Unearthing The PDP-15’s Advanced Monitor System

Bob Supnik, 15-Jul-01

Summary

On 13-May-2001, the PDP-15’s Advanced Monitor operating system was successfully booted on the SIMH (history simulator) emulation system. The discoveries and events leading up to this milestone illustrate the vital role that the Internet plays in enabling computer history enthusiasts world-wide to work together for computing preservation.

Background

DEC’s 18b computing family (the PDP-1, PDP-4, PDP-7, PDP-9, and PDP-15) was of significant historic interest. The PDP-1 was DEC’s first computer, and the base for the first “video game” (SpaceWar). The PDP-7 ran DEC’s first mass storage operating system (DECsys), and the first version of UNIX. Despite their historic importance, the 18b family was a limited financial success and was the first of DEC’s computing families to go out of production. Development ceased in the mid 1970’s, and the last 18b computer was produced in 1979. As a result, functioning 18b systems are rare, and many of the key software systems (DECsys, UNIX V1) have been lost. At the beginning of 2000, none of the later 18b software systems was available.

The Computer History Project is an Internet-based collection of computer history enthusiasts. Its goal is to recreate historic systems via emulation and to collect and transcribe software that ran on those historic systems.

Initial Documentation

The first step in preservation and recreation is to collect documentation about the target systems. For the 18b family, this was very challenging. The User Manuals for the early systems (PDP-1, PDP-4, PDP-7) are inaccurate, misleading, and sometimes just plain wrong. Peripheral and option documentation is sketchy or non-existent. The Maintenance Manuals and logic prints are really the only authoritative sources. For all these systems, documentation is rare, as fewer than 200 were produced in total.

Documentation for the later systems (PDP-9, PDP-15) is more plentiful but not necessarily better. The PDP-15’s User Manual, in particular the First Edition, is notorious for its inaccuracies. Again, the Maintenance Manuals, logic prints, and where available, diagnostics are the best source of accurate information.

The starting point for the documentation search was the DEC Archive, while it still existed. The Archive contained these documents:

- PDP-1 Handbook (second edition)
- PDP-1 I/O Manual
- PDP-1 Maintenance Manual
- PDP-4 Handbook
- PDP-7 User Manual (preliminary)
- PDP-7 Maintenance Manual (including logic prints)
- PDP-9 User Manual
- PDP-15 User Manual (first edition)

This sufficed to write a pair of preliminary 18b simulators, one for the PDP-1, the other for the rest.

Initial Software

With first pass simulators available, the next step was to collect software for testing and demonstration purposes. This proved to be as difficult as finding the documentation. For the PDP-1, the sources to Lisp and SpaceWar had been published. For the later 18b systems, no software was available on public sources. The restoration of Lisp illustrates the importance of the Internet in historic computer salvage. The source came from the Internet (transcribed by Gordon Greene); the PDP-1 macro assembler was derived from a PDP-8 cross-assembler (by Gary Messenbrink) found on the Internet; and the final debug was done remotely (by Paul McJones).

The initial PDP-7 software came from an old PDP-10 backup tape found by Dave Waks. Again, the Internet played a vital role in salvaging the code. The tape was transcribed on a 7-track tape drive by Paul Pierce. He shipped the raw bits over the Internet to Tim Litt, who decoded the PDP-10 backup formats. Tim in turn sent the transcribed bits to the author for debugging.

From 1998 to 2000, nothing was found for the PDP-9 or PDP-15. In 2000, Al Kossow found and transcribed a set of paper tapes from the McMaster physics lab. Among these tapes were some diagnostics and several copies of FOCAL for the PDP-15. The diagnostics sufficed to debug the PDP-15 simulator and get FOCAL running. At the same time, Dave Gesswein found a set of DECtapes for the Advanced Software System.

DECtapes

To revive the Advanced Monitor System, the SIMH team would have to find a way of dealing with DECtapes. In the 60's, DECtapes were the principal form of mass storage on DEC minicomputers; the only affordable alternative was fixed-head disk, which was non-removable. DECtapes posed multiple challenges:

- DECtapes must be simulated with great precision. DECtape software was timing-dependent; in addition, it relied on the ability to examine individual words as they were read into or written from memory.
- DECtapes are difficult to transcribe. The only way to transcribe a DECtape is on a real DECtape drive, which is a complex mechanical device and difficult to maintain. In addition, the DECtape format for the PDP-9/15 requires special handling on the PDP-8 and PDP-11, the systems most likely to have survived and to have working drives.

As a prerequisite to implementing DECtapes, the unresolved issues in the 18b simulators needed to be cleared up. Via the Internet, Al Kossow, Max Burnet, and Dave Gesswein provided additional critical documents:

- PDP-1 Handbook (third edition)
- PDP-4 Maintenance Manual
- PDP-9 Maintenance Manual
- PDP-9 Schematics
- PDP-15 User Manual (sixth edition)
- KE09A (EAE) Reference Manual
- KF09A (API) Reference Manual
- TC02 (DECtape) Instruction Manual
- RF15 (DECdisk) Maintenance Manual
- PDP-9 Advanced Software Systems Monitors Manual
- PDP-15 Foreground/Background Reference Manual

These sufficed to answer the outstanding questions.

To cope with the expectations of DECtape software, the SIMH DECtape emulator implemented a word-by-word, time-based model that provided full simulation of acceleration, deceleration, and tape turn-around. By source code count, it was the most complicated peripheral simulator in SIMH. The simulator successfully ran the DECtape exercisers for the PDP-9 before any attempt was made to run DECtape-based software.

Dave Gesswein was able to transcribe PDP-9/15 DECtapes, including both the Keyboard Monitor System and the Foreground/Background System, using a PDP-8/E with TD8-E controller. The TD8-E was a highly simplified version of a traditional DECtape controller. It read every frame off the tape and left the decoding of the format to software. Thus, it was the ideal device for reading PDP-9/15 DECtapes; it disregarded the format differences and delivered raw bits from the tape for software to decode.

The Keyboard Monitor System

All the prerequisites for reviving the PDP-15's DECtape operating systems – documentation, simulator, transcribed DECtapes – seemed to be at hand. There was one last problem. Unlike contemporary systems on the PDP-8 and PDP-11, the Advanced Software System did not bootstrap by reading DECtape block 0 into memory and jumping to it. Instead, it required a bootstrap paper tape that was loaded by the hardware read-in facility. For more than a year, all attempts to find this paper tape failed. Appeals on the Internet brought no response. The various private collectors on the Internet, and the Computer History Center, drew a blank.

In May 2001, an email exchange with Hans Pufal of Grenoble France about the PDP-10 revealed that Hans had a paper-tape bootstrap for the PDP-9. He had no direct way to transcribe it. Ingeniously, he scanned the paper tape in sections on a standard optical scanner, wrote a program to decode the pattern of holes, and verified the results by hand. He sent the results to the author on May 10, 2001, and I immediately tried it with Dave Gesswein's DECtape images.

Unfortunately, it didn't work. One issue was a lingering bug in the DECtape simulator. A second was that the Foreground/Background System required the Automatic Priority Interrupt option (API), which hadn't been implemented. But the major hurdles were undocumented software changes that occurred between the PDP-9 and PDP-15. As I wrote on May 11:

> I tried bootstrapping the ADSS [Keyboard Monitor] as well as the F/B monitors. For the former, the starting address is a SKP HLT. For the later, it's all 0's. The very next set of instruction picks up the starting address, masks the address with 070000, and performs other manipulations - clearly reconstructing the bootstrap address, provided that the BOOTSTRAP EXITS WITH A JMS RATHER THAN A JMP. So I changed the exit instruction (at 17745) to be JMS I 17755, and suddenly I am a lot further - not running mind you, but further. ADSS, in particular, prints out a nice error message IOPS03 021400, which indicates that the basic I/O system is alive if not well. (Somewhere I have documentation on its error messages.) I think F/B requires the API option, which isn't implemented.
>
> So this is tremendous progress! I am fairly sure that the boot process is:
>
> 1. read 36(8) DECtape blocks, starting at block 0, into memory, starting at location 100
> 2. use location 105 of the loaded image as the starting vector
> 3. if booting pdp-9 software, jump to it; for the -15, jms to it

The next day, the error message was traced to a customization in the interrupt skip chain:

> With Hans' bootstrap tape, modified (as I think) for the PDP-15 (exit instruction is JMS rather than JMP), I was getting an IOPS03 error - invalid interrupt. Tracing through the interrupt skip chain, I got to:
>
> IOT 1041
> JMP* handler

> I have no idea what IOT 1041 does; it's not any standard DEC device, and it is backwards from normal I/O tests, which are always:
>
> ```
> PSF
> SKP
> JMP* handler
> ```
>
> So it must have been a custom device SYSGEN'd into this version of the monitor for the installation. I nop'd the tests out, and the tape got to the keyboard monitor prompt! I don't know how the keyboard monitor works, so I typed in PIP, that resulted in
>
> ```
> .SYSLD 1
> IOPS03 ...
> ```
>
> so there's still more debug to do, but this is the first sign of life out of an 18b operating system!

The next day, this bug was traced to another undocumented change between the PDP-9 and PDP-15 bootstraps:

> Second difference in bootstrap for the 15 vs the 9: the load image is one block longer. The 9 bootstrap loads 17000(8) words from 100 to 17100. The -15 monitor is 17400(8) words long, from 100 to 17500. The failure to load the last 400 words accounts for all the crashes on keyboard monitor commands. I can now take a directory, print the information message, print the SCOM region, etc.
>
> I <still> can't load or run a system program, so there's more work to be done. One possibility is that the system is gen'd for more memory than I am allowing.

And that indeed proved to be the case. Loading the bootstrap in upper memory allowed the Keyboard Monitor System to run correctly. By running SYSGEN, references to custom devices were eliminated, creating a clean DECtape image.

From Keyboard Monitor To Foreground/Background

With the Keyboard Monitor System running, the next challenge was to bring up the Foreground/Background System. This required additional hardware: memory protection, automatic priority interrupts (API), and (although I didn’t realize it) a second terminal. All of these had their issues:

- Memory protection, though implemented, had never been tested and contained serious bugs.
- API required intrusive changes throughout the CPU simulator.
- SIMH had no capability for multiple terminals.

Fortunately, the PDP-9/15’s API closely resembled the PDP-10’s, allowing the latter to be used as a model for the implementation. The second terminal problem was solved by a kludge that allowed multiple terminals to access the controlling window and keyboard on a sequential, rather than a simultaneous, basis. With these changes, the Foreground/Background System was successfully run on May 28.

Next Steps?

The Keyboard Monitor and Foreground/Background Monitor do not exhaust the variety of environments available for the PDP-15. The disk-based operating systems -- DOS-15, RSX-PLUS-III, and MUMPS -- represent even higher levels of capability and would be of great interest historically. Unfortunately, no copies have been found. DEC junked its PDP-15 media archive at the end of the 80's, when the last systems went off contract. DECdisks and RP02's were huge and have all ended up on the scrap heap. Unless there is a complete save set on magnetic tape somewhere, the disk-based operating systems are irretrievably lost.

Acknowledgements

The revival of the Advanced Monitor Systems demonstrates the critical role of the Internet in creating a virtual community of computer history enthusiasts. The project would not have succeeded without the help of individuals whom I know mostly or exclusively through the Internet. In addition, the Internet allowed for rapid interchange of documents, software images, and folklore.

I am particularly indebted to:

- Max Burnet (Australia), for hardware documentation on the 18b systems
- Al Kossow (California), for hardware and software documentation on the 18b systems, as well as paper tape images
- Dave Gesswein (Canada), for hardware and software documentation on the 18b systems, as well as the DECtape images
- Hans Pufal (France), for finding the critical missing bootstrap tape, and transcribing it without a paper tape reader