VTserver: Installing V7 UNIX with No Tape Drive

Warren Toomey

wkt@cs.adfa.oz.au

Version 1.1, 3rd February 1998

1 Caveat

Before you even start looking at this software, there are some caveats which go with it.

1. The VTserver software should be considered in alpha release. Your feedback is most definitely requested, to fix any bugs and to improve the software.
2. I have never owned or used a real PDP-11. The VTserver software was developed using a number of PDP-11 emulators. For this reason, there may be hardware-specific bugs in the software which were not found using the emulators. As well, the terminology in this manual may not be correct PDP-11 speak. Again, please send me your suggestions and improvements!

2 Introduction

Several members of the PUPS mailing list (see http://minnie.cs.adfa.oz.au/PUPS/) have wanted a way of installing UNIX onto a PDP-11 which has no tape drive. One solution, if the PDP-11 has software already installed (RT-11, RSX etc.) is to leverage from the installed software to download and write UNIX onto a spare disk. This approach is not covered here, and someone who has done this should be encouraged to write it up.

The approach here has been to modify the initial UNIX tape loading software so that bits can be downloaded from a nearby computer over a serial cable. The modified boot straps thus perceive a *Virtual Tape* drive; hence the name of the software.

The VTserver software comes as two components: a set of PDP-11 software which acts as the client, and the server which is hosted on a second machine, running Unix.

The two computers are connected via an RS-232 null modem with hardware handshaking. A KL/DL-11 unit on the PDP-11 is used as the serial port: the highest transfer rate is 9,600 bps. Although this is a way to install UNIX on a PDP-11, its a pretty slow way.

3 Manifest

The VTserver software is available from ftp://minnie.cs.adfa.oz.au/pub/PDP-11/Vtserver as the file `vtserver1.0.tar.gz`. You should have received the following files as part of the VTserver software:

README: A quick overview of the software.

vtreadme.txt: This document in plain ASCII. There may be an equivalent PostScript file, **vtreadme.ps**.

v7_setup.txt: The original Bell Labs description on setting up 7th Edition UNIX. There may be an equivalent PostScript file, **v7_setup.ps**.

unix_license.txt: SCO licence covering 7th Edition UNIX binary distribution.

vtserver.c: The source code to the server program.

.vtrc: The server's configuration file.

vtboot.pdp: The machine code to bootstrap using the `vtserver`.

boot: Second stage bootstrap.

cat: Standalone cat(1) program.

tape_contents: A brief description of the fictitious tape's contents.

mkfs: Standalone mkfs(1) program.

icheck: Standalone icheck(1) program.

restor: Standalone restor(1) program.

Additional files, which are described in Section 17, are also available in this directory. You will need at least `v7rootdump`. This comes as a Gzip'd file, and you will need to unzip it. You probably want to get all of the .Z files from the FTP site above. Don't uncompress them, as they can be loaded onto the PDP-11 'as is'.

4 Required Seventh Edition Manuals

Before you being installing 7th Edition UNIX, you should print out the original installation instructions. These are contained in the files `v7_setup.txt` and `v7_setup.ps`. You will need to reference these instructions when you get to Section 10.

It is a good idea to have the other Seventh Edition UNIX manuals on-hand as you start using the system. You can install them onto your PDP-11 (see Section 17), or you can access them over the web at

http://www.de.freebsd.org/de/cgi/man.cgi?manpath=Unix+Seventh+Edition

5 Compiling `vtserver`

The first thing to do is to compile `vtserver.c`:

```
$ cc -o vtserver vtserver.c
```

The code is pretty vanilla ANSI C. The only system-specific section is that on setting the serial port to raw mode. I have used the Posix *termios* system calls, and you may need to port this if your system isn't Posix-compatible. Please send patches back to me for this.

The file `.vtrc` is the server's configuration file. Lines beginning with hashes are ignored. The first (non-hashed) line names the serial device. Remaining lines name the fictitious tape's records, and should be *boot*, *cat*, *tape_contents*, *mkfs*, *icheck*, *restor* and *v7rootdump*, in that order.

You will most likely need to change the name of the serial port device.

6 Connecting the PDP-11 to the `vtserver`

The next KL-11 unit after the console (i.e the one at vector 0176500) needs to be connected via RS-232 to the serial port on the Unix server where you compiled `vtserver.c`. Ok, I have no experience here as I don't have a real PDP-11. I'd strongly recommend using a null modem cable with full hardware handshaking lines (RTS/CTS and DTR/DSR).

You should configure the KL-11 for 9600 baud, 8 bits, no parity, and you should do the same for the Unix server. On my FreeBSD box, I did:

```
$ stty -f /dev/ttyid1 9600 cs8 clocal -crtscts
```

If you are using hardware handshaking lines (and you should), change this to

```
$ stty -f /dev/ttyid1 9600 cs8 -clocal crtscts
```

I'd also recommend having a copy of Kermit on the Unix server end. If no communication seems to be taking place, open the serial line with Kermit, and see if characters are being received from the PDP-11. An RS-232 breakout box here could also be useful.

7 Starting `vtserver`

`vtserver` is started with no arguments, and reads its configuration file `.vtrc`. You should have done the serial port settings before this. `vtserver` starts up with a short description of the tape's contents:

```
Virtual tape server, Revision: 1.13
Records are:
   0 boot
   1 cat
   2 tape_contents
   3 mkfs
   4 icheck
   5 restor
   6 v7rootdump

Opening port /dev/ttyd1 .... Port open
```

It will then sit there waiting for commands from the PDP-11. When commands are being performed, you will see something like the following:

```
Opened boot
qrqrqrqrqrqrqrqrqrqrqrqrqrqrqrqr EOF

Opened mkfs
rrrrrrrrrrrrrrrrrrrrr
```

As each file is opened, it is shown on the screen. 'r' signifies a read of 512 bytes. 'qr' signifies a read of 512 bytes at boot time. If the PDP-11 tries to read past the file's end, an EOF is shown. Finally, a progress message is shown every 100K of a file's download.

`vtserver` is terminated with a ctrl-C.

8 Tape Records: Original V7 and VTserver

The VT server has virtual tape records which match those on the original V7 installation tape. One thing to note here is that the `restor` program is described as being at record 4 in `v7_setup.txt`. With VT server, I have included the standalone *icheck* program at record 4. This allows you to do a rough filesystem check once you have a filesystem on a disk.

Therefore, `restor` is now at record 5, and the root filesystem dump is at record 6. Make sure you cross check with the initial `vtserver` screen (shown above) as you do the installation.

9 Booting via `vtserver`

The first stage of the boot process is to hand-enter the boot code which retrieves the file `boot` from the `vtserver`. This code should be entered at address 070000 onwards. The code is given in the `vtboot.pdp`, and in octal below.

```
0070000:  010706 005001 012703 176500 012704 070102 112400 100406
0070020:  105763 000004 100375 110063 000006 000770 005267 000046
0070040:  004767 000024 001401 005007 012702 001000 004767 000010
0070060:  110021 005302 001373 000750 105713 100376 116300 000002
0070100:  000207 025037 000000 000000 177777
```

Toggle in the code, and start execution at address 70000. If all goes well, you will see "Opened boot qrqrr..." from `vtserver`, and you will receive the following on the PDP-11 console.

10 Second-Stage Bootstrap

You should have the following on the PDP-11 console:

```
New Boot, known devices are hp ht rp rk rl tm vt
:
```

This is the UNIX second stage bootstrap, slightly modified to name the devices it knows about. You should now read the file `v7_setup.txt` (or its PostScript equivalent) as we will be following the boot process described there.

At the present, the UNIX second stage bootstrap only knows about:

hp: RP04, RP05 and RP06 disks.

rp: RP03 disks.

rk: RK05 disks.

rl: RL01 and RL02 disks.

ht: TU16 or TE16 tape drive.

tm: TU10 tape drive.

vt: The Virtual Tape drive described here.

If you don't have any of the named disks, you're outta luck. The next project (perhaps with someone else's help) is to add extra disk devices to the list. It should be relatively easy to modify the existing boot code from 2.9BSD or 2.11BSD to work here. The 7th Edition UNIX kernel would also have to be modified.

11 Filesystem Sizes

For some of the disk types, 7th Edition UNIX reserves space on the boot disk for swap. Therefore, you must ensure that your root filesystem doesn't overlap the swap space. Each compiled UNIX kernel has a different idea where the swap space is. Section 18 gives the configuration files for each UNIX kernel image supplied.

Note that for some disk types, the swap space is on the **second** disk unit, and starts at block zero. For these systems, any filesystem you create on the second disk unit (e.g `/usr`) must come *after* the swap space.

The following table shows you how big the root filesystem can be, on a device basis.

Device	Root (blocks)	Swap (blocks)	Description
hp	9614	8778	Swap on 2nd disk
rp	5000	2000	Swap on 2nd disk
rk	4000	872	Swap on boot disk
rl2	18000	2480	Swap on boot disk

You need to use the values in the second column when using `mkfs` below.

12 Making a Filesystem

Let's quickly go through the steps of installing the V7 root dump onto RK05 unit 0. Load the `mkfs` program, and enter a filesystem size of 4000 blocks on rk unit 0:

```
New Boot, known devices are hp ht rp rk rl tm vt
: vt(0,3)
.*.*.*.*.*.*.*.*.*.*.*.*.file sys size: 4000
file system: rk(0,0)
isize = 1280
m/n = 3 500
```

The VT tape software prints '.' and '*' alternating as it receives the 512-byte blocks from the `vtserver`. `mkfs` has printed out the number of i-nodes (1280), and the 'M' and 'N' values.

13 Restoring the Root Filesystem Dump

After `mkfs` has finished, the second stage boot command line should reappear. You can now run `icheck` (record 4) to verify the filesystem: use the same disk name as you did before (e.g rk(0,0)).

With that done, you can restore the root filesystem on to the newly-created disk system by using `restor` (record 5):

```
New Boot, known devices are hp ht rp rk rl tm vt
: vt(0,5)
*.*.*.*.*.*.*.*.*.*.*.*.*Tape? vt(0,6)
.Disk? rk(0,0)
Last chance before scribbling on disk.
*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*. (etc)
```

Given the size of the root filesystem, it should take around 45 minutes at 9600 bps to complete the restore.

14 Installing Disk Bootstraps

If the restor has worked, you now have a UNIX root filesystem on your disk. To boot the appropriate kernel for your disk type: type one of the following at the boot prompt:

```
New Boot, known devices are hp ht rp rk rl tm vt
: rk(0,0)rkunix
: rl(0,0)rl2unix
: hp(0,0)hptmunix
: rp(0,0)rptmunix
```

You can now follow the rest of the installation instructions in `v7_setup.txt`.

One of the first things you should do once you have booted to a shell prompt is to 'cd' into `/dev` and make appropriate devices for your system. Read `/dev/makefile` and `v7_setup.txt` for details.

With this done, you must then write the disk bootstrap code into the first block of the boot disk. Change into the `/mdec` directory and do:

Device	Command
hp	dd if=hpuboot of=/dev/rhp0
rp	dd if=rpuboot of=/dev/rrp0
rk	dd if=rkuboot of=/dev/rrk0
rl1	dd if=rluboot of=/dev/rrl0
rl2	dd if=rluboot of=/dev/rrl0

depending on your disk type. Fingers crossed, you can shut UNIX down (gracefully), and reboot using the disk.

15 Booting UNIX from Disk

At the moment there is a small problem which makes booting slightly more cumbersome than it should be. The RK and RL disk boot blocks come from a non-vanilla 7th Edition UNIX: vanilla V7 didn't have bootstraps for RKs or RLs. To boot from either of these disks, you first have to boot the VT bootloader. For example:

```
@boot <=== Normally you could type rl2unix here
New Boot, known devices are hp ht rk rl rp tm vt
: rl(0,0)rl2unix
mem = 178048
# STTY -LCASE
# sync
```

16 Comments on `hp` and `rp` Disks

Unlike the `rk` and `rl` devices, where `/dev/rl0` is the whole disk unit, the UNIX drivers for `hp` and `rp` disks create 'virtual' devices, each of which cover only a portion of the disk surface. The following table gives the available disk devices.

Device	# Blocks	Start Blocks	Comment
/dev/hp0	9614	0	cyl 0 thru 22
/dev/hp1	8778	9614	cyl 23 thru 43
/dev/hp4	161348	19228	cyl 44 thru 429
/dev/hp5	160930	187910	cyl 430 thru 814
/dev/hp6	153406	19228	cyl 44 thru 410 (rp04, rp05)
/dev/hp7	322278	19228	cyl 44 thru 814 (rp06)
/dev/rp0	81000	0	cyl 0 thru 405
/dev/rp1	5000	0	cyl 0 thru 24
/dev/rp2	2000	5000	cyl 25 thru 34
/dev/rp3	74000	7000	cyl 35 thru 405

This means, for example, that you could use `/dev/hp0` as your root partition, `/dev/hp1` for `/usr` and `/dev/hp4` for `home`.

17 The Rest of Seventh Edition

The distributed `v7rootdump` file contains just enough of a root filesystem to fit into 4,000 disk blocks (i.e an RK05). You will want to load the rest of the root filesystem, and `/usr`, onto your PDP-11.

It is up to you to partition your hard disk and mount `/usr` as required. An optional part of the VTserver software is the rest of the 7th Edition binary distribution in tar files. These are broken up into small units, because you'd get frustrated if you had to wait several hours to load a single file over to the PDP-11 using VT server.

The available files at the FTP site named above are:

v7doc.tar.Z contains usr/dict usr/doc usr/man usr/pub usr/spool and a few files that I couldn't fit onto the root dump.

v7lib.tar.Z contains usr/games usr/include usr/lib and the /lib/*.a files which again didn't fit onto the root dump.

Note that all of the tarballs have been created so that they can be extracted from `/`. The distribution does not contain any source code, due to licensing reasons. The files `v7rootdump`, `v7doc.tar.Z` and `v7lib.tar.Z` are copyright SCO and you must agree to the binary license described in the file `unix_license.txt` before you use them.

N.B /dev/tty1 doesn't seem to work yet!

I have added two programs for extracting these files to `/bin`: `uncompress` and `vtcat`. Once you have rebooted and partitioned your disks, you can used these tools to load the files from the two compressed tarballs.

For example, assume you have v7lib.tar.Z as record 7 on the `vtserver`. [You can shutdown `vtserver`, edit `.vtrc` and restart the server to do this.] To extract v7lib.tar.Z, do the following commands on the PDP-11:

```
$ cd /
$ vtcat /dev/tty1 7 | uncompress | tar vxf -
```

For those who are tight on disk space, you can untar both tarballs, and build new tarballs with only those parts of 7th Edition UNIX you require. You need to create 'old style' tarballs (use the Gnu `tar` –o option), and compress the tarballs with the –b12 option to `compress`.

18 System-specific Kernel Configuration Files

Nearly all of the supplied UNIX kernel images were built by someone else. Below, I give what I believe are the configuration files for each kernel. See the 7th Edition mkconf(1) for more details.

hphtunix	**hptmunix**
hp	hp
ht	tm
root hp 0	root hp 0
swap hp 1	swap hp 1
swplo 0	swplo 0
nswap 8778	nswap 8778

rphtunix	**rptmunix**
rp	rp
ht	tm
root rp 1	root rp 1
swap rp 2	swap rp 2
swplo 0	swplo 0
nswap 2000	nswap 2000

rkunix	**rl2unix**
rk	rl
tm	tm
root rk 0	root rl 0
swap rk 0	swap rl 0
swplo 4000	swplo 18000
nswap 872	nswap 2480

19 Changes from Vanilla 7th Edition UNIX

The following files have been added or changed from the vanilla 7th Edition UNIX given to the PUPS archive by Henry Spencer.

- New kernel images `rkunix`, `rl1unix` and `rl2unix`.
- A new directory `/mdec`, with the disk bootstrap blocks. `hpuboot` and `rpuboot` come from the directory `/usr/mdec`: the others come from a modified UNIX distribution given to the PUPS archive by Torsten Hippe.
- A new `/boot` which groks RL02s.
- New programs `/bin/vtcat`, `/bin/compress`, `/bin/uncompress` and `/bin/zcat`.
- A new program `/etc/fsck`, from Torsten Hippe, which works on V7 filesystems.

For those people who have valid source licenses for 7th Edition UNIX, I can give you the VT modifications to `/usr/src/cmd/standalone`. There are a few changes to the existing code, plus two new source files: the first stage bootstrap code, `vtboot.s`, and the VT client code, `vt.c`. I have also borrowed the `rl.c` code from 2.9BSD, to allow RL02s to be used.