET_ARCH=riscv64 buildenvvars`
 - `make TARGET=riscv TARGET_ARCH=riscv64 buildenv`
 - Entering world for riscv64:riscv
For ZSH you must run: `export CPUTYPE=`

Manual pages

- `src.conf(5)`
- `make.conf(5)`
- `build(7)`
- `release(7)`
- `development(7)`

Tools

- Language Server
 - <https://docs.freebsd.org/en/articles/freebsd-src-lsp/>
- Online Manual Page Editor
 - https://wang-yan-hao.github.io/man_page_editor/root/

There are still much, much more...

- More make command line options
 - Comments in the top of `src/Makefile.inc1`
- Non x86 system development
 - `make TARGET=riscv TARGET_ARCH=riscv64 kernel`
- Attach debugger
- Build with ccache
 - `WITH_CCACHE_BUILD` in `src.conf`
 - `pkg install ccache; /usr/local/share/doc/ccache/ccache-howto-freebsd.txt`
- External toolchain
 - Example: `aarch64-gcc6, riscv64-gcc9, llvm-devel`
 - `make CROSS_TOOLCHAIN=amd64-gcc kernel`
- ...etc, let's discuss with the community!

Ideas?

- We have good and experienced mentors, and eager good mentees
- <https://wiki.freebsd.org/IdeasPage>
- <https://wiki.freebsd.org/SummerOfCodeIdeas>

Reference

- FreeBSD Developers' Handbook
 - <https://www.freebsd.org/doc/en/books/developers-handbook/>
- FreeBSD Journal
 - <https://www.freebsdoundation.org/journal/browser-based-edition/>
- The Design and Implementation of the FreeBSD® Operating System, Second Edition
 - Addison-Wesley Professional ©2014 ISBN:0321968972 9780321968975
- FreeBSD Device Drivers
 - <https://nostarch.com/bsddrivers.htm>
- Designing BSD Rootkits
 - <https://nostarch.com/rootkits.htm>
 - <https://github.com/lwhsu/dbr/> (updated example code)
- FreeBSD Kernel Internals: Data Structures and Algorithms
 - <https://www.mckusick.com/courses/introorderform.html>
- FreeBSD Kernel Internals: An Intensive Code Walkthrough
 - <https://www.mckusick.com/courses/advorderform.html>



ports

ports(7) - contributed applications

- <https://man.freebsd.org/ports>
- Instructions to fetch, patch, build, install 3rd party software
 - Standardization
 - Maintenance

Become a Ports Developer!

```
echo DEVELOPER=yes >> /etc/make.conf
```

Quick Start

- (copy from other similar ports)
- (edit Makefile)
- make fetch
- make makesum
- make check-orphans
- Test build & run in a vanilla environment

Poudriere Setup

- `pkg install poudriere-devel`
- `/usr/local/etc/poudriere.conf`
 - `ZPOOL=tank`
`FREEBSD_HOST=https://download.FreeBSD.org`
`DISTFILES_CACHE=/tmp/distfiles`
`ALLOW_MAKE_JOBS=yes`
- `poudriere jail -c -j 13_2_amd64 -v 13.2-RELEASE -a amd64`
- `poudriere jail -c -j 13_2_i386 -v 13.2-RELEASE -a i386`
- `poudriere ports -c -f none -m null -M ~/freebsd-ports`

Poudriere Test

```
#!/bin/sh -ex

PORTS="$*"

ACTION="bulk -tr -b latest -NN -v -C"

JAILS="${JAILS} 13_2_amd64"
#JAILS="${JAILS} 13_2_i386"
#JAILS="${JAILS} 12_4_amd64 12_4_i386"

for j in ${JAILS}
do
    sudo poudriere ${ACTION} -j ${j} ${PORTS}
done
```

Tips

- Useful targets
 - {build,run,all}-depends-list
 - Install-missing-packages
 - patch
 - patch after patch
 - make makepatch
- `${WRKDIR}/.*`
- `ports/Mk/`
 - `bsd.ports.mk` - read the comments
- `ports/Mk/Uses`

Tools

- ports-mgmt/porttools
 - `port create`
- ports-mgmt/portlint
 - `portlint -abct`
- ports-mgmt/portfmt
 - `portfmt Makefile`
- ports-mgmt/portfmt
 - `portclippy Makefile`
- Portscout
 - <https://portscout.FreeBSD.org>

References

- The Porter's Handbook
 - <https://docs.FreeBSD.org/en/books/porters-handbook/>



doc

Repository Layout

- AsciiDoc
 - asciidoctor
- Hugo
- documentation/
 - <https://docs.freebsd.org>
- website/
 - <https://www.freebsd.org>

Environment

- `pkg install docproj`
- `pkg install docproj-fonts-cjk`

Build

- make
- make LANGUAGES=en

Check

- public/
 - .html files

Tools

- Online editor
 - <https://wang-yan-hao.github.io/FreeBSD-Online-Document-Editor/root/>

Translation

- Weblate
 - <https://translate-dev.freebsd.org>

References

- FreeBSD Documentation Project Primer for New Contributors
 - <https://docs.freebsd.org/en/books/fdp-primer/>



Before submitting

Licensing

- Copyright header
 - Author
 - SPDX

Submit

Issue Tracking

- Issue tracker:
 - <https://bugs.freebsd.org>
 - Bug report, feature request, change request
 - Can link core review system with “URL” field
 - CLI tools: pkg search bugzilla

Ask someone in the community to help you go through all the process!

Patch Review

- Code review system:
 - <https://reviews.freebsd.org>
 - `pkg install arcanist-php83`
 - `arc install-certificate https://reviews.FreeBSD.org`
 - `arc diff --create freebsd/main`
 - `arc diff --update DXXXXX freebsd/main`
 - <https://wiki.freebsd.org/Phabricator>

Ask someone in the community to help you go through all the process!

GitHub Pull Request

- There are too many documents
 - Search "Create GitHub Pull Request"
- <https://github.com/freebsd/freebsd-src/pulls>
 - Simple building & booting CI
- Check the Author information
- Sort the commits and commit messages

After submitting

Questions & Work with others

- Mailing list: <https://lists.freebsd.org>
 - freebsd-hackers
- Wiki: <https://wiki.freebsd.org>
- IRC: <https://wiki.freebsd.org/IRC/>
 - Especially channels on EFnet and LiberaChat
- Matrix
 - <https://wiki.freebsd.org/Matrix>
- Worldwide user groups
 - Conferences
 - AsiaBSDCon, EuroBSDCon, BSDCan
 - <https://www.freebsd.foundation.org/what-we-do/grants/>

Become a FreeBSD Developer

Engagement

- Join discussion
- Be helpful
- Attend conferences
- Know more developers

Articles

- Contributing to FreeBSD
 - <https://docs.freebsd.org/en/articles/contributing/>
- Committer's Guide
 - <https://docs.freebsd.org/en/articles/committers-guide/>

More Git

Git Tips

- <https://wiki.freebsd.org/GitTips>
- Local Hooks
 - src: `cp tools/tools/git/hooks/prepare-commit-msg .git/hooks`
 - ports: `git config --add core.hooksPath .hooks`
 - doc: `git config --add core.hooksPath .hooks`
- config
- remote
 - Work with other git hosting
 - Work with others
 - https://docs.freebsd.org/en/articles/committers-guide/#_collaborating_with_others

Git Tips (c.)

- format-patch
- rebase -i
- worktree
- Signed-off-by
 - git commit -s
- Signature
 - user.signingkey
 - commit.gpgsign = true
- push
 - --sign

Practices

Possible tasks

- ~~Type~~
- Update ports
 - <https://portscout.FreeBSD.org>
- Manual pages
 - Add/Check Examples
- Doc fixes
 - Out of date contents
- Tests
- sys/modules
 - Fix directly building in sys/modules
- Search bugzilla

Backup slides

Code Search & Cross Reference

- <http://fxr.watson.org/>
- <http://bxr.su/>
- <http://github.com/freebsd/freebsd-src>

Building on non-FreeBSD hosts

- <https://wiki.freebsd.org/BuildingOnNonFreeBSD>
 - Building in a docker container
 - <https://github.com/arichardson/freebsd/tree/crossbuild-aug2018/tools/build/cross-build/docker>

Debug

- ktrace/kdump
- <https://wiki.freebsd.org/Debugging>
- FreeBSD Journal November/December 2018, "Debugging the FreeBSD Kernel"
 - <https://www.freebsd.foundation.org/past-issues/freebsd-12/>

Non-coding contribution?

- Documentation!
 - Doc modern project
 - Translation
 - <https://translate-dev.freebsd.org/>
 - Blog & tutorial articles, presentations
 - Data collection
 - <https://papers.freebsd.org>
 - <https://github.com/freebsd/freebsd-papers>

Manual page tools

- `mdoc(7)`
- `mandoc -T lint`
- `textproc/igor`