Jumpstarting With a SPARC: Netbooting Sun Workstations

Revision 0

Foreword

This document was compiled by me (Europa) in 2024 and contains a combination of information from others that I have placed here for posterity (and updated where applicable) and information based on my own experiences. I would like to thank the many administrators and hobbyists that came before me and have made it possible for me to use and learn about my Sun systems in a manner that allows me to create this document through the recording of their knowledge. I would also like to specifically thank NCommander and their community for helping me when limited documentation was available. I hope this document will survive for years to come and be passed from one hand to the next. Archival and documentation is important, and I hope that this document aids with that effort.

A note about inline citations: As I said, some of this guide is taken from other sources. To streamline citation of those sources, I have put letters in parentheses next to the appropriate section headings. These correspond to webpages referenced in Appendix C. I do not wish to take credit for the work of others, and I believe firmly in crediting sources.

Abstract

If you're reading this document, it's likely that you have been brought here from a web search or someone linked it as an answer to a question you or someone else posed. The purpose of this document is, first and foremost, to provide a consolidated resource for those looking to boot their Sun workstations from the network based on my findings. There are many reasons one may wish to do this, including a lack of physical installation media or a lack of working removable media drives on the system one wishes to boot.

This document will make a few assumptions, namely that you have a basic understanding of networking and Unix-like operating systems, as well as assuming that you have root access (either directly or via su) to the server system, and that you'll be working from a root shell. As I am limited by the hardware I have at my disposal, the procedures listed in this document have been tested on a SPARCstation 5, an Ultra 1, and an Ultra 5. That being said, the basics of this document should hold true for any Sun system that can boot from the network in the same way that these systems can. I will primarily focus on booting from NetBSD (and OpenBSD where applicable) and Solaris. If you wish to apply these instructions to other platforms and operating systems, I have included the resources I pulled from as well as other useful resources in Appendix C.

Before you proceed, you must have a few things prepared: - A wired network. It does not need to be connected to the internet, although it may be beneficial for some client operating systems. - A system that can act as a server and a system that you wish to be your client. The server can be anything capable of wired networking and running an operating system capable of performing the tasks in this guide. - Your client's platform group. If you don't know this, please refer to Appendix B-i. - A programmed IDPROM and NVRAM on the client. Unless you have specifically ensured your timekeeper's battery has juice left, it's likely dead. If you need to program your timekeeper, refer to Appendix A. If you have all of those things, feel free to proceed to whichever section is applicable to your goals. Every network and configuration is different, and you may have to tweak these guides to meet your needs. I can say these worked on my machines but, as always with things like this, your mileage may vary. I make no guarantees about any of the stuff here other than that it has been tested by me to work. I, however, am an individual and only have so many resources at my disposal, so I cannot test every possible configuration. I will add more configurations with server and client operating systems as I test them and, if anyone has their own configurations that they would like to add, feel free to do so on Github. Have fun and best of luck.

I. SunOS 4 Diskless Install

I-a. NetBSD/OpenBSD [1]

1. Ensure you have the following services enabled in your rc.conf:

- rarpd
- bootparams (called bootparamd in rc.conf, but refered to as bootparams elsewhere)
- tftpd (part of inetd, enabled by uncommenting the tftpd line in /etc/inetd.conf)
- nfs_server (in addition to rpcbind, mountd, lockd, and statd)

2. These services are also in /etc/inetd.conf and may be useful to uncomment (IF you are on a trusted network):

- ftp
- telnet
- rlogin
- rsh
- rexec
- time

3. Create or modify the /etc/ethers file with your Sun's ethernet address and hostname:

`/etc/ethers:`

[1] FAQ: Frequently Asked Questions about Sun NVRAM, IDPROM, hostid

```
08:00:20:00:07:EA client
```

4. Modify /etc/hosts with the hostname you specified in /etc/ethers and the IP address you wish to assign your client

```
/etc/hosts:
...
192.168.0.11 client
```

5. Make the /tftpboot directory if it doesn't already exist.

```
# mkdir /tftpboot
```

6. Make a directory where you wish to put the contents of the install CD.

```
# mkdir /home/client
```

7. Create or modify the /etc/bootparams file.

```
/etc/bootparams:
installclient root=server:/home/client swap=server:/home/client/swap gateway=server:0xffffff
```

8. Add the location of the copied install CD to /etc/exports.

```
/etc/exports:
/home/client -alldirs -maproot=root
```

9. Restart (or start) all services listed in Step 1.

10. Mount the SunOS 4.1.x CD

```
# mount -t cd9660 -o ro /dev/cd0 /cdrom
```

11. Set up the client's filesystem heirarchy and swap file.

```
# cd /home/client

# dd if=/dev/zero of=swap bs=<swap size in megabytes>m count=1

# mkdir -p usr/kvm

# tar -xvpf /cdrom/export/exec/proto_root_sunos_4_1_4

`# cd /home/client/usr
```

12. Extract the common OS packages.

```
# for x in /cdrom/export/exec/sun4_sunos_4_1_4/*; do tar -xpf $x; done

# tar -xpf /cdrom/export/share/sunos_4_1_4/manual
```

13. Install the architecture-specific packages. (If you don't know what platform group or architecture your system is, refer to Appendix B-i. For the purposes of demonstration, I'm using sun4m.)

```
# cd kvm

# tar -xpf /cdrom/export/exec/kvm/sun4m_sunos_4_1_4/kvm

# tar -xpf /cdrom/export/exec/kvm/sun4m_sunos_4_1_4/sys
```

`14. Add the server and the client to the client's /etc/hosts file.`

```
# cd ../../
```

```
./etc/hosts:
<server ip address> server
192.168.0.11 client
```

15. Create the client's /dev nodes.

```
# cd dev

# sed -i 's|^PATH=|PATH=/sbin:|g' MAKEDEV

# ./MAKEDEV std

# ./MAKEDEV pty0

# ./MAKEDEV win

# cd ..
```

16. Create the client's /etc/fstab file.

```
./etc/fstab:
server:/home/client / nfs rw 0 0
server:/home/client/usr /usr nfs rw 0 0
swap /tmp tmp rw 0 0
```

(Special note: At this time it would be ideal to also uncomment the mount /tmp line in rc.local so that /tmp will be mounted and you can use X11 and OpenWindows, among other things.)

17. Copy critical files to sbin and vmunix to the root directory.

```
# cp usr/kvm/boot/* sbin

# cp usr/kvm/stand/sh sbin

# cp usr/bin/hostname sbin

# cp usr/kvm/stand/vmunix .
```

18. Copy the tftp boot file to /tftpboot and symlink it to the proper name for your system. (If you don't know how to do this, refer to Appendix B-iv.)

```
# cp usr/kvm/stand/boot.sun4m /tftpboot

# cd /tftpboot

# ln -s boot.sun4m C0A8000B.SUN4M
```

19. If all works out, at this point you should be able to tell the Sun to boot into single user mode from the network.

```
ok boot net -s
```

On the Sun:

20. Compile the source for the various utilities in the bigfs patch and put them in their proper places.

- /etc/dump
- /etc/fsck
- /usr/bin/df and /usr/5bin/df
- /etc/fsirand
- /etc/mkfs
- /usr/etc/quotacheck

(fsusageck.c will fail to compile but it doesn't seem to be necessary, as I don't actually find it anywhere in the system.)

21. Apply the NFS Jumbo Patch.

```
# cd /path/to/nfs/patch

# cp /usr/kvm/sys/`arch -k`/OBJ/nfs_client.o /usr/kvm/sys/`arch -k`/OBJ/nfs_client.old

`# cp `arch -k`/nfs_client.o /usr/kvm/sys/\`arch -k`/OBJ/nfs_client.o

# cp /usr/kvm/sys/`arch -k`/OBJ/nfs_server.o /usr/kvm/sys/`arch -k`/OBJ/nfs_server.old

# cp `arch -k`/nfs_server.o /usr/kvm/sys/`arch -k`/OBJ/nfs_server.o

# cp /usr/kvm/sys/`arch -k`/OBJ/nfs_vnodeops.o /usr/kvm/sys/`arch -k`/OBJ/nfs_vnodeops.old

# cp `arch -k`/nfs_vnodeops.o /usr/kvm/sys/`arch -k`/OBJ/nfs_vnodeops.o
```

22. Recompile the kernel.

```
# cd /sys/`arch -k`/conf

# cp GENERIC GENERIC.1
# config GENERIC.1

# cd ../GENERIC.1

# make

# cp /vmunix /vmunix.save

# cp vmunux /vmunix
```

23. Reboot into multi-user mode this time. If everything was done correctly it should boot fully and you can login as root without it panicking. (Note: if your default boot device is set to something other than net, you should `halt` instead of reboot and then issue the `boot net` command at the OpenBoot "ok" prompt.)

II. Solaris Diskful Install

II-a. Solaris [2]

1. Create a directory where you want the server to place the Solaris install files.

```
# mkdir /var/Solaris
```

2. Navigate to the Tools directory on the install disc.

```
# cd /path/to/Tools
```

3. Run the setup_install_server script and direct it to the directory created in Step 1

```
# ./setup_install_server /var/Solaris
```

Special note for those making a Solaris 10 install server: There has been behavior observed by nick99nack that, when Solaris 10 is installed on a ZFS pool, you may be unable to export the install directory on account of it "is not a local filesystem." If that is an issue that you run into, use the solution he provided me below to create a UFS-formatted file that can be mounted as a loop device:

```
# mkfile 10g /var/jumpstart.img [you can put it anywhere]
# lofiadm -a /var/jumpstart.img
# newfs /dev/lofi/1
# mount /dev/lofi/1 /var/Solaris
```

4. If your distribution of Solaris comes on multiple discs, you will need to, at this point, eject the first disc and put in the second disc. Volume Manager should mount the disc for you.

[2] Installing Solaris Over the Network - hosted on tenox.pdp-11.ru

5. Navigate to the Tools directory on the new disc and run the add_to_install_server script

```
# ./add_to_install_server /var/Solaris
```

6. Repeat this as many times as necessary.
7. Create or modify the /etc/ethers file with your Sun's ethernet address and hostname:

```
/etc/ethers:
8:0:20:c0:ff:ee installclient
```

8. Modify /etc/hosts with the hostname you specified in /etc/ethers and the IP address you wish to assign your client

```
/etc/hosts:
...
192.168.0.10 installclient
```

9. Navigate to the Tools directory of the install server and run the add_install_client script, specifying your client's platform group and hostname.

```
# ./add_install_client installclient sun4u
```

Special note for those making a Solaris 10 install server to boot Solaris 9 or earlier: You need to run the inetconv command to make the changes the script made to your inetd.conf file reflect in the actual settings of the smf service!

```
# inetconv
```

10. If everything went according to plan, you can tell your Sun workstation to boot from the network and it will boot into the Solaris installer.

```
ok boot net
```

III. NeXTSTEP/OPENSTEP Diskful Install

III-a. NetBSD/OpenBSD

1. Ensure you have the following services enabled:

- rarpd
- bootparams (called bootparamd in rc.conf, but refered to as bootparams elsewhere)
- tftpd (part of inetd, enabled by uncommenting the tftpd line in /etc/inetd.conf)
- bootps (part of inetd, enabled by uncommenting the bootps line in /etc/inetd.conf) or dhcpd
- nfs_server (in addition to rpcbind, mountd, lockd, and statd)

2. Create or modify the /etc/ethers file with your Sun's ethernet address and hostname:

```
/etc/ethers:
08:00:20:c0:ff:ee installclient
```

3. Modify /etc/hosts with the hostname you specified in /etc/ethers and the IP address you wish to assign your client

```
/etc/hosts:
...
192.168.0.10 installclient
```

4. Make the /tftpboot directory if it doesn't already exist.

```
# mkdir /tftpboot
```

5. Make a directory where you wish to put the contents of the install CD.

```
# mkdir -p /export/installcd
```

6. Mount the install CD.

```
# mount -t ufs -o ro,ufstype=nextstep-cd /dev/cd0 /mnt
```

7. Copy the contents of the CD you mounted in the previous step to the directory you created in Step 5.

```
# cp -a /mnt/* /export/installcd
```

8. Symbolically link the Sun boot file to the /tftpboot directory, where the name is your Sun's IP address represented in hexadecimal followed by your platform group. (If you don't know how to do this, please see Appendix B-iv. If you don't know your platform group, please refer to Appendix B-i.)

```
# ln -s /export/installcd/private/tftpboot/sparc/bootnet /tftpboot/C0A8000A.SUN4M
```

9. Create or modify the /etc/dhcpd.conf or /etc/bootptab file.

```
/etc/dhcpd.conf:
deny unknown-clients;
allow bootp;
subnet 192.168.0.0 netmask 255.255.255.0 {
range 192.168.0.100 192.168.0.150
}
group {
host installclient {
hardware ethernet 08:00:20:00:07:ea;
filename "C0A8000B.SUN4M";
fixed-address installclient;
option root-path "/export/installcd";
}
}
```

```
/etc/bootptab:
installclient:\
:ht=ether:\
:ha=080020C0FFEE:\
:sm=255.255.255.0:\
:lg=<server ip address>:\
:ip=192.168.0.10:\
:hn=installclient:\
:bf=C0A8000A.SUN4M:\
:bs=auto:\
:rp=/export/installcd/:\
:vm=auto:
```

10. Create or modify the /etc/bootparams file.

```
/etc/bootparams:
installclient root=install.server.ip.address:/export/installcd private=install.server.ip.add
```

11. Add the location of the copied install CD to /etc/exports.

```
/etc/exports:
/export/installcd -mapall=root -alldirs
```

12. Restart all services listed in Step 1.

**If you enabled tftpd and/or bootpd, be sure to restart inetd.*

13. If all works out, at this point you should be able to tell the Sun to boot from the network and it'll boot into the install CD.

```
ok boot net
```

III-b. Solaris

1. Ensure you have the following services enabled:

- rarpd
- bootparams (called bootparamd in rc.conf, but refered to as bootparams elsewhere)
- tftpd (part of inetd, enabled by uncommenting the tftpd line in /etc/inetd.conf)
- bootps (part of inetd, enabled by uncommenting the bootps line in /etc/inetd.conf) or dhcpd
- nfs_server (in addition to rpcbind, mountd, lockd, and statd)

2. Create or modify the /etc/ethers file with your Sun's ethernet address and hostname:

```
/etc/ethers:
08:00:20:c0:ff:ee installclient
```

3. Modify /etc/hosts with the hostname you specified in /etc/ethers and the IP address you wish to assign your client

```
/etc/hosts:
...
192.168.0.10 installclient
```

4. Make the /tftpboot directory if it doesn't already exist.

```
# mkdir /tftpboot
```

5. Make a directory where you wish to put the contents of the install CD.

```
# mkdir -p /export/installcd
```

6. Mount the install CD.*

```
# mount -F ufs -o ro,ufstype=nextstep /mnt
```

**I have not specifically tested this part of the procedure yet, as my CD was already dumped. If the above command does not work for you, you can dump the CD over the network to the NFS share (see step 11 for exporting the filesystem) from a machine that can mount the disc.*

7. Copy the contents of the CD you mounted in the previous step to the directory you created in Step 5.

```
# cp -a /mnt/* /export/installcd
```

8. Symbolically link the Sun boot file to the /tftpboot directory, where the name is your Sun's IP address represented in hexadecimal followed by your platform group. (If you don't know how to do this, please see Appendix B-iv. If you don't know your platform group, please refer to Appendix B-i.)

```
# ln -s /export/installcd/private/tftpboot/sparc/bootnet /tftpboot/C0A8000A.SUN4M
```

9. Create or modify the /etc/bootptab file.

```
/etc/bootptab:
installclient:\
:ht=ether:\
:ha=080020C0FFEE:\
:sm=255.255.255.0:\
:lg=<server ip address>:\
:ip=192.168.0.10:\
:hn=installclient:\
:bf=C0A8000A.SUN4M:\
:bs=auto:\
:rp=/export/installcd/:\
:vm=auto:
```

10. Create or modify the /etc/bootparams file.

```
/etc/bootparams:
installclient root=install.server.ip.address:/export/installcd private=install.server.ip.add
```

11. Add the location of the copied install CD to /etc/exports (remembering to execute `shareall` afterward).

```
/etc/dfs/dfstab:
share -F nfs -o root="" /export/installcd
```

12. Restart all services listed in Step 1.

**If you enabled tftpd and/or bootpd, be sure to restart inetd.*

13. If all works out, at this point you should be able to tell the Sun to boot from the network and it'll boot into the install CD.

```
ok boot net
```

Appendix A: Programming the IDPROM [3]

A-i. What is the NVRAM, IDPROM, hostid, and ethernet address, and why is it important?

Before proceeding, a warning: improper use of the OpenBoot PROM console can brick your system. I know the information contained in this document to be correct, but I cannot guarantee you won't brick your machine if you go beyond what I talk about here. Proceed with caution. This is not to scare you, but to inform you.

Your Sun's NVRAM and IDPROM hold firmware configuration settings. This includes the default boot device, diagnostic behavior, default display device, and, perhaps most importantly for this guide, your ethernet address and hostid.

If you're reading this section of the guide (and, honestly, in general), it's likely that your Sun's NVRAM battery is dead. This means three important things: a) your Sun will always boot up in diagnostic mode (causing it to try unsuccessfully to boot from the network), b) your IDPROM information will be invalid, and c) any changes you make to settings will be cleared after you turn your machine off. This is a guide all about network booting and, with an invalid ethernet address, you can't do a lot. So, with that in mind, let's look at the makeup of the IDPROM data:

Byte(s) Contents 0 Always 01 1 First byte of hostid (machine type - for ease of use you can use real-machine-type, but that may not work on older boot proms. DO NOT SET THIS TO A MACHINE TYPE THAT IS NOT YOUR MACHINE'S TYPE) 2-7 6 byte ethernet address (first three bytes are 08, 00, 20) 8-b Date of manufacture (all 0s, doesn't matter) c Second byte of hostid d Third byte of hostid e Fourth byte of hostid f IDPROM checksum (bitwise xor of bytes 0-e)

[3] SunOS 4.1 Diskless Installation Guide

A-ii. A crash course in IDPROM programming

To program your ethernet address to 8:0:20:c0:ff:ee and your hostid to 80coffee on a sun4m machine based on the table above:

```
1 0 mkp
real-machine-type 1 mkp
8 2 mkp
0 3 mkp
20 4 mkp
c0 5 mkp
ff 6 mkp
ee 7 mkp
0 8 mkp
0 9 mkp
0 a mkp
0 b mkp
c0 c mkp
ff d mkp
ee e mkp
0 f 0 do i idprom@ xor loop f mkp
```

If you're on a SPARCserver 1000, type `update-system-idprom` at the "ok" prompt.

There is also another way to set the idprom on some revisions of the OBP: `8 0 20 c0 ff ee 80c0ffee mkpl` and hitting Return followed by hitting CTRL+D followed by CTRL+R. If it gives no output and returns you to the `ok` prompt, it has succeeded. If it shows a Sun Microsystems copyright, it has failed. This is sometimes due to the idprom not being invalidated before running the command. If this is the case, run `f idprom@ 1 xor f mkp` and hit Return, then trying the invocation again. If that still doesn't work, then try the method listed above.

Reset the machine with the `reset` command and it should reboot with the new ethernet address and hostid.

A-iii. Hostid Machine Type Listing

First Byte of hostid	Machine
01	2/1x0
02	2/50
11	3/160
12	3/50
13	3/2x0
14	3/110
17	3/60
18	3/e

First Byte of hostid	Machine
21	4/2x0
22	4/1x0
23	4/3x0
24	4/4x0
31	386i
41	3/4x0
42	3/80
51	SPARCstation 1 (4/60)
52	SPARCstation IPC (4/40)
53	SPARCstation 1+ (4/65)
54	SPARCstation SLC (4/20)
55	SPARCstation 2 (4/75)
56	SPARCstation ELC
57	SPARCstation IPX (4/50)
61	4/e
71	4/6x0
72	SPARCstation 10 or SPARCstation 20
80	SPARCclassic, LX, 4, 5, SPARCserver 1000, Voyager, Ultra, Blade

Appendix B: Platform Groups, useful OpenBoot settings, and other errata

B-i. Platform Group listing: [4]

These are the platform groups for many of the SPARC systems you may encounter:

Model	Platform Group
4/2x0	sun4
4/1x0	sun4
4/3x0	sun4
4/4x0	sun4
SPARCstation 1/1+	sun4c
SPARCstation 2	sun4c
SPARCstation 10/20	sun4m
SPARCstation 5/4	sun4m
SPARCstation IPC/IPX	sun4c
SPARCclassic/SPARCstation LX/ZX	sun4m
SPARCclassic X	sun4m
Ultra 1	sun4u
Ultra 1E	sun4u

[4] Installing Solaris Over the Network - hosted on tenox.pdp-11.ru

Model	Platform Group
Ultra 2	sun4u
Ultra 30	sun4u
Ultra 5/10	sun4u
Ultra 60	sun4u
Ultra 80	sun4u
Blade 1000	sun4u
Blade 2000	sun4u
Blade 100/150	sun4u
Blade 2500 (Silver)	sun4u
Blade 1500 (Silver)	sun4u
Netra T1	sun4u
SPARCserver 1000	sun4d
SPARCserver 3x0/4x0	sun4
SPARCserver 6x0MP	sun4m
Ultra Enterprise/UltraSPARC-based Fire V-seies	sun4u
Fire T1000/T2000	sun4v
Fire T5120/T5220	sun4v
Fire T5140/T5240/T5440	sun4v
x86	i86pc

B-ii. Operating system compatibility matrix

This is a compatibility matrix for operating systems on Sun systems. This information is correct to the best of my knowledge, but may become outdated over time.

System	Compatible Operating Systems
Sun-1	Sun UNIX 0.7
Sun-2	SunOS 1.0 - 4.0.1, NetBSD (Tier II)
Sun-3	SunOS 3.0 - 4.1.1_U1, NetBSD (Tier II), OpenBSD (up to 2.9)
sun4	SunOS 4.0.3e - 4.1.4, Solaris 2.0 - 7, NetBSD (Tier II), OpenBSD (up to 5.9)
sun4c	SunOS 4.1.2 - 4.1.4, Solaris 2.1 - 7, NetBSD (Tier II), OpenBSD (up to 5.9)
sun4d	Solaris 2.2 - 8
sun4m	SunOS 4.1.2 - 4.1.4, Solaris 2.1 - 9, NetBSD (Tier II), OpenBSD (up to 5.9), NeXTSTEP 3.3 and OPENSTEP 4.x (SuperSPARC and microSPARC CPUs only)

System	Compatible Operating Systems
sun4u	Solaris 2.5 - 10 (7 is first 64-bit release), NetBSD (Tier I), OpenBSD
sun4v	Solaris 10 3/05 HW2 and newer and OpenBSD
SPARCserver 600MP	SunOS 4.1.2 - 4.1.4, Solaris 2.1 - 2.5.1, NetBSD (Tier II), OpenBSD (up to 5.9)
UltraSPARC I	Solaris 2.5 - 9 (7 is first 64-bit release), NetBSD (Tier I), OpenBSD

B-iii. Useful OpenBoot settings

B-iii-a. diag-switch?

This determines if the system goes through an extended self-test during startup. It also sets the default boot device to the network by default while this setting is set to true. It will be set to true by default every time you turn on your system on if the timekeeper battery is dead. I would recommend setting this to false to shorten the time it takes to subsequently reboot.

B-iii-b. auto-boot?

This determines if the system tries to auto-boot upon startup. This is set to true by default. It's a matter of personal preference whether it's enabled or not. I prefer to boot to the OpenBoot prompt and then choose my boot device from there. Be aware this setting will be largely inconsequential when you cold boot the system if the timekeeper battery is dead.

B-iii-c. boot-device

This controls what the default setting is when the boot command is issued with no arguments. The default is for it to try booting from disk first and then from the network. Like auto-boot? It's a largely inconsequential setting if your timekeeper battery is dead.

B-iii-d. output-device

I put this here because sometimes you don't want the default output device to be what the firmware defaults to, or you want to change settings for the default output device. This setting allows you to specify the default output device (generally screen or ttya by default depending on if you have a keyboard attached or not) and any settings for said output device (I set my framebuffer to run at 1024x768x60 using this setting, for instance).

B-iv. Working with iso files

B-iv-a. Using lofiadm to mount iso images. (Solaris)

In some versions of Solaris (I've tested with Solaris 10, but it may exist in other versions as well) there exists the lofiadm utility. This exists as a way to work with loopback files (disc/k image files). For the purposes of this guide, we will be looking at using it to mount CD images.

You can invoke the lofiadm utility with the following:

```
# lofiadm -a /path/to/image.iso
```

This will add the iso file as a loopback filesystem on the first available lofi device. You can also specify a lofi device, like so (where n is the lofi device number):

```
# lofiadm -a /path/to/image.iso /dev/lofi/n
```

The output of either of these commands will return the lofi device that the iso file was added to. You can then mount the device like so (where n is the lofi device number):

```
# mount -F hsfs -o ro /dev/lofi/n /mnt
```

You can also combine both commands together:

```
# mount -F hsfs -o ro $(lofiadm -a /path/to/image.iso) /mnt
```

B-iv-b. Using vnconfig to mount iso images. (NetBSD/OpenBSD)

To mount cdrom images in NetBSD and OpenBSD, you will want to use vnconfig. To create a device file that is linked to your image, run the following:

```
# vnconfig vnd0 /path/to/image.iso
```

Then it will be possible to mount this device as you would a normal cdrom:

```
# mount -t cd9660 /dev/vnd0a /mnt
```

Once that is finished, you may work with the image as you would any other cdrom filesystem.

B-v. Naming your tftp boot file.

This is a short and simple guide for how to name your Sun's tftp boot file, which needs to be named in a specific way.

Sun workstations require the tftp boot file to be named for their IP address represented in hexadecimal followed by their platform group.

First, you need to determine your IP address in hexadecimal. This can be done with the bc command.

Secondly, you need to determine your platform group. See Appendix B-i for more information. For the sake of this demonstration, I'll be choosing the sun4u platform group.

```
# bc
obase=16
192
C0
168
A8
0
0
10
A
```

Thirdly, you must symbolically link the boot file to your /tftpboot directory.

```
# ln -s /path/to/boot/file /tftpboot/C0A8000A.SUN4U
```

Appendix C: Cited works and further reading

C-i. Cited works

All cited works are also linked in the document.

Installing Solaris Over the Network - hosted on tenox.pdp-11.ru

FAQ: Frequently Asked Questions about Sun NVRAM, IDPROM, hostid

SunOS 4.1 Diskless Installation Guide

C-ii. Further reading

NetBSD Diskless HOW-TO pages

diskless(8) OpenBSD man page