Reading a 4th Edition Research Unix tape image on Plan 9

Anthony Sorace
a.9srv.net

Strand 1 Technologies

ABSTRACT

A recently-discovered magnetic tape representing a full system image of the 4th Edition of Research Unix was succesfully decoded. The end result was a tar file, easily readable on a modern system, but a pair of intermediate results are also interesting: a file in the tp format, and another representing the disk filesystem from the rk05 disk. These match the formats of a pair of Plan 9's *tapefs*(4) tools: fs/tpfs can read most of the initial bootloader and fs/v6fs can read the rk05 image that represents most of the tape.

Introduction

A magnetic tape containing what was believed to be a full system image of the 4th Edition of Research Unix was recently discovered in storage at the University of Utah.[1] Prior to this discovery, no full system image of this version was known to exist. The tape was driven to the Computer History Museum for reading and archiving.

1. Reading the tape

At the Computer History Museum, they used the utility readtape[2] to turn the raw signal on the magnetic tape into a series of files, primarially a recording of the analog signals (in a format specific to readtape) and a *tap* file representing those contents in a more manageable form. That file was then converted into several others, for easier processing and reading.

2. Extracted Files

The process of reading the tape using readtape produced several direct outputs, available from the Internet Archive.[3] The most notable of these is analog.tap, which is used to generate all the others we are interested in.

2.1. SIMH tap file

While reading the analog waveform from the tape, readtape also generated a file in SIMH *tap* format.[4] I had initially hoped to use fs/tapfs to read this file directly, but this was unsuccessful, due to naming confusion on my part. Despite the fact that SIMH is frequently used to simulate or work with old unix systems, the *tap* format in question here is not the same as the *tap* format used by some early Research Unix systems, but instead a totally unrelated format developed by SIMH for use with its emulated tapes

---

[1] https://discuss.systems/@ricci/115504720054699983
[2] https://github.com/LenShustek/readtape
[3] https://archive.org/details/utah_unix_v4_raw
[4] https://simh.trailing-edge.com/docs/simh_magtape.pdf

and disks. This unfortunate name collision confusion cost more time than was reasonable.

2.2. "raw" file

Fortunately, someone else was able to convert the SIMH *tap* file to a "raw" format[u] and make it available for download; that format is one the Plan9 tools can start to work with.

My first attempt was to pass the whole archive to fs/tapfs, but this gave three problems. First was a "cksum failure" message, although only a single instance, unlike when trying to parse the SIMH *tap* file. Second, the file listing and contents in /n/tapefs looked, at first glance, plausible for a boot image, but was incorrect.

```
:; fs/tapfs  analog.tap
cksum  failure
:; ll  /n/tapefs
--rwxrwxrwx   M 82  129  100088  262  Dec  30   1972  dldr
--rwxrwxrwx   M 82  129  100088  262  Dec  30   1972  dtf
--rwxrwxrwx   M 82  129  100088  259  Dec  30   1972  list
--rwxrwxrwx   M 82  129  100088  259  Dec  30   1972  mboot
--rwxrwxrwx   M 82  129  100088  262  Dec  30   1972  mcopy
--rwxrwxrwx   M 82  129  100088  262  Dec  30   1972  rkf
--rwxrwxrwx   M 82  129  100088  259  Dec  30   1972  tboot
--rwxrwxrwx   M 82  129  100088  262  Dec  30   1972  uboot
```

The list of files is correct, but the permissions, user, and group all seem suspicious, while the file size is very concerning. And while the sizes listed match the actual contents, those contents themsselves seemed wrong for a boot image: each contained only several lines of assembly instructions. A representative example:

u https://adhd.irenes.space/@ireneista/statuses/01KCWP28PVTNWQD67QMXTV7Q3F

```
:; cat /n/tapefs/dldr

        bne     joe
        mov     $_done,(r4)+
        rts     pc

strun:
        cmp     r0,$'
        bne     joe
        mov     $_run,(r4)+
        rts     pc

stdisp:
        mov     $_sdisp,(r4)+
        jsr     pc,stprint
        mov     $_fdisp,(r4)+
        rts     pc

stprint:
        jsr     pc,stpromp
        mov     $_nline,(r4)+
        rts     pc

stpromp:
        jsr     pc,skip
        cmp     r0,$'
        jeq     retpc
:;
```

Finally, none of this either added up to the overall tape size or represented anything like a functional system.

Comparing a hex dump of the raw image, produced with xd -c, to the format described in /sys/src/cmd/tapefs/tapfs made it clear that there was a mismatch. While *tapefs*(4) suggests that *tap* was the format for "tapes from the pre-Fifth Edition era", it also describs a follow-on format, *tp*, used for "tapes from the Fifth through Seventh Edition research Unix systems." Most Research Unix releases were not as rigidly defined as we tend to think of software releases today, reflecting more a "point in time" snapshot of an evolving system. Having two similar formats around the transition from 4th Edition to 5th Edition Unix would make it not surprising to find artifacts crossing that imagined line.

Indeed, fs/tpfs handles the archive properly. We see the same files listed, but with much more sensible sizes, permissions, and contents:

```
:; fs/tpfs analog.raw
0 bad checksums, 496 good
:; ll /n/tapefs
--rwxrwxrwx  M 89 6 1  228 Oct 24  1973 dldr
--rwxrwxrwx  M 89 6 1 2406 Oct 24  1973 dtf
--rwxrwxrwx  M 89 3 1   74 Oct 25  1973 list
--rwxrwxrwx  M 89 3 1  492 Oct 25  1973 mboot
--rwxrwxrwx  M 89 6 1  432 Nov 11  1973 mcopy
--rwxrwxrwx  M 89 6 1   80 Nov 11  1973 rkf
--rwxrwxrwx  M 89 3 1  450 Oct 25  1973 tboot
--rwxrwxrwx  M 89 6 1  512 Oct 25  1973 uboot
```

This resolves the first two of our problems above: we get no checksum errors, and we get sensible contents. The overall size of the extracted data, however, is still much larger than we're seeing here, and this is certainly not a full Unix system.

2.3. bootloader

The first 38400 bytes of the raw file represent the loader image. Extracting that as a separate file and passing that to fs/tpfs produces exactly the same results as using the *raw* file directly. Aside from being much easier to work with at 38400 bytes instead of ~2.5MB, its main function is to demarcate where the *tp* archive ends in analog.raw. We can then extract the remainder separately.

2.4. RK05 Disk Image

The final, and most significant, artifact is disk.rk.[v] This represents the RK05 diskpack which contained the actual file system for the installed system; it is the same as analog.raw with the bootloader section removed. In another example of blurring the edges of firm release boundaries, Plan9 s fs/v6fs, intended for reading disk file systems used on 5th and 6th Edition systems, correctly interprets this image. At this point, we can freely explore the 4th Edition file system:

```
:; fs/v6fs disk.rk
:; ll /n/tapefs
d-rwxr-xr-x  M 76 3 1    944 Jun 12 1974 bin
d-rwxr-xr-x  M 76 3 1     64 Jun 10 1974 dev
d-rwxr-xr-x  M 76 3 1    256 Jun 12 1974 etc
d-rwxr-xr-x  M 76 3 1    256 Jun 12 1974 lib
d-rwxr-xr-x  M 76 3 1     32 Jun 10 1974 mnt
d-rwxrwxrwx  M 76 3 1     32 Jun 10 1974 tmp
--rw-r--r--  M 76 3 1  27624 Jun 12 1974 unix
d-rwxr-xr-x  M 76 3 1    240 Jun 12 1974 usr
:;
:; sed 4q /n/tapefs/usr/sys/ken/main.c
#
/*
 *      Copyright  1973 Bell Telephone  Laboratories  Inc
 */
:;
```

3. Conclusion

At this point, we can use Plan9 s tools to explore both the boot section of the tape and the full Unix image. We rely on an external tool to convert the SIMH *tap* file into a format reflecting what we d expect from the tape, but can then progress from there. It is tempting to suggest creating a new member of the *tapefs* (4) family to directly interpret the SIMH *tap* format, given its use for all manner of images in SIMH, but that remains an unexplored project.

---

[v] http://squoze.net/UNIX/v4/disk.rk