

Installing FreeBSD 1.0: 30 Years On

BY TOM JONES

Getting FreeBSD 1.0 running on real hardware is an opportunity for me to explore computing from before my time. My earliest memories of using computers pop up with Windows 98—I'm sure there were other computers in my life. The machine on which my friends and I played Castle Wolfenstein was probably a hand me down or a surplus office machine. It wasn't until I had the first computer that was really mine (a G4 iBook in the mid 2000s) that I started to look inside things and explore operating systems.

I jumped at the chance to cover the subject of this article. I am sure the editorial board was envisioning this would cover getting FreeBSD 1.0 to run in one of the many x86 emulators, but I am not one to let an opportunity to overdo something go by.

Retro computing is a growing interest. For a long time in the UK, Charity Shops (think Goodwill) were unable to take electronics. That created a situation where throwing away computers was the easy option for those unwilling to try to sell them on a cold Sunday morning at a car boot sale.

This key factor created an absolutely perfect situation for our computing history to be disposed of and destroyed. For the last decade, I have tried to save the machines I could as they came by. This is why I have an inventory of 50 machines waiting for space in the Center for Computing History, a vt320 on my desk, and a DEC-PRO350 as a conversation piece in the hackerspace.

A collection is one thing, but these machines don't want to be in a museum, they want to run!

In the last decade, there has been a growing movement of retro computing enthusiasts. There are channels like Action Retro, Lazy Game Reviews, and This Does Not Compute which go to obsessive extremes to push old machines to their absolute limits. This can be done by installing accelerator cards from when the machines were still supported and through adapters, modern storages, and peripherals that get machines to scream-along at speeds the original owners could only have imagined.

In this world of renewed interest in classic computers, we get to learn from the large enthusiast community and use the excellent tools and adapters they have created. But we also have to pay the price of the popularity, and with the extra interest and demand, prices for old hardware have gotten silly.

How Do I Find a Machine to Run FreeBSD?

If this was a question from a normal person (not someone deranged trying to relive 1993) I think most of us that have used FreeBSD for a while would offer similar advice:

- research your hardware and figure out what is supported,
- if you are getting a new machine, make sure it has support.

I wanted to buy a machine that managed to meet three criteria—era accurate—supported by FreeBSD 1.0—(ideally) supported by 386BSD.

Why era accurate? Whatever machine I got hold of, I wanted to be able to give a long life. Its first role is to run FreeBSD 1.0, but I also wanted to be able to photograph it, take it to conferences, and use it as a platform for learning more about computing in the foundational times of free BSD UNIX. "What is the newest hardware that FreeBSD 1.0 will run on" is a different article, and not the article I wanted to cover. Era accurate seemed to be the path of least resistance.

Supported by FreeBSD 1.0 should be self-evident, but why 386BSD? I want more from the machine than just 1 article. I want to be able to show people where things came from—386BSD as a project holds a lot of mystique.

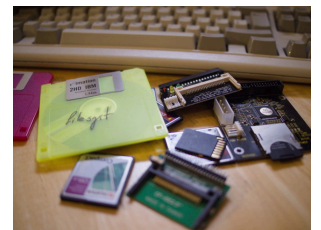
From 30 years in the future, it is very difficult to understand what sort of machines were actually around in 1993. This is a form of the set dressing problem that period TV shows and movies hit all the time. You can't really look at a magazine from the 1990s and see the typical home from the 1990s. Instead, people build up their possessions over a long period. A home from the 1990s might have a piece or two that is fashionable, but it won't look like that years' Ikea catalog.

We are saved a little bit from this by computing in the early 1990s moving so quickly—machines had shorter life times. I would set out to get hardware that would fit into the era well.

To find hardware that will actually run our OS we need to find the release notes and installation instructions. Today, figuring out if FreeBSD will run on a machine without the machine can be quite hard, but, hopefully, the original team did a better job at spelling out the supported hardware.

There is a collected CD of FreeBSD 1.0 available from the project and the Internet archive. It contains the release floppy disks, source code, and ports and packages for FreeBSD. In the CD are also the release announcement and installation instructions for various methods.

The FreeBSD 1.0 release announcement gives us some information about the minimum requirements to run. It gives us an idea about which supported processors, memory requirements and additional hardware which may work.



Finding a Machine

With a rough outline of the supported hardware, the next step is to track down a machine. First, I thought I might be able to borrow a 386 from 'someone'. After a couple of weeks of trying IRC channels and striking out, it seemed that anyone with a working 386 was quite precious of the machine and not willing to lend it to a stranger to write an article about installing FreeBSD.

My next bet was to buy a machine.

Here I encounter a problem unique to the island I am currently stuck on. We do not have old computers. Okay, that isn't true, I can get all the Amigas I want and probably a container load of Research Machines PCs from the early 2000s, but an actual 386 is a different matter.

Building from components didn't seem to be a good call. Maybe if I had some history with machines of this vintage, but finding compatible components to build a computer today can be difficult to figure out. How was I supposed to get compatibility information about 30-year-old components which were made by hundreds of different vendors.

Complete systems seemed to be a bust too. The few I saw on eBay in the UK were going for 1000GBP+, a little bit outside of my normal 386 budget.

Eventually, after complaining about my uniquely me situation on a phone call with some friends, one of them started digging around on eBay and found a Czech seller. My friend, more familiar with this period of computing was able to confirm that these smelled good and were the right vintage to fit into this article.

Some hemming and hawing and then I settled on the one with a large CPU frequency display on the front—ever one to ignore practicalities and land on aesthetics. (They were to me, functionally the same).

The seller's specifications for the machine are:

CPU AMD 368DX 40MHz
4MB RAM + 128kb Cache
MB Shuttle Hot-327
VGA ISA Trident TVGA 9000c 512kb
HDD ST 3243A - 214 MB
HDD Controller UMC PIO-001 RP 20070R 12
Soundcard ESS 688F Pine Technology
CD-ROM Wearnes 622 - K.O.
FDD TEAC FD-235HF

It was clear to me that I wanted to get this machine onto the Internet in the course of this article and so I picked up the only Ethernet interface I could find that was compatible—an Etherlink II 3c509 with BNC and AUI connections.

While the INSTALL files says that FreeBSD will run in 4MB of memory, this seemed like an easy fix, and I picked up quite cheaply an additional 32MB of RAM to give the system a little more breathing room.

Installing FreeBSD 1.0 (first attempt)

With the machine delivered, I was excited to test the hardware and get to installing FreeBSD. The system shipped with MS-DOS

6.22, an operating system that is entirely alien to me. I made sure the hardware was functional by grabbing the shareware of wolf3d and playing through a couple of levels.

The CD-ROM archive offers a few options for installing FreeBSD 1.0, all of which start with creating a set of boot floppies that build a minimal system from which the full install can be finalized.

The minimal install requires 3 floppy disks. To avoid having to rewrite a single disk, I bought a box of 'new' floppies from amazon (10 for 10GBP). I also acquired from the local hackerspace a selection of USB floppy drives to help me bootstrap into the past.

The three disks for installing are a kernel floppy, the file system floppy, and the cpio floppy. The kernel floppy comes in several different flavors depending on your SCSI controller. I don't have a SCSI controller at all, and so either would do.

Creating the disks from a modern operating system was incredibly painful. The USB floppy drives just don't want to work, but when the moon was right, I was able to create the three required disks from FreeBSD-14 with this dd command:

```
$ sudo dd if=kcoppy-ah-floppy of=/dev/da0 bs=30b
```

The installation process requires you to boot with the kernel floppy in the floppy drive:

Error! Filename not specified.

Once the kernel is loaded you are presented with a prompt to insert the file system disk and continue.

Then finally you get to the installer.

I was excited when I got here, I had acquired a modern piece of hardware, a SD IDE adapter to allow me to preserve the MS-DOS 6.22 so I could continue verifying the hardware.

This device did not work at all for me.

Crestfallen and impatient, I decided to hell with it and wiped the MS-DOS disk entirely (who needs dos anyway, I'm getting UNIX!). Setting up FreeBSD to take the entire 200MB of spinning rust for itself.

With the disk formatted a minimal system is installed. You reboot from the kernel floppy again, this time going to a **kc>** prompt where you type 'copy' for the kernel to copy itself to the disk.

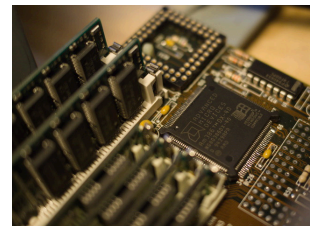
Next you reboot, and when prompted, install the cpio floppy. It will walk you through what it needs you to do given a minimal installed system. Next, you have to grab the install sets, installing at a minimum the **bin.tar.gz.xx** file set.

The full install media is made available across a set of dozens of disks, arranged so they can be loaded one floppy at a time onto the system to finish the install. If you were lucky enough to have tape, serial, a MS-DOS partition, or network access you could pull them all as a single archive.

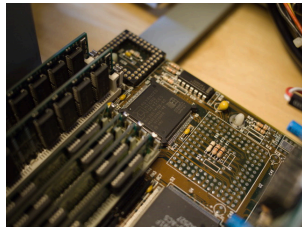
By not reading ahead in the instructions and being impatient, I made a mistake that made this much more difficult for me. My DOS partition was gone, and I didn't have a network. Tape wasn't an option at all, I've never even seen a supported tape.

Maybe I could pull the files from the CD drive? I hit the eject button on the drive for the first time (weirdly it wasn't part of my shooter-based stress test) and while the motor whined, nothing happened. The CD drive never came to life.

In the list of options is the ability to install via serial. I can do



serial. I hooked up a USB serial adapter to the serial port on the PC and began trying to get Kermit running, which does not ship on the install floppies at all.



The install instructions talk about a 'dos' floppy, but even with several checks, I couldn't find it inside the CD archive. Eventually, I caught the note early on that the 'dos' floppy is an optional disk of utilities which can be pulled from the CD archive. I put Kermit onto a floppy booted into my primordial system and copied across. Hours of fighting later, I managed to figure out the correct modern Kermit commands to speak to 1993 Kermit and started transferring files. At 9600bps, restart higher rate, I tried pushing it up and eventually managed a reliable 56k.

Three days later (or maybe an hour or two, if I am more realistic) I had the base images bin copied onto my host and could finish the install. Each set (binary, src, X) is installed with an extract shell script that comes from the preinstall environment.

Once you have installed the set you want, you need to run the 'configure' script that does the finalization of the preinstall environment and gives you a full working FreeBSD system.

Here I got to pick the hostname and IP address for interfaces.

With the install done, I booted into my new FreeBSD system and immediately posted a victory message both on social media and to the editorial team—"6 months away and the system works! That means the article is pretty much done right?"

Reinstalling FreeBSD 1.0 (second attempt)

My first install was fun, but I had installed onto very old spinning rust, and this worried me. I wanted to do some modernizations. The popularity of vintage computing like this has led to a bunch of devices to replace the finicky and annoying parts of retro computing.

Compact flash (also a rather old technology) is compatible with IDE but has the benefit of being much easier to purchase and not as likely to die randomly with a scream of horror.

I managed to get a CF adapter that lives in a PCI bracket making swapping disks for the OS really easy. Of the CF cards I bought, only 1, a 250MB one, will boot on my system. I suspect the bios freaks out seeing a 1GB (let alone a 16GB) hard drive.

The system I had installed had some oddities and I thought they might be related to the fractured install process from above (it took several weeks of occasional effort to get the first install together).

First, when the system comes up, it always prints a message saying '**Automatic reboot in progress**' even when it doesn't reboot. While there was an adduser man page there was no command to add a new user. I have new media now, so I thought I would try again.

For the second install, the floppy phases were the same, the disks survived three months without randomly dying.

In the interim, I picked up a Farallon Ether10-T Starlet 9 base10-T Ethernet hub. This device with the correct terminators and 50 OHM BNC cables allowed the 3c509 network card I bought to pull my 386 box onto the Internet (and into the future!).

My second install was much, much smoother. I did the base install to get to the minimal system. Once booted from it, I could then use the network to grab the images. I considered ftp to move them to the 386 but spotted the NFS option in the install document and tried that. FreeBSD 1.0 will happily mount a NFS share on a FreeBSD 14 box :D.

FreeBSD 1.0 at home

Don't fancy trawling eBay to find old, expensive, noisy and unreliable hardware, but want to try out FreeBSD 1.0? I guess I can see why you might not want to reproduce my steps, be warned that you are missing out on a certain amount of joy by cheating and using QEMU.

I have created a backup of the CF card install I did, and I have placed it here: <https://people.freebsd.org/~thj/freebsd-ninety-three.img.xz>

Grab that image, uncompress it. With a standard install of QEMU (I used QEMU 7.0 on FreeBSD 14-CURRENT) you can launch the image with the following command:

```
$ qemu-system-i386 -hda freebsd-ninetythree.img
```

You will see from the system log that the machine doesn't find any interesting hardware. Left as an exercise to the reader is figuring out networking (if ISA emulation proves too hard, I would look into slip).

The only CF card that worked in my system was still a little small at 250MB, but there is enough of a system there for you to play with. If you want to try on a larger (or smaller disk) then you can figure out all the QEMU bits required to get it going.

With the install done, I rechecked adduser, but the command was still not there. Reading the man page this time, I discovered that the man page was just the process for adding a new user to the system! No automation for you, it is 1993.

Installing Software

The CD-ROM archive comes with quite a lot of additional third party software for FreeBSD. There are both the ports collection, software that has been modified to work on FreeBSD, and prebuilt packages of ports, and some additional software.

From ports, I built top so I could have the 386 idle in interesting ways.

I also grabbed vim and zsh from the prepackaged software to make living on the machine a little nicer.

The next thing I wanted to do was try and serve web content from the 386. I grabbed both ncsa-httpd (all releases available as a git repo) cern-httpd. Source-based distribution of software has changed a lot in 30 years. We have no idea how good we have things now.

ncsa-httpd ships with a Makefile to build which was much friendlier than the cern-httpd script, so I started with ncsa-httpd. ncsa-httpd doesn't include FreeBSD in its list of targets, but it does have NetBSD, so I tried that. Sadly, the era correct NetBSD is different enough that this didn't work out of the box. I ended up modifying the Makefile to pick up the FreeBSD build support that is in the c files, but not the build script.

With ncsa-httpd built, I went through a process of running the generated binary and creating the required directories and moving the right files it complained about being missing until I could get it to start. An install target for make would have been too much to ask for.

I whipped up an index.html file and started serving up some pages.

Not yet being willing to put my 386 on the internet, I, instead, settled for seeing how much work my 386 could actually do as a

web server, by running the wrk web server benchmark.

In case you need to know a 40 MHz 386DX with 36MB of RAM can serve about 22 requests a second without any tuning.

Running X

I have managed to install X from the sets on the CD. I have not managed to get X to display anything and, instead, each attempt requires me to reboot to regain access to the system console.

Trying X highlighted that I was missing virtual consoles. Grepping the documentation didn't reveal much.

I stumbled onto the `syscons` command (not driver as we have today, command). When trying to use `syscons` to change the virtual tty, all I got was a helpful message about needing to rebuild the kernel with `syscons` enabled.

Rebuilding the Kernel

There is a Makefile with a familiar `world` phase in `/usr/src`, but the kernel part of building the system isn't documented there at all. I stumbled into the `config` man page, which hinted enough that I could build a new kernel.

Building a new kernel is roughly:

- cd to the conf sub directory (usually `/sys/ARCH/conf`), for us `/sys/i386/conf`
- run `config` with a system configuration file (I copied the `SYSCONS` config as `NINETYTHREE`)
- cd to `../..../compile/SYSTEM_NAME` (`NINETYTHREE` in this case)
- run `make depend`
- run `make all`
- copy the `386bsd` kernel binary to /
- reboot

The kernel build took 33 minutes (5 for `make depend`, 28 for `make all`).

FreeBSD 1.0 now and forever

I keep getting lost trying to get different parts of the system going. My next steps are figuring out how to get running on my extra virtual consoles, and once I find a serial mouse, running X. Rather than a straightforward install story, this project has ended up being an absolute ton of fun for me. I really didn't need a web server for this article, but I couldn't resist getting one up and running once I had the idea.

I picked up enthusiast magazines from the period to learn the interesting goings on in computing in the early 1990s. I got the Dr.Dobbs articles on 386BSD to help stage some of these pictures.

Sure, I had to fight around on ebay to buy over-priced parts, but that has its own draw, there is a high when you finally find a listing for something obscure you were looking for.

FreeBSD 1.0 is like looking at your now adult little brother and seeing a small child there. All the wonder and magic of the world is back. The bugs I encountered feel entirely fixable. I am sure it is a rough system, but the edges are asking to be smoothed. I entirely understand why the early contributors got involved, it is hard to resist finding something to hack on.

This all said, I don't think you, or anyone else, should go lightly into a project like this. The hardware is difficult to find 30 years on and it won't get easier. Prices for hardware are going to go up and reliability of components will go down.

The options are to experience the joys of the platform without having to go through the suffering of drives and disks dying, finding termination for the BNC cables, or teaching software that FreeBSD is an operating system. There are many choices to emulate this hardware and much of the release information is available with source tarballs to build applications.

I think your 386 FreeBSD should be virtual, but you will have to fight me to get this 386 FreeBSD.

TOM JONES is the FreeBSD Engineering Manager at Klara Inc, he only gets to play with fun old computers in his spare time.

FreeBSD 1.0 dmesg

```
FreeBSD 1.0.2 (GENERICAH) #0: Sun Nov 14 18:22:24 PST 1993
root@gndrsh.cdrom.com:/usr/src/sys/compile/GENERICAH
CPU: i386DX (386-class CPU)
real mem = 38006784
avail mem = 35909632
using 819 buffers containing 6709248 bytes of memory
Probing for devices on the ISA bus:
pc0 at 0x60-0x6f irq 1 on motherboard
pc0: type color
sio0 at 0x3f8-0x3ff irq 4 on isa
sio0: type <16450>
sio1 at 0x2f8-0x2ff irq 3 on isa
sio1: type <16450>
sio2 not found at 0x3e8
sio3 not found at 0x2e8
lpt0 not found at 0x3bc
lpa0 at 0x378-0x37f on isa
lpa1 not found at 0x278
fd0 at 0x3f0-0x3f7 irq 6 drq 2 on isa
fd0: unit 0 type 1.44MB 3.5in
wd0 at 0x1f0-0x1f7 irq 14 on isa
wd0: unit 0 type SMART CF
ahb0 not found
aha0 not found at 0x330
wt0 not found at 0x300
mcd0 not found at 0x300
ed0 at 0x280-0x28f irq 5 maddr 0xd8000 msize 8192 on isa
ed0: address 02:60:8c:7c:22:04, type 3c503 (16 bit)
ed1 not probed due to maddr conflict with ed0 at 0xd8000
ed1 not probed due to irq conflict with ed0 at 5
ie0 not found at 0x360
is0 not probed due to I/O address conflict with ed0 at 0x280
npx0 on motherboard
npx0: 387 Emulator
biomask 4040 ttymask 3a netmask 3a
```