



Installing Linux on a ~~Dead~~ Badger Octane MIPS R12000.



1. Prepare Gentoo box for netbooting (Gentoo Install, DHCP/BOOTP server install, TFTP server install with netboot image)

Install Gentoo Linux

```
#emerge net-ftp/tftp-hpa
```

Download ip30r10k.img to /tftpboot

```
#emerge dhcp
```

```
#nano -w /etc/dhcp/dhcpd.conf
```

```

    ddns-update-style none;
    subnet 192.168.1.0 netmask 255.255.255.0 {
    pool {
        range dynamic-bootp 192.168.1.209 192.168.1.229;
    }
    option domain-name-servers 192.168.1.254;
    option routers 192.168.1.254;
    authoritative;
    allow bootp;
    host sgi {
        hardware ethernet 08:00:69:08:db:77;

        # TFTP Server to download from (same as DHCP server)
        next-server 192.168.1.200;

        # IP address to give to the SGI machine
        fixed-address 192.168.1.209;

        # Filename for the PROM to download and boot
        filename "/ip30r10k.img";
    }
}

```

```
#/etc/init.d/dhcpd start ; /etc/init.d/in.tftpd start
```

```
#/rc-update add dhcpd default ; rc-update add in.tftpd default
```

2. Set the IP Address on the Octane at the boot firmware prompt:

```
>>setenv netaddr 192.168.1.209
```

3. Netboot the Octane at the boot firmware prompt:

```
>>bootp(): root=/dev/ram0
```

This will obtain a temporary IP configuration from the DHCP/BOOTP server, obtain the linux image file using TFTP and start the system using it in a RAMdisk (/dev/ram0).

4. Once the Octane is netbooted, you can configure it for networking:

```
# net-setup 192.168.1.209 192.168.1.254
```

```
//IP is 209, GW is 254, with sshd (by default)
```

```
# ifconfig
```

```
# passwd
```

```
//assigns root passwd for ssh access
```

```
# nano -w /etc/resolv.conf
```

```
//add nameserver 192.168.1.254
```

To open a remote terminal session from another Linux box, type **ssh 192.168.1.209** (if it complains about the key, type **rm ~/.ssh/known_hosts** first)

5. Configure hard disk partitions:

```
# dd if=/dev/zero of=/dev/sda bs=512 count=1 //zero-out old SGI disklabel
# fdisk /dev/sda //create SGI disklabel & partitions
//as shown in Gentoo Handbook
# mke2fs /dev/sda1 //this formats /boot partition as ext2
# mke2fs -j /dev/sda3 //this formats / partition as ext3
# mkswap /dev/sda2 //this prepares the SWAP partition
# swapon /dev/sda2
```

6. Mount partitions:

```
# mkdir /mnt/gentoo
# mount /dev/sda3 /mnt/gentoo
# mkdir /mnt/gentoo/boot
# mount /dev/sda1 /mnt/gentoo/boot
```

7. Get structure and files for / filesystem (100MB) and the portage snapshot (30MB) as well as configure the portage options in make.conf:

```
# date MMddHHmmYYYY //specify UTC time (i.e UTC=EST + 5)
# cd /mnt/gentoo
# wget -c http://distfiles.gentoo.org/releases/mips/current/stages/stage3-mips4-2007.0.tar.bz2
# tar -xjpf stage3-mips4-2007.0.tar.bz2
# wget -c http://mirror.csclub.uwaterloo.ca/gentoo-distfiles/snapshots/portage-2.1.4.tar.bz2
# tar -xjf portage-2.1.4.tar.bz2 -C /mnt/gentoo/usr/
# vi /etc/make.conf //add USE="ip30"
//add MAKEOPTS="-j2"
```

8. Chroot to the / filesystem and prepare environment:

```
# cp -L /etc/resolv.conf /mnt/gentoo/etc/resolv.conf
# mount -t proc none /mnt/gentoo/proc/
# mount -o bind /dev /mnt/gentoo/dev
# chroot /mnt/gentoo /bin/bash
# env-update
# source /etc/profile
# export PS1="(chroot) $PS1"
# date MMddHHmmYYYY //again, specify UTC time (i.e UTC=EST + 5)
```

9. Update portage tree and choose system profile:

```
# emerge --sync //run emerge-webrsync if it fails
# rm /etc/make.profile
# ln -s /usr/portage/profiles/default-linux/mips/2007.1-dev/ip30/o32 /etc/make.profile
# emerge portage //installs latest upgrades to the portage tree
```

10. Get kernel source and kgcc64, compile kgcc64, compile kernel and modules:

```
# ls /usr/share/zoneinfo
# cp /usr/share/zoneinfo/Canada/Eastern /etc/localtime
# emerge mips-sources //13 minutes
# emerge kgcc64 //50 minutes
# ls -l /usr/src/linux
# cd /usr/src/linux
# zcat /proc/config.gz > .config //get current config
# make oldconfig //accept all defaults
# make menuconfig //unset the cross compiling flag
//turn off initramfs (General)
//disable Odyssey graphics
```

```
# make vmlinux CROSS_COMPILE=mips64-unknown-linux-gnu- //takes about 2 hours
//may need to copy ldscripts dir
//to /usr/src/linux if it fails
# cp vmlinux /boot ; cp vmlinux / ; cp vmlinux /boot/kernel-2.6-20070802.ip30
```

11. Get the system ready for usage:

```
# nano -w /etc/fstab
# nano -w /etc/conf.d/hostname //add HOSTNAME=octane
# nano /etc/conf.d/net //add the following lines
    dns_domain_lo="sgi.com"
    config_eth0=( "192.168.1.209 netmask 255.255.255.0 brd 192.168.1.255" )
    routes_eth0=( "default via 192.168.1.254" )
# rc-update add net.eth0 default
# nano -w /etc/hosts //add the following line
    192.168.1.209 octane.sgi.com octane
# passwd //sets root password
# emerge syslog-ng
# rc-update add syslog-ng default
```

12. Install Arcload boot loader & finish install:

```
# emerge arcload dvhtool
# cd /usr/lib/arcload; ls
# nano arc-octane.cf //ensure root=/dev/sda3 in the file
//sda1 is boot,sda2 is swap, sda3 is root

# dvhtool --unix-to-vh sash64 sash64
# dvhtool --unix-to-vh arc-octane.cf arc.cf
# dvhtool --unix-to-vh /usr/src/linux/vmlinux vmlinux
# dvhtool --print-volume-directory
# exit
# cd
# umount /mnt/gentoo/proc /mnt/gentoo/boot /mnt/gentoo/dev /mnt/gentoo
# reboot //hard reboot may be required
```

13. Change Octane boot variables at the boot firmware prompt & boot Linux:

```
>> setenv OSLoader sash64
>> setenv OSLoadFilename ip30(video,sda,nosmp)
>> setenv SystemPartition scsi(0)disk(2)rdisk(0)partition(8)
>> setenv OSLoadPartition scsi(0)disk(2)rdisk(0)partition(8)
>> boot
```