

mfsBSD in Base

BY SOOBIN RHO

mfsBSD is an in-memory FreeBSD.

What makes mfsBSD different is that it runs an instance of FreeBSD completely in memory — hence the mfs (memory file system). Of course, this means we can boot into FreeBSD without affecting our existing drives at all when using mfsBSD. We can, for example, use it for troubleshooting cloud or on-prem servers¹. It's fascinating to see what kind of problems people have managed to solve using mfsBSD (search mfsBSD on the mailing lists). My personal favorite would be installing FreeBSD in a system where there's only a single drive available for whatever reason. I would use FreeBSD installation media if I could just stick in a USB drive, but if access to external devices was prohibited, I would first build an mfsBSD image; install the image to the drive; boot into mfsBSD; and run `bsdinstall`. Here's an example:

First, build mfsBSD:

```
# The patch set that integrates mfsBSD into base is under review and is available at:
# https://reviews.freebsd.org/D41705
cd /usr/src/release
make mfsbsd-se.img

# se here refers to mfsBSD special edition, which comes packed with
# dist files - base.txz and kernel.txz - which are required for bsdinstall.
cd /usr/obj/usr/src/${ARCH}/release/
ls -lh
```

Then, write mfsBSD to the drive:

```
# Install the mfsBSD image to your target drive.
# Replace ada0 with your target drive.
dd if=./mfsbsd-se.img of=/dev/ada0 bs=1M
reboot
```

Boot into mfsBSD and then execute `bsdinstall`:

```
# Copy the special edition dist files so that bsdinstall can use them for installation.
mkdir /mnt/dist
mount /dev/ada0p3 /mnt/dist
mkdir /usr/freebsd-dist
cp /mnt/dist/<version>/*.txz /usr/freebsd-dist/
bsdinstall
```

This is possible because mfsBSD loads a FreeBSD system entirely in memory. After mfsBSD has been loaded, the original disk can be modified completely as all mfsBSD files are now running in memory. As Matuška describes in his 2009 white paper, "mfsBSD is a

toolset to create small-sized but full-featured mfsroot based distributions of FreeBSD that store all files in memory.”²

Brief History of mfsBSD

[Martin Matuška](#) wrote mfsBSD. Looking back at the repository’s commit history, he made the first ever commit on November 11, 2007, which is around the time FreeBSD 7.0 BETA was released. “This project [mfsBSD] is based on ideas from the Depenguinator project,” which was a 2003 project by Colin Percival to create a toolset to remotely install FreeBSD on dedicated servers that only offer Linux distributions³. Matuška wanted to provide the Depenguinator functionality for the FreeBSD 6.x, and that’s how mfsBSD started.

Since then, Matuška maintained <https://mfsbsd.vx.sk/> to distribute images of mfsBSD as it gained popularity and has maintained its source code on GitHub⁴ for the past seventeen years, fixing an uncountable number of bugs along the way, and adding support for `zfsinstall`, compressed tar of `/usr`, and so on.

In May, 2023, one year before this article was written, a Google Summer of Code project to integrate mfsBSD into base began.

Google Summer of Code

How did it all start? I was reading *Hacker News*. (probably procrastinating on my college assignments at the time). That’s how I first came across Google Summer of Code (GSoC). One of the top comments there said that FreeBSD was one of the participating organizations, and what I found interesting about the commentary was that there was no mention of anything else, as if everything else was self-explanatory.

I got hooked right away. The most surprising thing about FreeBSD was that macOS is a derivative of FreeBSD, and also that Netflix uses FreeBSD for its CDN. The GSoC application process involves submitting a project proposal. It was strongly recommended that applicants find a project idea from each of the organizations’ project idea lists (unless you had your own idea that you’d like to pursue). I had a look at the list, and the mfsBSD project was the most interesting to me because other project ideas seemed closer to kernel development than I was comfortable with.

After shooting an email to my mentors, I got email back from [Joseph Mingrone](#) and [Juraj Lutter](#); had a brief zoom call; and a few weeks later I got an acceptance from GSoC. After that, we had what’s called a community bonding period, at which all of the contributors and mentors gathered around and had a virtual meeting for half an hour introducing ourselves. That was Friday, May 12, 2023.

mfsBSD in Base

Three months and twenty-two days later, the project to integrate mfsBSD into base finally came to completion, after a lot of back and forth between debugging (a lot of the bugs were resolved by googling and digging through all the past GitHub Issues), and testing (sh scripts with two of my laptops, Thinkpad T440 and P17), and me asking too many questions to my mentors. A set of three patches were pushed to Phabricator.⁵



In May, 2023, one year before this article was written, a Google Summer of Code project to integrate mfsBSD into base began.

Basically, the first commit “mfsBSD: Vendor import mfsBSD” imports mfsBSD as `contrib/mfsbsd`. The main commit “release: Integrate mfsBSD image build targets into the release tool set” adds `mfsbsd-se.img` and `mfsbsd-se.iso` targets in `release/Makefile` as `release/Makefile.mfsbsd`. The last commit “release(7): Add entries for the new mfsBSD build targets” adds corresponding entries on `share/man/man7/release.7`. What this means is that we can now build mfsBSD in the same release Makefile we use for building all the FreeBSD installation media, such as cdrom, dvdrom, memstick, and mini-memstick, as `make release WITH_MFSBSD=1`.

Now, the patch set is under review. mfsBSD previously existed outside the FreeBSD release tool chain, and only the release versions have been produced. What I envision with this patch set is to make mfsBSD images available as a part of base, and with this, we will be able to build custom mfsBSD images by invoking `cd /usr/src/release && make release WITH_MFSBSD=1`, which will then create `mfsbsd-se.img` and `mfsbsd-se.iso` at `/usr/obj/usr/src/${ARCH}/release/`.



This means is that we can now build mfsBSD in the same release Makefile we use for building all the FreeBSD installation media.

References

1. Matuška, Martin. (2022). FreeBSD on Hetzner dedicated servers — VX Weblog. [online] Available at: <https://blog.vx.sk/archives/353>
2. Matuška, Martin. (2009). mfsBSD Toolset to create memory filesystem based FreeBSD distributions. [online] Available at: <https://people.freebsd.org/~mm/mfsbsd/mfsbsd.pdf>
3. Colin Percival. (2003). Depenguinator — FreeBSD remote install. [online] Available at: <http://www.daemonology.net/depenguinator/>
4. Matuška, Martin. (2024). mmatuska/mfsbsd. [online] GitHub. Available at: <https://github.com/mmatuska/mfsbsd>
5. <https://reviews.freebsd.org/D41705>

SOOBIN RHO is a college senior at Augustana University in South Dakota. Born in South Korea but raised in Dubai, he ended up going to a college in the US and is now a part-timer at The Bancorp’s Cybersecurity Department, which he will join as an Information Security Analyst after graduation. He has been a FreeBSD contributor since 2023 Google Summer of Code.