

# SGI Indigo<sup>2</sup> hosted by Linux!

Kjetil Erga

September 8, 2012

## 1 Introduction

I have a Silicon Graphics Indigo<sup>2</sup> system that had a failing harddisk. I got lucky and managed to create a copy the filesystems before it failed completely. After this, I experimented on getting the Indigo box to boot IRIX 5.3 over the network and get all of it's files using NFS. I was successful, and will provide the configuration files I used here in this document. Two Slackware Linux boxes were used to host everything.

## 2 Extracting the File System

I just want to mention what commands I used to actually extract the file system from the failing disk. I removed the disk from the Indigo box and placed it in another Linux box with a compatible SCSI controller. This is the commands I used to copy everything from the IRIX 5.3 "system" partition at */dev/sda1*:

```
root@ace:/tmp# mkdir /tmp/irix
root@ace:/tmp# mount -t efs /dev/sda1 /tmp/irix
mount: warning: /tmp/irix seems to be mounted read-only.
root@ace:/tmp# cd /tmp
root@ace:/tmp# rsync -av --hard-links /tmp/irix /tmp/irix-copy
root@ace:/tmp# tar -cvzpf irix-copy.tgz irix/
```

There was also a "user" partition at */dev/sda7* that I copied with the same commands.

## 3 The Boxes (Machines)

There are three boxes involved in this scheme:

- 192.168.0.1 - Linux Box 1 - DNS and DHCP host.
- 192.168.0.46 - Linux Box 2 - NFS and TFTP host.
- 192.168.0.50 - SGI Indigo<sup>2</sup>

The scavenged file system has been unpacked on a USB memory stick and mounted on the second Linux box at */mnt/usb/irix*.

## 4 DHCP and DNS Configuration

The first Linux box at 192.168.0.1 uses *dnsmasq* to serve both DHCP boot information and DNS records. Here is the important configuration from */etc/dnsmasq.conf* that causes the correct boot information to be sent to the Indigo box:

```
dhcp-host=08:00:69:08:b8:8d,192.168.0.50,infinite,set:irix
dhcp-boot=net:irix,/mnt/usb/irix/unix,boothost,192.168.0.46
```

This causes the Indigo box with MAC address 08:00:69:08:b8:8d to be given an IP address and instructions to download a boot image at /mnt/usb/irix/unix from 192.168.0.46.

Here is the lines from */etc/hosts* on the first Linux box that shows the hostnames of the other boxes:

```
192.168.0.46          aone.private.net    aone
192.168.0.50          indigo.private.net  indigo
```

## 5 TFTP Configuration

The second Linux box at 192.168.0.46 has a TFTP daemon enabled in */etc/inetd.conf* like so:

```
tftp dgram udp wait root /usr/sbin/in.tftpd in.tftpd -v -v -s / -r blksize
```

Note that the whole root directory has been exported. This is a potential security risk, but it is easier when experimenting. The kernel image that the Indigo box downloads with TFTP is located at */mnt/usb/irix/unix*.

## 6 NFS Configuration

The second Linux box at 192.168.0.46 uses a NFS daemon to serve all files from the scavenged IRIX file system. */etc/exports* is setup with an entry like this:

```
/mnt/usb *(rw,sync,no_root_squash,no_all_squash,no_subtree_check)
```

The actual root of the IRIX file system is at */mnt/usb/irix*.

## 7 Boot Parameter Server Configuration

The "diskless" IRIX booting system also requires a special boot parameter daemon to be enabled. This is called *bootparamd* and is normally part of the NFS daemon collection. I had to modify the startup script for NFS to actually enable this. The following was added in */etc/rc.d/rc.nfds* on the second Linux box at 192.168.0.46:

```
if [ -x /usr/sbin/rpc.bootparamd ]; then
    echo " /usr/sbin/rpc.bootparamd"
    /usr/sbin/rpc.bootparamd
fi
```

The boot parameter daemon uses a configuration file at */etc/bootparams*, with the following contents (all on one line):

```
indigo root=aone:/mnt/usb/irix swap=aone:/mnt/usb/irix/swap
sbin=aone:/mnt/usb/irix/sbin
```

The boot parameter daemon is very picky with regards to the contents of */etc/hosts*. These two lines, and only these (except for comments) had to be present on the second Linux box:

```
127.0.0.1          localhost
192.168.0.50       indigo.private.net  indigo
```

By default, there is an entry for 127.0.0.1 that points to the "real" hostname of the machine, this MUST be removed!

## 8 NVRAM Configuration

The Indigo box has a "Command Monitor" shell that can be entered prior to boot. This stores various settings used when booting, in the form of "NVRAM" environment variables. Here is a dump of all the variables on the functional Indigo box:

```
OSLoadPartition=bootp()boothost:/mnt/usb/irix
OSLoader=unix
AutoLoad=yes
TimeZone=PST8PDT
console=g
diskless=1
keybd=NO
dbaud=9600
volume=80
sgilogo=y
autopower=y
netaddr=192.168.0.50
eaddr=08:00:69:08:b8:8d
ConsoleOut=video()
ConsoleIn=keyboard()
cpufreq=200
gfx=alive
```

The important ones that must be set are "OSLoadPartition", "OSLoader", "diskless", "netaddr" and possibly "eaddr".

## 9 IRIX File System Modifications

Since the scavenged file system is from a time when the box booted from it's own hard disk, it must be modified to reflect the new network oriented reality. Remember that all the changes here are actually done on the USB stick at that is mounted at /mnt/usb on the second Linux box.

It seems that IRIX uses information from it's own */etc/hosts* to setup the IP address. Because of this, the following line had to be changed:

```
192.168.0.50    IRIS
```

Failure to do this may cause the NFS server connection to be severed during startup, causing an infinite hang.

I got a lot of error messages during startup because of invalid mount points, so I changed */etc/fstab* to look like this:

```
/dev/root / efs rw,raw=/dev/rroot 0 0
/vswap/vswap1 swap swap pri=5,vlength= 102400 0 0
aone:/mnt/usb/irix/sbin /sbin nfs rw 0 0
aone:/mnt/usb/irix/swap /swap nfs rw 0 0
```