Net Install IRIX From Linux

Requirements

Linux requirements:

DHCP
TFTP
RSH
mksh

IRIX Requirements:

IRIX Disks:

Installation Tools and Overlays
Overlays 2- 3
Foundation 1 - 2
Applications
Complementary Applications
Development Foundation
Development Libraries

Patience

Introduction

Not having SCSI CD-ROM drives for my SGI Octanes, I was not deterred from trying to do an IRIX install. What I do have is a lot of hardware sitting around, and a Saturday evening that I wasn't doing anything. Reading various guides, and feeling rather clever myself, I dove into the task. After some tweaking, I met with success. For the purposes of this guide, the following information is relevant:

SGI Box:
Hostname Octane
IP 192.168.1.11
OS (to be installed) IRIX 6.5.29
MAC 08:00:69:0A:2F:EF

Linux Box:
Hostname Hecate
IP 192.168.1.5

OS Fedora Linux 18
File locations:
/mnt/raptor1/IRIX
/irix (ln -s /mnt/raptor1/IRIX /irix, will save you effort in the future)
firewalld, iptables and selinux are all disabled for this process.

Directions:

Install Media Preparation

Rip an image of each disk. You will need to use an OS that supports EFS to do this. Since I do not have a CD Drive on my server, I used a laptop running Slackware Linux to extract the files.

$ dd if=/dev/cdrom of=~/installTools.iso

Repeat for each disk. I named each image to match the disk name (so I have installTools.iso, overlay2.iso, etc)

Extract the contents of each disk

Mount each ISO image and copy the files out to a folder. Please note that you will need support for EFS in your kernel, which the default Red Hat and Fedora kernels do not (and they didn't understand my question when I asked about it). You may need to build a custom kernel to do this.

$ sudo mount -o loop -t efs ~/installTools.iso /mnt/cdrom
$ mkdir ~/installTools
$ cp -rfv /mnt/cdrom/* ~/installTools
$ sudo umount /mnt/cdrom

Once again, I named each folder to match the disk name. Since I wasn't using that laptop for the installation, I copied the files over to an NFS available directory on my server. My folder looks like this:

[armanox@hecate IRIX]$ ls
applications devFound foundations1 instTools overlay3
compApp devLib02 foundations2 overlay2

DHCP

DHCP Servers are a pain to set up. I'm not going to write in detail about actually setting up a DHCP server, but I will share my configuration

```
[root@hecate ~]# cat /etc/dhcp/dhcpd.conf
#
# DHCP Server Configuration file.
#   see /usr/share/doc/dhcp*/dhcpd.conf.example
#   see dhcpd.conf(5) man page
#
subnet 192.168.1.0 netmask 255.255.255.0 {
        option routers                  192.168.1.1;
        option subnet-mask              255.255.255.0;
        option domain-search            "home";
        option domain-name-servers              192.168.1.1;
        option time-offset              -18000;         # Eastern Standard Time
        pool {
        range dynamic-bootp 192.168.1.1 192.168.1.254;
        }
        allow bootp;
        authoritative;

        host octane {
                hardware ethernet 08:00:69:0A:2F:EF;
                fixed-address 192.168.1.11;
                option host-name "octane";
                next-server 192.168.1.5;
        }

}
```

TFTP

Ah, TFTP. Every net OS install seems to use it. After installing tftp (yum install tftp-server), it has to be enabled and all that. You also have to have it point to where your IRIX install files are located.

```
[root@hecate ~]# cat /etc/xinetd.d/tftp
# default: off
# description: The tftp server serves files using the trivial file transfer \
#       protocol.  The tftp protocol is often used to boot diskless \
#       workstations, download configuration files to network-aware printers, \
#       and to start the installation process for some operating systems.
service tftp
{
        socket_type             = dgram
        protocol                = udp
```

```
        wait            = yes
        user            = root
        server                  = /usr/sbin/in.tftpd
        server_args             = -s /mnt/raptor1/IRIX/
        disable                 = no
        per_source              = 11
        cps             = 100 2
        flags           = IPv4
}
```

After editing that file, I had to enable xinetd (systemctl start xinetd.service)

RSH

This was the fun one. My next Nessus scan will throw a fit over this one. The IRIX installer uses rsh locate and grab its file during the install process. First install rsh-server
$ sudo yum install rsh-server

Next, configure it.

```
# default: on
# description: The rshd server is the server for the rcmd(3) routine and, \
#       consequently, for the rsh(1) program.  The server provides \
#       remote execution facilities with authentication based on \
#       privileged port numbers from trusted hosts.
service shell
{
        socket_type             = stream
        wait            = no
        user            = root
        log_on_success          += USERID
        log_on_failure          += USERID
        server                  = /usr/sbin/in.rshd
        disable                 = no
}
```

And rlogin

```
[root@hecate ~]# cat /etc/xinetd.d/rlogin
# default: on
# description: rlogind is the server for the rlogin(1) program.  The server \
#       provides a remote login facility with authentication based on \
#       privileged port numbers from trusted hosts.
```

```
service login
{
        socket_type             = stream
        wait            = no
        user            = root
        log_on_success          += USERID
        log_on_failure          += USERID
        server                  = /usr/sbin/in.rlogind
        disable                 = no
}
```

Also make sure to add rlogin and rsh to the end of /etc/securetty. Then we need to actually enable the logins. For this process I'm making them passwordless, and this is very insecure.

Make the following match changes to /etc/hosts.equiv and /root/.rhosts

```
[root@hecate ~]# cat /etc/hosts.equiv
+
[root@hecate ~]# cat /root/.rhosts
+ root
```

Restart xinetd after making the changes.

KSH

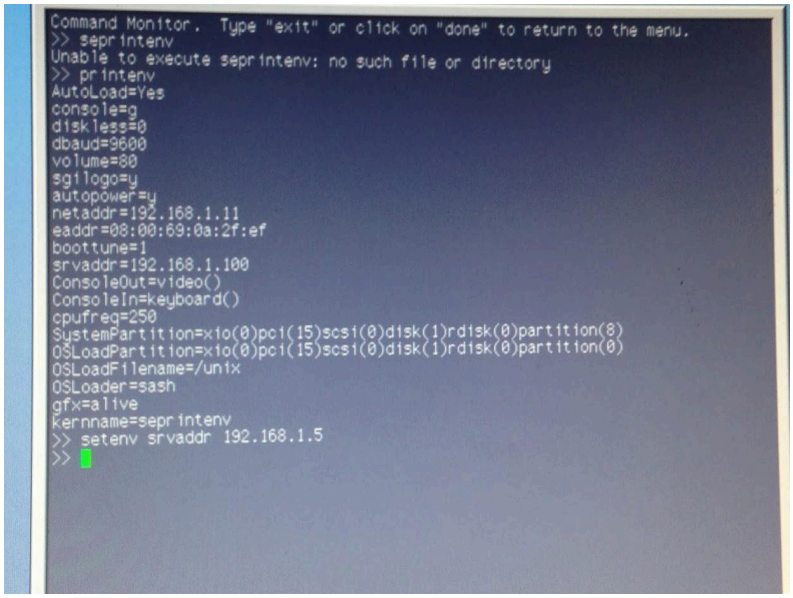
IRIX requires KSH to complete this task. I found mksh to work, other Korn Shells may or may not.

I also had to link /usr/bin/mksh to /bin/sh for the duration of the install.

Installation:

Boot box into PROM (hit escape). Go into the command prompt

setenv srvaddr 192.168.1.5

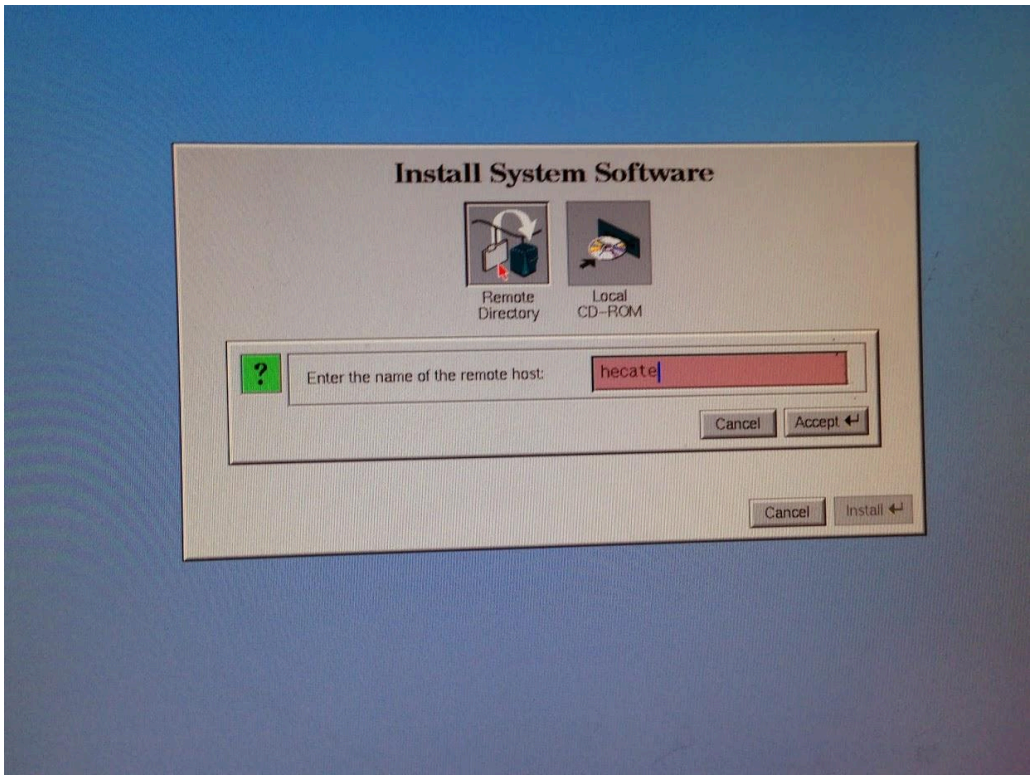


boot -f bootp()192.168.1.5:/instTools/stand/fx.64 -x

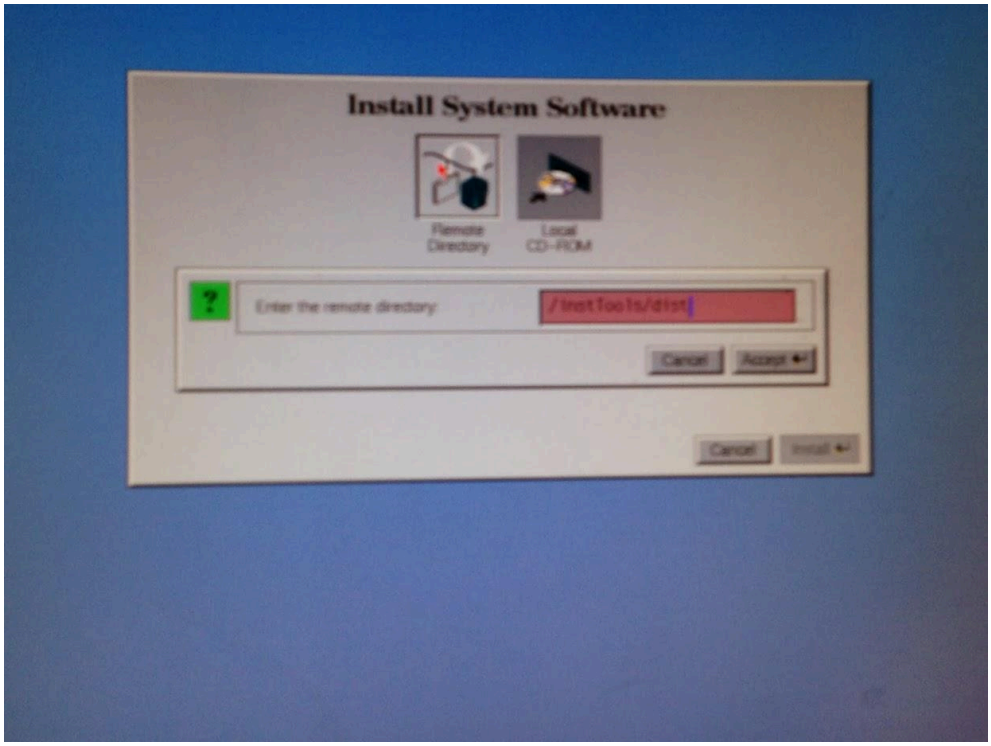
Repartition drive (TODO: Actually add instructions)

Click done

Click Install System Software (2)
Select Remote Directory



Enter your server here (IP or hostname, DNS works on my network)
Enter your remote directories relative to TFTP root (so, in my example, it becomes /instTools/dist). Your installation tools and overlay 1 disk should be selected for the from field, the rest should just be opened. Follow standard install directions from here.



During the install, for adding more dists, in my example I used hecate:/irix/<foldername>/dist for each new dist. These are relative to / on your server, not relative to the TFTP root.