Computers (/Computers.cl) → Retro (/Retro.cl) →

Published on | Jul 06, 2020 | by FozzTexx

Booting a 486 From Floppy with the Most Up-to-Date Stable Linux Kernel

Booting a 486 From Floppy with the Most Up-to-Date Stable Linux Kernel

▶

Back in August of 2019 I had a Sing-Along Week contest on RetroBattlestations (https://www.reddit.com/r/retrobattlestations/comments/cr6x9a/contest_singalong_week_until_aug_25/) with the challenge of playing a MIDI file synchronized to a speech synthesizer singing the lyrics. In my stash of cards I found an MQX-32M MIDI card (Roland MPU-401 clone) which I wanted to use for converting MIDI signals to interface to an Apple II+ with an Echo II speech synthesizer, but ended up using a laptop with a USB MIDI interface. The Pizza Week contest (https://www.reddit.com/r/retrobattlestations/comments/fz6is8/contest_pizza_week_through_april_19th/) made me more interested in finding a 486 in a pizza box case and so my search began for a suitable case.

Searching eBay wasn't easy because there's not really a good search term that would find the kind of case I wanted. I didn't bother trying to make a saved search since this project wasn't a high priority. Instead every once in a while something would inspire to go hit eBay and try some other search terms. A couple of weeks ago I spotted this 486 and made a guess of the size based on the size of the 5.25" bay. It's fairly slim, it has a riser card to turn the cards sideways and the case *is* shorter than an ISA card, but I don't feel like it's slim enough to call it a pizza box. It's close though!

The motherboard is a TMC PAT48PG4 (https://stason.org/TULARC/pc/motherboards/T/TMC-RESEARCH-CORPORATION-486-PAT48PG4-VER-1-2A.html) and it came with 32MB of RAM (technically 36MB, it had four 30 pin SIMMs in it too) and 4 cards: VGA, SoundBlaster, modem, and multi-function IO. The seller had tested it and was sold as non-working, won't POST, although he said the cards all worked. I mostly wanted it for the case so I wasn't concerned. Of course when I got it I had to see if the motherboard was really dead. It didn't work for me either but then I discovered if I hit the reset button it would give me some error beeps. I tried another power supply and it came up after hitting reset! It still doesn't cold boot when powered on, not sure if there's a bad cap or if the very slight battery damage (battery was removed before I got it) is preventing the reset line from being pulled after power on. Nonetheless it was working enough that I needed to explore its capabilities more.

My ultimate goal is to get a *very recent* Linux distro and Python 3 installed to a "large" hard drive, but being that this is a 486 installing a current Linux distro isn't trivial. The only Linux distros that still support installing from floppy media are quite dated. Normally I would bypass a floppy install entirely and just boot an older computer via PXE and then install over the network. I thought I could put iPXE on a floppy (https://ipxe.org/appnote/buildtargets) and stick in an ISA NIC but iPXE just hangs without any error messages right after it's loaded from floppy.

The BIOS is ancient and struggles with handling the "large" hard drive I hooked up. The drive is 8.45GB and the BIOS can only see it as an 8.0GB drive. Enabling LBA in the BIOS causes the computer to hang during POST after it detects the drive. Win98 fdisk insists the drive is only 504MB. FreeDOS can't see the drive at all. I tried sticking in a NIC with XTIDE ROM on the socket and XTIDE also insists there's no drive connected.

Since I wanted to see how Linux would detect the drive that meant I needed to find a way to boot Linux. After a bit of googling I discovered the make tinyconfig option (http://tiny.wiki.kernel.org/) which makes a very small (but useless) kernel, small enough to fit on a floppy. I enabled a couple of other options (https://weeraman.com/building-a-tiny-linux-kernel-8c07579ae79d), found a small enough initramfs, and was able to get it to boot on the 486. And as expected Linux has no problem with seeing that the drive is connected and the drive's full capacity.

Next step is to actually get Linux installed to the hard drive. I'd rather not roll my own distro but maybe I'll have to. Another possibility is to boot Linux from floppy and then download a kernel and initrd from a current distro and kexec over to it. But that feels to me like reinventing iPXE.

Compiling the Linux Kernel From Source

A quick rundown of the steps to build the floppy image:

- git clone https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git
    - v5.8-rc2-1-g625d3449788f at the time of writing
- make ARCH=x86 tinyconfig
- make ARCH=x86 menuconfig
    - Set processor to 486: CONFIG_M486=y
        - Processor type and features > Processor family > 486
    - Enable tty: CONFIG_TTY=y
        - Device Drivers > Character devices > Enable TTY
    - Enable printk: CONFIG_PRINTK=y
        - General Setup > Configure standard kernel features (expert users) > Enable support for printk
    - Enable initramfs: CONFIG_INITRAMFS_COMPRESSION_GZIP=y
        - General Setup > Initial RAM filesystem and RAM disk (initramfs/initrd) support > Support initial ramdisk/ramfs compressed using gzip
    - Enable ELF: CONFIG_BINFMT_ELF=y
        - Executable file formats > Kernel support for ELF binaries
- make ARCH=x86 bzImage

You'll need a rootfs, get system-image-486 from Aboriginal Linux (http://landley.net/aboriginal/downloads/binaries/), extract rootfs.cpio.gz

Test boot with qemu:

```
qemu-system-i386 -kernel arch/x86/boot/bzImage -initrd ../system-image-486/rootfs.cpio.gz
```

Create a blank floppy image:

```
dd if=/dev/zero of=linux-boot.img bs=1k count=1440
mkdosfs linux-boot.img
syslinux --install linux-boot.img
mount -o loop linux-boot.img /mnt
cp arch/x86/boot/bzImage /mnt
cp rootfs.cpio.gz /mnt
```

Create /mnt/syslinux.cfg:

```
DEFAULT linux
LABEL linux
  SAY Now booting the kernel from SYSLINUX...
  KERNEL bzImage
  APPEND initrd=rootfs.cpio.gz
```
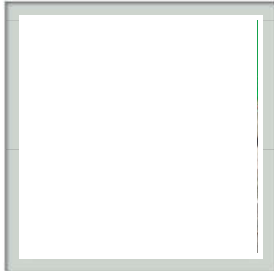
Write the image to a floppy:

```
umount /mnt
fdformat /dev/fd0
ddrescue -f -D linux-boot.img /dev/fd0
```

Join The Discussion

Have something to say? Go ahead and enter it here →

Submit | Lost Password?

Search

Most Recent Posts

Booting a 486 From Floppy with the Most Up-to-Date Stable Linux Kernel (/a.cl/283/booting-a-486-from-floppy-with-the-most-up-to-date-stable-linux-kernel)

by FozzTexx

October Brings Horror (/a.cl/282/october-brings-horror)

by FozzTexx

Retro Computer Day was a Success! Mostly (/a.cl/281/retro-computer-day-was-a-success-mostly)

by FozzTexx

A Retro Computer Day Experiment (/a.cl/280/a-retro-computer-day-experiment)

by FozzTexx

A Three Finger Salute for the Apple][Plus (/a.cl/279/a-three-finger-salute-for-the-apple--plus)

by FozzTexx

🐦 **Recent Tweets**

FozzTexx Retweeted
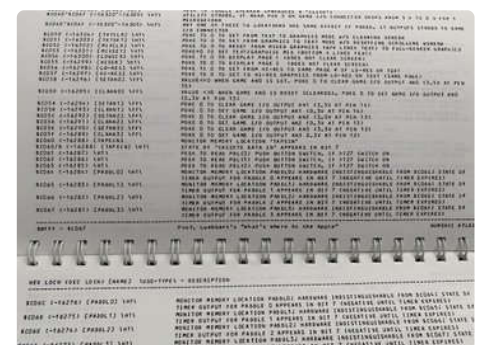
FozzTexx
@FozzTexx

Replying to @hackaday

Come on hackaday, you can do better than this. The Linux kernel hasn't supported the 386 since 2012. Why are you saying that it still does? It's easy to check too, just run a menuconfig and go see that 386 isn't there anymore.

1h

FozzTexx
@FozzTexx

Replying to @FozzTexx

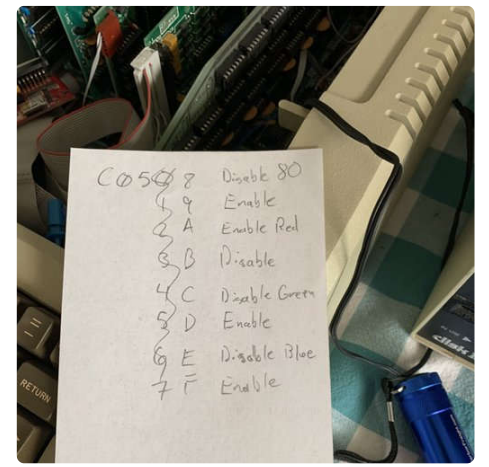The paddle address all use POKE but the R/W line isn't connected on the DVM so I'm not sure how they got away with using that I/O space.

2h

FozzTexx
@FozzTexx

Replying to @FozzTexx

Whoops, correct numbers.

2h
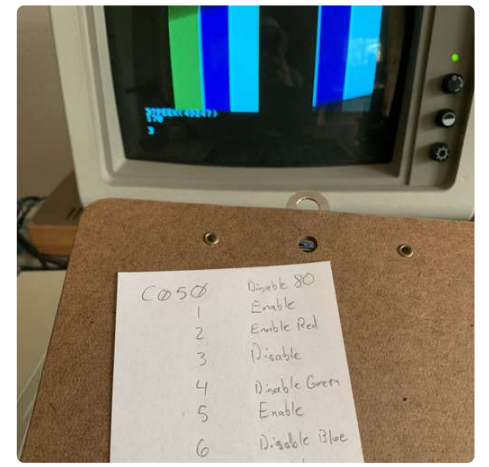
FozzTexx
@FozzTexx

Replying to @FozzTexx

And now they are documented! Yes the red gun really is backwards from all the others.

2h

FozzTexx
@FozzTexx

Replying to @FozzTexx

Found the registers! They used the paddle I/O space.

2h

(https://twitter.com/FozzTexx)

Most Recent Articles