DOSContainer design documentation

Bas v.d. Wiel

April 14, 2024

Chapter 1

Introduction

DOSContainer was born out of a dissatisfaction with the way game packs were created for computer platforms on the MiSTerFPGA platform. Many such collections focus on turning computers into consoles, which I felt took away from the intricacies of these old systems. Having quite a good memory of my own DOS days, I decided to codify all of that knowledge into a tool that will run on modern platforms.

I started out with a proof of concept using the Bash shell on a Linux system, because that's also the operating system that runs on the MiSTer. After some interesting weeks I actually finished the project and it still lives on GitLab at the time of this writing. You can use it go generate disk images with it, and they will even work! Maintenance on the script itself quickly turned into a massive nightmare, as did the long list of external dependencies.

This manual covers the Rust version of DOSContainer, which was conceived after I had proven to myself that the idea behind DOSContainer was feasible. Coding the tool in Rust would fix a few major annoyances for me, least of which was the long list of dependencies. By default, Rust spits out statically compiled binaries. This means that you get a single executable file, tailored natively to the platform it was built for, that will just work without any external libraries or dependencies needed. So no DLL hell, not even on Windows!

Another aspect that greatly improved was control. Using Rust I get to control every single bit that gets written to the disk image, as well as a whole set of ready-made libraries that handle things like file downloading, checksums, parsing YAML etc. In short, Rust gives me quality control. It also provides me with a built-in mechanism for testing my code, so I can prove that things work as intended as the codebase sprawls.

The manual you are reading now comes with pre-alpha releases of the tool. That means that the core functionality is not yet completed. Releases start at version 0.0.3 because the first two minors were taken by the Bash version. I'm not developing those anymore, so the 0.0.3 version will be what gets released as the first real alpha.

The feature set on the alpha release will be:

- Create valid VHD-format disk image files. Not even Windows does that!
- Partition and format bootable disks identical to IBM PC-DOS 2.00.
- Install all of IBM PC-DOS 2.00 on the image optionally.
- Install Alley Cat from YAML.

At the time of this writing DOSContainer ticks the first box and is almost there for the second. Once I can publish with a running version of Alley Cat on the MiSTer's PCXT core, I'll release DOSContainer and start working on collecting and packaging DOS games from the early 1980's.

Part I

User manual

Chapter 2

DOSContainer command reference

DOSContainer in its current state is hardly functional at all. Regardless of that, this chapter outlines the command line interface for this program. Depending on the operating system on your computer, you need to use a different binary file. On Windows this file is caled `doscontainer.exe` while on any other platform it's just `doscontainer`. Whenever you see a reference to `doscontainer` in a command line, you should substitute the .exe version if you are on Windows. Functionally all versions are 100% identical.

On Linux and other UNIX-like systems such as macOS an example of the command would look like this:

```
$doscontainer build alleycat.yaml
```

..and on Windows this same example would be something like:

```
C:\Progra~1\DOSK8S\doscontainer.exe build alleycat.yaml
```

2.1 Running commands

DOSContainer's commands are invoked by adding a single word to the invocation of the binary itself, as illustrated above: the word `build` is the command in these examples. DOSContainer in its current pre-alpha form knows three commands:

analyze
: Print debugging information on any VHD-format disk image.

build
: Constructs a VHD image from an input file in YAML format.

download
: Downloads a file from any HTTPS URL you feed it. Not very useful right now.

The `analyze` command needs the filename of a VHD-format disk image as a parameter, like so:

```
$doscontainer analyze alleycat.vhd
```

Given a valid VHD-file, it will output all sorts of debugging information on the VHD container format itself, the partition table, the filesystem and the files that are found on the disk image. The reason why this command exists at all, is that it was useful during development. It was left in the tool because tearing it out would take effort, and it could in fact be useful to others as well.

The `build` command is the bread and butter of the DOSContainer tool. The command itself is very simple because it just takes a single YAML file. It generates a VHD-file in the same directory from where you invoked the DOSContainer program. So assuming DOSContainer is in your path somewhere like installed in `/usr/local/bin` on Linux, you could invoke it from anywhere. Let's say you have a `games` folder in your home directory from which you run DOSContainer:

```
$doscontainer build alleycat.yaml
```

or for Windows:

```
C:\Users\JohnDoe\doscontainer.exe build alleycat.yaml
```

..would yield a file named `alleycat.vhd` right in your own home directory. If you happen to be working right on the MiSTer itself, you could run this from `/media/fat/games/PCXT` and you'd be able to mount the new file and use it directly.

Part II

Background documentation

Chapter 3

Disk drives and virtual images

DOSContainer concerns itself with disk image files. These are modern, virtual recreations of what used to be physical hard drives. For that reason I'm going to expand a bit on what a physical hard drive was like in the 1980's. It helps to understand what you're dealing with before trying to use it, right?

Physical hard drives all have a geometry that can be expressed as a triplet consisting of Cylinders, Heads and Sectors-per-track. For a mental picture of how this works, imagine a hard drive to be like a vinyl record player. Instead of a single needle that traces the groove on a single side of a record, we deal with a stack of magnetic disks that each have the equivalent of a needle on both sides. A typical hard disk in the IBM XT era would have two rotating platters on a spindle. Each platter would have a read/write head on both sides, adding up to a total of 4 heads inside the assembly.

The mechanism is capable of moving the heads between the center and the outer edge of the platters in steps. The number of discrete steps that the mechanism is capable of, adds up to the number of cylinders. A cylinder describes the concentric circles that the heads would trace when they are set to a specific cylinder value. A single such circle is called a track, which in turn is divided into sectors. On our XT-era disk, the number of sectors per track would usually be 17. Coupled with the fact that a typical sector stores 512 bytes of data, we can figure out the capacity of our drive as the multiplication of the number of cylinders, heads, sectors per track and the sector size. Any triplet of Cylinder, Head, Sector also uniquely identifies a single sector on the drive assembly.

In this day and age of multi-terabyte hard disks, the CHS geometry is no longer used in practice. Nowadays our computers treat any hard drive like a big pile of sectors that they simply address from 0 to however many there are. The ATA-6 standard uses a 48-bit value for the sector number, so we'll be good until single drives reach the 144 PetaByte limit. You want to read data from a sector? Just plug its number into the drive, issue a read command and the drive does the rest. In reality the drive is still a stack of platters, heads and sectors. We just don't address them in this manner anymore.

DOSContainer aims to take the hassle out of the ancient dark art of perpar-

ing a hard disk for use with a 1980's era PC. The main purpose for this is to be able to generate bootable hard disk images for use with emulators such as PCEm, 86Box or the MiSTerFPGA AO486 and PCXT cores. These all aim to reproduce a period-appropriate facsimile of an IBM-compatible PC complete with their peripherals and clunky hard disk support of the day. It really doesn't help much that you can just plug in a VHD image and have a virtual hard drive to play with if you don't know where to even begin.

DOSContainer not only delivers a fully prepped hard disk image that is indistinguishable from a byte-for-byte copy from an actual old PC. It also gives you the needed numbers to punch into the emulated PC's BIOS so that things will actually work.

Part III

Design

Chapter 4

Operating Systems

4.1 IBM PC-DOS 2.00

The first version to support hard drives at all. Only supports FAT12 filesystems. When using FDISK to generate a single partition to fill the whole drive and make it bootable, IBM puts it in the 4th slot in the partition table instead of in the first. Functionally this makes no difference but for accuracy we should mirror this behavior. Max partition size is 32MB. Drive size is limited by the CHS geometry being restricted to 17 sectors per track for the PC/XT environment. The XTIDE BIOS can probably handle much more but that's not period appropriate.