



DECStation Linux installation HOWTO

Beware, this document is work in progress, and still incomplete!

Overview

1. [Getting the software](#)
 2. [Setting up a boot and root FS server](#)
 3. [Attaching a serial console](#)
 4. [Netbooting](#)
 5. [Partitioning a local disk](#)
 6. [Installing Linux on a local partition](#)
 7. [Bootting with a local root FS](#)
 8. [Adding packages to your system](#)
-

Getting the Software

There is no nicely packaged DECStation linux distribution like RedHat, but there are lots of packages available that run on DECStation linux. So we'll have to download several files from different servers to get our system going. I will not go into detail about the packages needed on your (intel) boot server, the easiest way is to install a RedHat system, that includes most features you'll need.

At least you'll need the following packages installed: **tftp**, **nfs-server**, and **bootp** (besides standard networking setup).

For your DECStation, you need the following:

- A boot kernel. Download the correct **ECOFF** kernel from my [kernel selector page](#).
- A root filesystem that can be used as NFS-based root. This document uses [declinuxroot-current.tgz](#)

And lateron:

- RPM Packages to expand your system. You can find lots and lots of packages on ftp.linux.sgi.com.
 - The kernel source to compile your own kernel and start hacking!
-

Setting up the boot and root FS server

Most DECStations can boot from a bootp / tftp server (some can only boot from a MOP server, I have no information about MOP servers on linux). So now we have to setup bootp, tftp, and NFS export on your server. These examples all come from a i386 linux box running RedHat linux, the actual location and/or your file syntax may be different on your server. When in doubt, consult your manuals.

Before setting up your server, you must have allocated an IP address in the same subnet your ethernet card is connected to. Also you'll have to note the Ethernet address of your DECStation.

NFS server for your root filesystem

Your server will provide a root filesystem for your DECStation over NFS. In order to do that, you'll have to configure NFS, export a part of your filesystem to the DECStation (that means, give the DECStation permission to access that part of the server's filesystem).

1. Install the `nfs-server` package.
2. Make a directory to contain the root filesystem for the DECStation, for example `/export/mips`.
3. Edit the `/etc/exports` file to allow access from your DECStation to `/export/mips`.
`/export/mips decstation(rw,no_root_squash)`
 (Assuming your decstation's IP hostname will be 'decstation')
4. Extract the `declinuxroot-current.tgz` in `/export/mips`
5. Place the `util-linux-2.7-19.mipsel.rpm` in `/export/mips`

TFTP Server for the DECStation kernel

The TFTP server is needed only for downloading the kernel binary from your server to your DECStation. The tftp protocol is a very simple protocol, and has no security at all. That's why usually tftp servers are only used on boot servers, and most likely tftp is not configured on your linux system by default. Due to the lack of security, tftp server run 'chroot', that means that they change their root location to some special directory, that only contain the bootfiles that must be accessible by tftp, and no other files. The usual location for these files is `/tftpboot`.

To set up your tftp server:

1. Install the tftp package
2. Add the tftp daemon to `/etc/inetd.conf`:
`tftp dgram udp wait root /usr/sbin/tcpd in.tftpd /tftpboot`
3. Create the `/tftpboot` directory.
4. Copy the kernel image file from the tar archive to `/tftpboot/vmlinux.ecoff`.

Bootp for booting your DECStation

Follow these steps to configure your bootp server.

1. Install the bootp package
2. Make sure your `/etc/inetd.conf` file contains a line like:
`bootps dgram udp wait root /usr/sbin/tcpd bootpd -d 9 -c /tftpboot`
3. Edit the `/etc/bootptab` file to contain the info needed for booting:
 You must at least give the following entries:
`decstation:\`

```

:ht=1:\
:ha=08002b376376:\
:ip=10.10.1.12:\
:bf="vmlinux.ecoff":\
:sm=255.255.255.0:\
:gw=10.10.1.4:\
:rp="/export/mips":\
:td=/tftpboot:\
:hd=/tftpboot:\
:hn:

```

The **ha=** tag notes the ethernet address of the decstation. The **rp=** tag defines the location on your server where you have build your NFS root tree for the DECStation.

Under some conditions (depending on the software version of your server), you might need static arp entries to get the decstations booting:

Add the following to your favourite rc file eg. **/etc/rc.d/rc.local**.


```
/sbin/arp -s 10.10.1.12 08:00:2b:37:63:76
```

Finally it might be a good idea to restart your server (or at least, restart NFS, and send a SIGHUP to inetd).

Attaching a serial console

The current state of the DECStation port doesn't include support for framebuffer cards and / or local keyboards. So we have to use a serial terminal as system console. I myself use kermi on my server to access the serial console. This has the extra advantage, that you can log the session to a file on your server. This is very usefull for debugging, or sharing your success with fellow hackers :-).

DEC normally uses the PRINTER port for serial consoles, not the 'communication' port, that port is reserved for modems. So, depending on your model, you'll have to search for a port (DB25 or DEC modular connector) labeled as printer port. You might need a NULL-modem cable to connect to your terminal (or PC serial port), depending on your model and cable.

For my 5000/260 I have to use the port labeled 3 (communications port 3).

Next you have to change the console from the keyboard / internal frambuffer to the serial port. This is done by setting a NVRAM environment variable. It might be best to consult your manual what value is used with your model. See the [table below](#) for a list of console settings.

Now power-toggle your DECStation, and the selftest and prom messages should appear on the serial console.

Normally, the serial console is auto-selected when the system is powered-up WITHOUT a video card or a keyboard. So you could try to pull out the video card, and start the system with just the serial console plugged in.

Netbooting

If everything went OK until now, you should have your setup ready for the first linux boot attempt. Because Linux can not autodetect where the console is located, we have to supply boot arguments to the kernel, specifying the console. This depends on the DECStation model, but you might try several possible options in sequence if you don't know which one to use. My DECStation 5000/260 uses ttyS2. Secondly, we have to specify the root location. Assuming we have correctly configured bootp, we can just specify /dev/nfs as root FS. Finally we have to add the 'rw' argument to let the kernel mount the NFS root filesystem read-write (default is read-only, which doesn't make sense on a NFS root).

OK, now for the DECStation magic to boot over bootp / tftp. Lets try one of the following command:

```
>>boot 3/tftp/vmlinux.ecoff console=ttyS2 single rw (for DECStation 5000)
```

```
>>boot -f tftp()vmlinux.ecoff "console=ttyS3 single rw" (for DECStation 3100, note the quotes!)
```

See the [table below](#) for other models.

The digit '3' for the 5000 should be replaced by the module number containing the network card. You can find out what modules are in your system by giving the 'cnfg' command at the prom prompt.

You might want to watch the messages file on your server for messages from the bootp / tftp daemons. If the boot fails, they might provide usefull information on what went wrong.

Now, assuming your server sends you a kernel, you should see something like the following:

```
-tftp boot(3), bootp 10.10.1.4:/tftpboot/vmlinux.ecoff
-tftp load 1169824+113264+122480
Launching kernel...
This DECstation is a DS5000/2x0
Loading R4000 MMU routines.
CPU revision is: 00000440
Primary instruction cache 16kb, linesize 16 bytes)
Primary data cache 16kb, linesize 16 bytes)
Secondary cache sized at 1024K linesize 32
Linux version 2.2.1 ... etc.
```

If it stops before '**Launching kernel**', the tftp file might not be transferred correctly. Some DECStations only allow kernels smaller than 1Mb to be loaded over tftp. I have no solution to that at this time.

The message '**This DECstation is a ...**' is the last message printed through the prom-io. Thus if this is your last line of output, you possibly have a wrong '**console=**' argument.

The correct location for your root FS will be passed to the DECStation by your bootp server. We start first booting single user only, because you the getty process might have troubles using your serial line (mine at least has).

If all goes well, you finally should see something like:

```
Enabling swap space.
Initializing random number generator...
bash#
```

Congratulations, you have just succesfully booted linux on your DECStation!

You might try to continue, and go on to multi-user mode. Just type **'exit'** to the bash prompt, and init will boot to level 3.

After a while you should see the message:

```
The system is ready
```

Now inetd should be running, and even if you can't get the console login prompt to work, you might try to log in over telnet. The root account on the NFS root system has no password, so you should be able to log in.

If you don't get this working, maybe the [troubleshooting guide](#) has some hints for you.

Partioning a local disk

If your kernel has recognised your SCSI adapter, and has detected a local SCSI disk, you might be brave, and try to install Linux on your local hard disk. You should have seen messages like the following during your boot:

```
SCSI ID 7   Clock 25 MHz CCF=0 Time-Out 167 NCR53C9x(esp236) detected
ESP: Total of 1 ESP hosts found, 1 actually in use.
scsi0 : ESP236
scsi : 1 host.
  Vendor: DEC      Model: RZ58      (C) DEC  Rev: 0100
  Type:   Direct-Access                      ANSI SCSI revision: 02
Detected scsi disk sda at scsi0, channel 0, id 2, lun 0
  Vendor: DEC      Model: RZ58      (C) DEC  Rev: 0100
  Type:   Direct-Access                      ANSI SCSI revision: 02
Detected scsi disk sdb at scsi0, channel 0, id 3, lun 0
scsi : detected 2 SCSI disks total.
esp0: target 2 [period 200ns offset 15 5.00MHz synchronous SCSI]
SCSI device sda: hdwr sector= 512 bytes. Sectors= 2698061 [1317 MB]
[1.3 GB]
esp0: target 3 [period 200ns offset 15 5.00MHz synchronous SCSI]
SCSI device sdb: hdwr sector= 512 bytes. Sectors= 2698061 [1317 MB]
[1.3 GB]
```

Beware, the next step will wipe out all information on the disk!

The only partition format that we can *create* with DECStation linux at the moment is a DOS-type partition table. The kernel recognizes several other formats, but **fdisk** will create a DOS-type table by default. Try creating a linux native and a linux swap partition using **'fdisk /dev/sda'**. I would suggest to make the first partition your root-fs, and use a second partition as swap space. It might be wise to reboot after creating new partitions.

You should now see a kernel message that it has recognized the new partitions:

Partition check:

```
sda: sda1 sda2 (MSDOS partition table)
```

Then, we can go on to the real installation.

Installing Linux on a local partition

We are still running on our NFS-root, but we have a local disk with (at least) two partitions. We first try to create the filesystem:

```
# mke2fs /dev/sda1
```

Then, we initialise the swap partition:

```
# mkswap /dev/sda2
```

Now we mount the new filesystem:

```
# mount /dev/sda1 /mnt
```

And we copy our root-fs from the NFS server to the local partition:

```
# cd /
```

```
# tar ctf - . | (cd /mnt; tar xpf -)
```

In order to get all things right, we have to edit some files on your local disk FS:

- /etc/fstab (change `/dev/root` to `/dev/sda1`, type `ext2` instead of `nfs`)
You might want to add your swap partition too.
- /etc/rc.d/rc.sysinit (remove the comment '#' before the lines that remount the root-fs near line 95)

It should look like:

```
# Remount the root filesystem read-write.
```

```
# Not for NFS root.
```

```
echo "Remounting root filesystem in read-write mode."
```

```
mount -n -o remount,rw /
```

After that, you should unmount the local disk, and shut down.

Bootting with a local root FS

Assuming that you followed the example above, we now try to use the local disk as root device. Please note that this time we do want to mount the FS read-only by the kernel, so don't add the 'rw' argument.

```
>>boot 3/tftp/vmlinux.ecoff console=ttyS2 root=/dev/sda1 single (for  
DECStation 5000)
```

For other models, you should use the same boot command as for your netbooting, but add

```
root=/dev/sda1, and remove 'rw'.
```

You can look for the following line during boot:

```
VFS: Mounted root (ext2 filesystem) readonly.
```

If it reads ext2, it should be your local disk. If you were able to run multi-user, using NFS root, you should be able to go to multi-user now too.

Adding packages

The rpm command is on the root-system, so you can just use rpm to add packages. Packages for DECStation linux should be named ***.mipsel.rpm**, not ***.mips.rpm**. Lots of packages are available from <ftp://ftp.linux.sgi.com/pub/linux/mips>

Models / Commands

| Model | Console setting | Netboot command | Console port label |
|--|-----------------------------------|---|----------------------|
| DECStation 2100 | ? | <code>boot -f tftp() "console=ttyS3 single rw"</code> | <input type="text"/> |
| DECStation 3100 | <code>setenv console 8</code> | <code>boot -f tftp() "console=ttyS3 single rw"</code> | <input type="text"/> |
| DECStation 5000/20, /25, /33, /50 | <code>setenv console s</code> | <code>boot 3/tftp console=ttyS0 single rw</code> | <input type="text"/> |
| DECStation 5000/120, /125, /133, /150 | ? | <code>boot 3/tftp console=ttyS2 single rw</code> | <input type="text"/> |
| DECStation 5000/200 | <code>setenv console s</code> | <code>boot 6/tftp console=ttyS3 single rw</code> | <input type="text"/> |
| DECStation 5000/240, /260 | <code>setenv console s</code> | <code>boot 3/tftp console=ttyS2 single rw</code> | <input type="text"/> |



© Copyright 1998, 1999 [The DECstation Linux Porting Team](#).
 These pages are maintained by Karel van Houten (K.H.C.vanHouten@kpn.com).
 Last change: 21-9-99

