SN Itanium Beta Engineering Forum: Class Handouts May 16, 2001

Preliminary Information

LIMITED RIGHTS LEGEND

TRADEMARKS

Contents

1 TurboLinux Frontier IA-64 Beta 3 (2001-03-07) Release Notes

This is the third beta release of TurboLinux Frontier IA-64. The distribution is available for download at the TurboLinux ftp site:

```
ftp://ftp.turbolinux.com/pub/ia64/TLF-ia64-Beta3.iso
ftp://ftp.turbolinux.com/pub/ia64/TLF-ia64-Beta3-nox.iso
ftp://ftp.turbolinux.com/pub/ia64/Beta3/{RPMS,SRPMS}
```

The first URL points to the ISO 9660 image which you can burn on a CD-ROM. This will give you a bootable CD from which to install the TurboLinux IA-64 distribution on your system. This image contains a graphical installer which is used to install the contents of the CD.

The second ISO image contains a script-based installer which can be used if you have problems starting the X server during the install process. You can also use this disk as a “rescue disk”.

The third URL gives the locations of the binary and source RPMs. Updated RPMs for this release will be available at:

```
ftp://ftp.turbolinux.com/pub/ia64/Beta3/updates
```

IMPORTANT: The C-library and tool-chain in this release contain ABI changes that are not backward compatible with previous TurboLinux IA-64 releases that were based on glibc 2.1.

If you are using this release, we highly recommend that you subscribe to our ia64-dist mailing list. All TurboLinux ia64 related questions should be directed there. For instructions on how to subscribe, please visit:

```
http://www.turbolinux.com/mailman/listinfo/ia64-dist
```

WHAT'S NEW
==========

* The kernel is 2.4.2 + 02/28 IA-64 patch. The default SMP and UP kernels were compiled with workarounds for B- and C-stepping processors only. There is no support for A stepping processors. Also, the VHPT is disabled by default.

* The kernel now contains the new kernel debugger code from Keith Owens.

* Latest version of the gnupro teach-in (2.96-ia64-000717, snapshot 1117) with all recent patches applied.

* Updated version of gdb and strace.

* Updated XFree 4.0.2. This version is known to work on the following cards:

Working cards

(PCI) VGA compatible controller: ATI Technologies Inc Rage XL (rev 39)
(AGP) VGA compatible controller: ATI Technologies Inc Rage 128 PF (rev 0)
(PCI) VGA compatible controller: ATI Technologies Inc 3D Rage IIC 215IIC [Mach64GT IIC] (rev 58)

(AGP) VGA compatible controller: 3Dfx Interactive, Inc. Voodoo 3 (rev 1)
(PCI) VGA compatible controller: ATI Technologies Inc Rage 128 RE (rev 0)
(AGP) VGA compatible controller: S3 Inc. Savage 4 (rev 2)
(AGP) VGA compatible controller: Matrox Graphics, Inc. MGA G200 AGP (rev 3)
(PCI) VGA compatible controller: ATI Technologies Inc Rage 128 RK (rev 0)
(AGP) Nvidia Quadro
(AGP) VGA compatible controller: Matrox Graphics, Inc. MGA G400 AGP (rev4) *

* X starts using 99.9% of CPU after displaying Gnome welcome screen. Using Option "noaccel" allows the server to start.

Other cards may also work, but haven't been explicitly tested. The installer will currently only recognize the above cards. If you have a different card, you will have to manually select it during the X configuration.

RECOMMENDED BIOS VERSIONS
=========================

1) Workstations (BigSur)
This release has been tested with BIOS 89B and 99.

2) Servers (Lion)
We recommend that you upgrade to BIOS version 71.

KNOWN PROBLEMS
==============

1) IMPORTANT. The settings for the QLogic SCSI adapter on Lions MUST be changed to support >4GB addressing. If this is not done, the install will fail. To change the setting, press ALT-Q when the QLogic BIOS screen comes up. Next, select the following menu items:

```
  Configuration settings
    Host adapter settings
      >4GByte addressing
```

Save your changes, and exit from the QLogic configuration program.

You don't have to enable >4G addressing on BigSur systems.

2) The integrated EEPRO network adapter in Lion systems currently suffers from the following issues:

- If you have more than 1GB of RAM, networking will fail almost immediately after you start using it. To get around this, pass mem=1024MB as an argument to lilo.

- Transferring certain files from an NFS server to the Lion will cause NFS to hang. The rest of the networking subsystem will continue to operate normally. This is being looked into by Intel. In the mean time, either don't transfer the problem files via NFS, or install an alternate network adapter.

3) Due to some unresolved issues with the current tool-chain, the following packages are not available in this release:

- ispell

GRAPHIC INSTALLATION INSTRUCTIONS
=================================

To install using the new Stardust installer, take either the boot floppy or the Stardust CD and insert it into your system. When the system starts up, you will see the “Shell>” prompt.

Restrict the shell path to the fsX device which is mapped to CD-ROM. This device mapping is not static (it depends on the number of bootable partitions the EFI shell finds on LS-120 and hard disks on the system). For example, if the shell reports fs1 mapped to CD-ROM you should

```
set path fs1:\
fs1:
install
```

You will next see the “Welcome” screen. Hit <ENTER> to move to the next screen. You will be asked to pick a keyboard. Select the appropriate keyboard for your system and press ENTER.

After keyboard selection, the system will attempt to configure a number of system devices and will also probe for PCI devices on your system. This may take anywhere from a second (for systems without PCI devices) to almost a minute (for systems with lots of devices and drivers that take some time to load). Once the drivers menu comes up, it will display the PCI storage and network devices for your system, and the drivers for these devices. It will also display whether a driver for a particular device was loaded or not.

Loading device drivers from a floppy disk is currently not supported.

The next screen will be the installation selection menu. Here you can specify CD-ROM or NFS installation.

For CD-ROM installations, hit <ENTER> when asked to insert the CD into the drive.

For NFS installations, you must specify the network configuration information. You can select either DHCP or a static network address. Once the network is configured for NFS installation, you will see a window that requests parameters for NFS. NFS installations require an NFS server and an NFS path to the TurboLinux distribution.

In the case of CD-ROM or NFS installations, the file-system (local for CD-ROM and remote for NFS) with the installer is mounted to /tmp/tlimage.

To set up NFS install server, you need to create a directory tree structure as follows:

```
                Top (top directory)
                 |
              Turbolinux
                 |
    ----------------------------------
    |            |           |
  RPMS       instimage     pkginfo
```

Populate these directories with the similar tree structure from the install CD-ROM. Then, export the top directory so others can mount to it.

After this step is complete, there will be a series of GUI utilities that navigate you through the rest of the installation.

In the very first GUI window, you need to select the installation type (again). Then, select a system to be installed ("Development System", "Xfree" or "All in one...", etc.). If "Advanced" is selected, a list of available packages can be manually selected for installation.

The disk partitioning utility displays a list of available disk devices to be partitioned for the system. Select a disk device (ex./dev/sda) for auto-partitioning and click "Next" to go to the next window.

NOTE: auto-partition DESTROYS ALL DATA on the selected device. The auto-partitioning will create a 125MB first primary partition for EFI partition (accessible from Linux as /boot/efi). It will then create two extended partitions for Linux root and Linux swap (512MB), mounted as the fifth and sixth (extended) partition.

If you want to do manual partitioning or if your disk is already partitioned and you wish to preserve the existing setup choose "Advanced" option to go to manual partitioning screen. You will need to specify two mount points -- the EFI partition should be mounted on /boot/efi and Linux root partition should be mounted on /.

In the format utility window, the newly created partitions are selected to be formatted by default. If you want to preserve the original data, please uncheck the selected partitions. Selecting "Next" will start the formatting process and proceed into the packager utility's installation window. Each packages name is displayed as it gets installed as well as the information on the total installation's size and time.

After all the packages are installed, the "Root Password" window will prompt for the new root password. Enter the new password here and select "next" to proceed into the configuration phase.

There will be some auto-detection of the keyboard and mouse in the next several windows. Then, double click on a monitor model that matches your actual monitor. Also, select the display mode and pixel depth. Then select "next" to proceed to the network configuration utility.

Answer "Yes" to start configuring the network. "No" can be selected for a stand alone system. Follow the instruction to configure either DHCP or static IP.

Next, in the configuring time zone window, select the appropriate time zone from the list and then select "Timeserver" to set up a time server that your system needs to sync up the time with. The default time zone is set to US/Pacific. Select "Next" to proceed to the user configuration utility.

Here you can add new users, modify the existing users information or select "Next" to proceed.

After the system reboots, you can start the system by running the linux.nsh script.

SCRIPT-BASED INSTALLATION
==========================

Insert the CD-ROM and power up the system. When you see the EFI shell prompt restrict the shell path to the CD-ROM device (see above). Then boot the kernel and pass it the root= option pointing to the CD-ROM. For example, if your CD-ROM is shown mounted on fs1 and it is connected as the secondary IDE master (default setup on Lion systems) do a:

```
  set path fs1:\
  eli vmlinux root=/dev/hdc
```

The root= argument in the above example assumes that your CD-ROM is attached as IDE secondary master. If this is not the case you will have to specify correct CD-ROM path. If your CD-ROM is:

- primary master:root=/dev/hda
- primary slave:root=/dev/hdb
- secondary master:root=/dev/hdc
- secondary slave:root=/dev/hdd

After the kernel finishes booting, the first four virtual consoles are active. The first one runs the installer and the next three have shell prompts. You can switch between virtual consoles by pressing ALT-F[1-4].

The installer now runs through the install process and gives some help along the way. Be sure to answer questions with yes -- y won't work. The installer performs the steps outlined below.

1. Partition hard disk

The first partition will be used for the bootloader and must be a FAT16 partition. You may want up to 100MB, especially if you plan on building a few kernels. Also create a linux swap and a linux partition for root. This is a sample fdisk session where keyboard input is enclosed in _underscore_:

```
The number of cylinders for this disk is set to 8761.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): _p_
Disk /dev/sda: 64 heads, 32 sectors, 8761 cylinders
Units = cylinders of 2048 * 512 bytes
   Device Boot    Start       End    Blocks   Id  System

Command (m for help): _n_
Command action
   e   extended
   p   primary partition (1-4)
_p_
```

```
Partition number (1-4): _1_
First cylinder (1-8761, default 1): _<CR>_
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-8761, default 8761): _+100M_

Command (m for help): _t_
Partition number (1-4): _1_
Hex code (type L to list codes): _L_
  0 Empty           17 Hidden HPFS/NTF 5c Priam Edisk     a5 BSD/386
  1 FAT12           18 AST Windows swa 61 SpeedStor       a6 OpenBSD
  2 XENIX root      1b Hidden Win95 FA 63 GNU HURD or Sys a7 NeXTSTEP
  3 XENIX usr       1c Hidden Win95 FA 64 Novell Netware b7 BSDI fs
  4 FAT16 <32M      1e Hidden Win95 FA 65 Novell Netware b8 BSDI swap
  5 Extended        24 NEC DOS         70 DiskSecure Mult c1 DRDOS/sec (FAT-
  6  FAT16          3c PartitionMagic 75 PC/IX           c4 DRDOS/sec (FAT-
  7 HPFS/NTFS       40 Venix 80286     80 Old Minix       c6 DRDOS/sec (FAT-
  8 AIX             41 PPC PReP Boot   81 Minix / old Lin c7 Syrinx
  9 AIX bootable    42 SFS             82 Linux swap      db CP/M / CTOS / .
  a OS/2 Boot Manag 4d QNX4.x          83 Linux           e1 DOS access
  b Win95 FAT32     4e QNX4.x 2nd part 84 OS/2 hidden C: e3 DOS R/O
  c Win95 FAT32 (LB 4f QNX4.x 3rd part 85 Linux extended e4 SpeedStor
  e Win95 FAT16 (LB 50 OnTrack DM      86 NTFS volume set eb BeOS fs
  f Win95 Ext'd (LB 51  OnTrack DM6 Aux 87  NTFS volume set f1  SpeedStor
 10  OPUS           52  CP/M           8e  Linux LVM       f4  SpeedStor
 11  Hidden FAT12    53  OnTrack DM6 Aux 93  Amoeba          f2  DOS secondary
 12  Compaq diagnost 54  OnTrackDM6      94  Amoeba BBT      fd  Linux raid auto
 14  Hidden FAT16 <3 55  EZ-Drive        9f  BSD/OS          fe  LANstep
 16  Hidden FAT16    56  Golden Bow      a0  IBM Thinkpad hi ff  BBT

 Hex code (type L to list codes): _6_
Changed system type of partition 1 to 6 (FAT16)
```

Using the commands above, create Linux swap and Linux root partition. Allow at least 200-300 MB for the swap partition and leave the rest for Linux root. Don't forget to change partition types for Linux swap. Sample configuration on our system looks like this:

```
Command (m for help): _p_

Disk /dev/sda: 64 heads, 32 sectors, 8761 cylinders
Units = cylinders of 2048 * 512 bytes
    Device Boot    Start       End    Blocks   Id System
    /dev/sda1            1       101    103408    6 FAT16
    /dev/sda2          102       402    308224   82 Linux swap
    /dev/sda3          403      8761   8559616   83 Linux

Command (m for help): _w_
```

If you already have your hard disk partitioned exit fdisk with _q_ command.

2. Initialize Linux root partition

This step will initialize the Linux root partition and install all the packages, both IA-64 and IA-32. It also sets up a few things on the disk, like the root passwd. It will prompt for the root partition number. Once the packages have been installed, it will ask for network config information.

3. Initialize the EFI partition

4. Initialize the swap partition

5. Reboot

After your system comes back into the EFI shell you should see the newly created EFI partition on your hard disk. There will be two Linux kernel images, one for the uniprocessor kernel [vmlinux] and one for the SMP kernel [vmlinux.smp]. You can boot SMP kernel on a UP system no problem. You can boot UP kernel on a SMP system but extra CPUs will be ignored. For example, if you want to boot SMP kernel and your Linux root partition is on SCSI disk 1 partition 3 you would type

```
   Shell> eli vmlinux.smp root=/dev/sda3
```

If you want to build a custom kernel you will have to copy it on the EFI partition which is mounted under /boot/efi. Don't copy kernels with cp because this results in data corruption. Copy your kernel as follows:

```
  # cat mylinux >/boot/efi/mylinux
```

If your want to run X-Windows on you machine, you will need to create a valid XF86Config file in /etc/X11. Example configuration files for the Rage 128 and the Rage 128 XL can be found at the following URL:

  ftp://ftp.turbolinux.com/pub/ia64

You can also try running:

```
  X -configure
```

to let XFree detect your card and create a basic configuration file for you.

CONTACT INFORMATION:
===================

Support Email: ia64-dist@turbolinux.com

TurboLinux Inc.
2000 Sierra Point Parkway #402
Brisbane, CA 94005
(650)244-7777

http://www.turbolinux.com

2 #ifcfg-eth0 generated by TurboLinux Installation

```
DEVICE=eth0
IPADDR=163.154.10.26
NETMASK=255.255.255.128
NETWORK=163.154.10.0
BROADCAST=163.154.10.127
ONBOOT=yes
```

3 modules.conf

```
alias net-pf-6 off

alias parport_lowlevel parport_pc

alias eth0 eepro100

alias scsi_hostadapter qla1280

alias sound-slot-0 cs4281
```

4 Network

```
# /etc/sysconfig/network generated by TurboLinux installation

NETWORKING=yes

PROFILENAME="No_Profile"

HOSTNAME=bigsur0

DOMAINNAME=engr.sgi.com

GATEWAY=163.154.10.1

GATEWAYDEV=eth0

FORWARD_IPV4=no

IPX=no

TIMESERVERATBOOT=no

TIMESERVERTYPE=ntp

TIMESERVERHOST=(none)

TIMESERVERRESYNC=(none)
```

5 rc.sysinit

```
#! /bin/sh
#
# /etc/rc.d/rc.sysinit - run once at boot time
#
# Taken in part from Miquel van Smoorenburg’s bcheckrc.
#

# Set the path
PATH=/bin:/sbin:/usr/bin:/usr/sbin
export PATH

devfsd /dev

# Read in config data.
if [ -f /etc/sysconfig/network ]; then
    . /etc/sysconfig/network
else
    NETWORKING=no
    HOSTNAME=localhost
fi

# Set the hostname.
hostname ${HOSTNAME}
echo hostname: ‘hostname‘

# Set the NIS domain name
if [ -n "$NISDOMAIN" ]; then
    domainname $NISDOMAIN
else
    domainname ""
fi

if [ -f /fsckoptions ]; then
fsckoptions=‘cat /fsckoptions‘
    else
fsckoptions=’’
fi

if [ ! -f /fastboot ]; then
echo "Checking root filesystems."
fsck -C -V -a $fsckoptions /

rc=$?

# A return of 2 or higher means there were serious problems.
if [ $rc -gt 1 ]; then
echo
echo
echo "*** An error occurred during the file system check."
echo "*** Dropping you to a shell; the system will reboot"
echo "*** when you leave the shell."

PS1="(Repair filesystem) \#"; export PS1
sulogin
```

```
echo "Unmounting file systems"
umount -a
mount -n -o remount,ro /
echo "Automatic reboot in progress."
reboot
elif [ "$rc" = "1" -a -x /sbin/quotacheck ]; then
echo "Checking root filesystem quotas"
/sbin/quotacheck -v /
fi
fi

if [ -x /sbin/quotaon ]; then
    echo "Turning on user and group quotas for root filesystem"
    /sbin/quotaon /
fi

# check for arguments

mount -t proc /proc /proc
if grep -i nopnp /proc/cmdline >/dev/null ; then
    PNP=
else
    PNP=yes
fi

# set up pnp
if [ -x /sbin/isapnp -a -f /etc/isapnp.conf ]; then
    if [ -n "$PNP" ]; then
echo "Setting up ISA PNP devices"
/sbin/isapnp /etc/isapnp.conf
    else
echo "Skipping ISA PNP configuration at users request"
    fi
fi

# Remount the root filesystem read-write.
echo "Remounting root filesystem in read-write mode."
mount -n -o remount,rw /

if [ ! -f /etc/HOSTNAME ]; then
    echo ${HOSTNAME} > /etc/HOSTNAME
fi

# Clear mtab
>/etc/mtab

# Enter root and /proc into mtab.
mount -f /
mount -f /proc

if [ -f /proc/ksyms ]; then
    USEMODULES=y
else
    USEMODULES=
fi
```

```
if [ -x /sbin/depmod -a -n "$USEMODULES" ]; then
    # Get ready for kerneld if module support in the kernel
    echo -n "Finding module dependencies... "
    depmod -a
    echo "done"
fi

# load sound modules
if ! grep -i nomodules /proc/cmdline >/dev/null ; then
    if [ -n "$USEMODULES" ]; then
if grep -s "alias tlsound" /etc/modules.conf > /dev/null ; then
    modprobe tlsound
    if grep -s "alias tlmidi" /etc/modules.conf > /dev/null ; then
modprobe tlmidi
    fi
fi
    fi
fi

if [ -x /sbin/kerneld -a -n "$USEMODULES" ]; then
    if [ -f /proc/sys/kernel/modprobe ]; then
# /proc/sys/kernel/modprobe indicates built-in kmod instead
echo "/sbin/modprobe" > /proc/sys/kernel/modprobe
    else
/sbin/kerneld
KERNELD=yes
    fi
fi

# Add raid devices
if [ -f /proc/mdstat -a -f /etc/raidtab -a -x /sbin/raidadd ]; then
echo "Starting up RAID devices."
raidadd -a

rc=$?

if [ $rc = 0 ]; then
raidrun -a
rc=$?
fi

# A non-zero return means there were problems.
if [ $rc -gt 0 ]; then
echo
echo
echo "*** An error occurred during the RAID startup"
echo "*** Dropping you to a shell; the system will reboot"
echo "*** when you leave the shell."

PS1="(RAID Repair) \#"; export PS1
sulogin

echo "Unmounting file systems"
umount -a
mount -n -o remount,ro /
echo "Automatic reboot in progress."
reboot
```

```
fi
fi

# Check filesystems
if [ ! -f /fastboot ]; then
echo "Checking filesystems."
fsck -C -R -A -V -a $fsckoptions

rc=$?

# A return of 2 or higher means there were serious problems.
if [ $rc -gt 1 ]; then
echo
echo
echo "*** An error occurred during the file system check."
echo "*** Dropping you to a shell; the system will reboot"
echo "*** when you leave the shell."

PS1="(Repair filesystem) \#"; export PS1
sulogin

echo "Unmounting file systems"
umount -a
mount -n -o remount,ro /
echo "Automatic reboot in progress."
reboot
elif [ "$rc" = "1" -a -x /sbin/quotacheck ]; then
echo "Checking filesystem quotas"
/sbin/quotacheck -v -R -a
fi
fi

# Mount all other filesystems (except for NFS and /proc, which is already
# mounted). Contrary to standard usage,
# filesystems are NOT unmounted in single user mode.
echo "Mounting local filesystems."
mount -a -t nonfs,proc

# Save crash dump data (if any)
if [ -x /sbin/vmdump ]; then
        echo "Configuring system to save crash dumps"
        /sbin/vmdump config
fi

# Save crash dump data (if any)
if [ -x /sbin/vmdump ]; then
        echo "Saving crash dump data (if any)"
        /sbin/vmdump save
fi

# Start up swapping.
echo "Activating swap partitions"
swapon -a

# set the console font
if [ -x /sbin/setsysfont ]; then
    /sbin/setsysfont
```

```
fi

if [ -x /sbin/quotaon ]; then
    echo "Turning on user and group quotas for local filesystems"
    /sbin/quotaon -a
fi

# Clean out /etc.
rm -f /etc/mtab~ /fastboot /fsckoptions
>/var/run/utmp

# Delete UUCP lock files.
rm -f /var/lock/LCK*

# Delete stale subsystem files.
rm -f /var/lock/subsys/*

if [ -n "$KERNELD" ]; then
    touch /var/lock/subsys/kerneld
fi

# Delete stale pid files
rm -f /var/run/*.pid

# Delete X locks
rm -f /tmp/.X*-lock

# Delete Postgres sockets
rm -f /tmp/.s.PGSQL.*

# Set the system clock.
echo -n "Setting clock"

ARC=0
UTC=0
if [ -f /etc/sysconfig/clock ]; then
    . /etc/sysconfig/clock

    # convert old style clock config to new values
    if [ "${CLOCKMODE}" = "GMT" ]; then
    UTC=true
    elif [ "${CLOCKMODE}" = "ARC" ]; then
    ARC=true
    fi
fi

if [ -x /sbin/hwclock ]; then
    CLOCKFLAGS="--hctosys"
    CLOCK=/sbin/hwclock
else
    CLOCKFLAGS="-a"
    CLOCK=/sbin/clock
fi

case "$UTC" in
  yes|true)
    CLOCKFLAGS="$CLOCKFLAGS -u";
    echo -n " (utc)"
  ;;
esac

case "$ARC" in
  yes|true)
    CLOCKFLAGS="$CLOCKFLAGS -A";
    echo -n " (arc)"
  ;;
esac

echo -n ": "
$CLOCK $CLOCKFLAGS

date

# Right, now turn on swap in case we swap to files.
echo "Enabling swap space."
swapon -a 2>&1 | grep -v "busy"

# Initialize the serial ports.
if [ -f /etc/rc.d/rc.serial ]; then
. /etc/rc.d/rc.serial
fi

# Load modules (for backward compatibility with VARs)
if [ -f /etc/rc.d/rc.modules ]; then
/etc/rc.d/rc.modules
fi

# Use any system tuning features that user might have installed
if [ -f /etc/rc.d/rc.turbo ]; then
  /etc/rc.d/rc.turbo
fi

# If a SCSI tape has been detected, load the st module unconditionally
# since many SCSI tapes don’t deal well with st being loaded and unloaded
if [ -f /proc/scsi/scsi ] && cat /proc/scsi/scsi | grep -q ’Type:
Sequential-Access’ 2>/dev/null ; then
if cat /proc/devices | grep -qv ’ 9 st’ ; then
if [ -n "$USEMODULES" ] ; then
# Try to load the module.  If it fails, ignore it...
modprobe st 2>/dev/null
fi
fi
fi


# Generate a header that defines the boot kernel.
if uname -a | grep -qi smp ; then
  SMP="1"
  UP="0"
else
  SMP="0"
  UP="1"
fi
cat > /boot/kernel.h << EOF
```

```
/* This file is automatically generated at boot time. */
#ifndef __BOOT_KERNEL_H_
#define __BOOT_KERNEL_H_

#ifndef __BOOT_KERNEL_SMP
#define __BOOT_KERNEL_SMP $SMP
#endif

#ifndef __BOOT_KERNEL_UP
#define __BOOT_KERNEL_UP $UP
#endif

#endif
EOF


# Now that we have all of our basic modules loaded and the kernel going,
# let’s dump the syslog ring somewhere so we can find it later
dmesg > /var/log/dmesg

# Feed entropy into the entropy pool
/etc/rc.d/init.d/random start
```

6 Inittab

```
#
# inittab       This file describes how the INIT process should set up
#               the system in a certain run-level.
#
# Author:       Miquel van Smoorenburg, <miquels@drinkel.nl.mugnet.org>
#               Modified for RHS Linux by Marc Ewing and Donnie Barnes
#Modified for TurboLinux by Christian Holtje
#                                          <docwhat@turbolinux.com>

# Default runlevel. The runlevels used by RHS are:
#   0 - halt (Do NOT set initdefault to this)
#   1 - Single user mode
#   2 - Multiuser, without NFS (The same as 3, if you do not have networking)
#   3 - Full multiuser mode
#   4 - unused
#   5 - X11 With GDM
#   6 - reboot (Do NOT set initdefault to this)
#
id:3:initdefault:

# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit

l0:0:wait:/etc/rc.d/rc 0
l1:1:wait:/etc/rc.d/rc 1
l2:2:wait:/etc/rc.d/rc 2
l3:3:wait:/etc/rc.d/rc 3
l4:4:wait:/etc/rc.d/rc 4
l5:5:wait:/etc/rc.d/rc 5
l6:6:wait:/etc/rc.d/rc 6

# Things to run in every runlevel.
ud::once:/sbin/update

# Trap CTRL-ALT-DELETE
ca::ctrlaltdel:/sbin/shutdown -t3 -r now

# When our UPS tells us power has failed, assume we have a few minutes
# of power left.  Schedule a shutdown for 2 minutes from now.
# This does, of course, assume you have powerd installed and your
# UPS connected and working correctly.
pf::powerfail:/sbin/shutdown -f -h +2 "Power Failure; System Shutting Down"

# If power was restored before the shutdown kicked in, cancel it.
pr:12345:powerokwait:/sbin/shutdown -c "Power Restored; Shutdown Cancelled"

# Run gettys in standard runlevels
1:12345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6

#EOF
```

7 Linux Loader

The Linux Loader is an EFI application. It is itself loaded by the firmware off a FAT partition (on a floppy or hard disk) -- the only filesystem it knows (so far); could be linked with the firmware image (as a ’built-in’ command started by the EFI ’Shell’).

It uses the EFI “Console” and a number of EFI “BootServices”

1. “HandleProtocol” to get the “ShellInterface” which provides the command-line arguments (eg., boot device, kernel image, and root);
2. “LocateHandle” to get the “BlockIo” devices first, then enumerates “HandleProtocol” to get the boot device (where the kernel image is) interface;
3. "ReadBlocks" (not directly a “BootService”, but a 'method' of the interface in (2) as low-level EFI Block I/O) to locate and read the header of the kernel image (provides start address, base address & memory size);
4. “AllocatePages” to get a contiguous space ('mem size' at 'base addr' from (3) -- assumes translation as set in (8)) to contain the kernel image.
5. “ReadBlocks” (actual 'Linux loader')
6. “AllocatePages” to get memory for the "boot params” -- assumes addr at end of IVT
7. “GetMemoryMap” to init “boot params” -- assumes “command line” (eg., “root=/dev/sda2”) at end of mem map
8. “ExitBootServices” then: set rr7, itr0 & dtr0 for kernel, and jump to entry point (from (3))

8 Lilo

LILO.EFI: Linux bootloader for EFI/IA-64 based systems

Stephane Eranian <eranian@hpl.hp.com>
August 4th, 2000 Copyright (C) 2000 Hewlett-Packard Co.

8.1 Introduction

This document describes how to use LILO on IA-64 platforms to load Linux/ia64 kernels. This document describes version LILO.EFI 0.8.

8.2 Command line options

```
lilo [-hVfvqac] [-t sec] [kernel [kernel options...]]
```

-h
Display a list of all possible command line options.

-V
Print the version number and exit.

-t sec
Specify the number of seconds before actually starting to load the kernel image into memory. The default value is 10 seconds.

-f
Disable going to failsafe session. A failsafe session happens when LILO does not find any of the images specified from either the command line or from alternate kernel image capability. By default, LILO will try to load “vmlinux” with no arguments as a last resort.

-v
Turn on verbose mode. LILO prints more message about what it is doing.

-q
Do not ask for a keypress when invoked in interactive mode. The interactive mode is entered when LILO is invoked with command line arguments or when it is interrupted in the middle of loading an image.

-a
Always check for alternate kernel image. When LILO detects that it has been invoked with some command line argument, the default behavior is to NOT look for an alternate image. This option overrides this behavior and LILO is checking for alternate images no matter what.

-c
Do not look for ACPI information. The default behavior is to parse the ACPI tables to build interrupts routing information to pass to the kernel. This option is mostly reserved to debugging.

-i file
Use file as the initial ramdisk (bootstrapping system).

8.3 Booting a kernel

The way an IA-64 system is setup today, you have what's called an EFI bootable partition in the system. The first recognized bootable partition is always mapped to handle fs0. The partition is an MSDOS or VFAT partition from the point of view of the Linux kernel.

To be able to boot a Linux kernel from EFI, you simply need to copy the kernel image onto that partition. This can be easily accomplished using the mtools or by mounting the EFI partition and copying the file.

In the following discussion, we assume that:

- you have a kernel image in the EFI partition called vmlinux

- that partition is called fs0

- you have installed in that partition the lilo.efi executable

So to boot a kernel, simply power cycle the machine. Once you get to the EFI shell prompt, simply go to the first bootable device fs0, by simply doing:

```
Shell> fs0:
```

```
fs0:\>
```

At this point you can invoke LILO:

```
fs0:\> lilo
```

Lilo can be invoked with no arguments. In this case it will:

- pick up the kernel image name vmlinux if it exists, otherwise it will return with an error.

- no argument will be passed to the kernel, except for the BOOT_IMAGE= string (see later for discussion).

You can specify command line options to LILO just like you would do on the PC lilo boot prompt. For instance if you want to specify an alternate root filesystem, you can say:

```
fs0:\> lilo vmlinux root=/dev/sda5
```

You can specify as many parameters as you want. They syntax follows that of the PC LILO, i.e. list of value pairs (or even single values) separated by space.

A more complicated example would be:

```
fs0:\> lilo vmlinux root=/dev/sda2 console=tty console=ttyS0,115200n8
```

In any case, LILO will print which kernel and which arguments to the kernel it is going to use. There is a small countdown for you to stop the loading in case you notice something wrong. By default, this is a 10 seconds timeout. You can adjust this by using the -t option. So a typical output looks like follows:

```
LILO for EFI/IA-64 XX
```

```
lilo: kernel is 'vmlinux'
```

```
lilo: arguments are ’root=/dev/sda2’
lilo: Press ENTER to load or ANY other key to cancel.
vmlinux autoboot in  9 seconds
```

If you press enter, you simply shorten the delay and go straight to loading.

If you press any other key, you enter the ’Interactive Mode’, where you can specify another kernel image and arguments.

[IMPORTANT: In this environment, you can specify any file name you want. Unlike for PC, you are not constrained to a list of predefined kernel image, it can be any file that is accessible from any EFI partitions. Also there IS NO lilo to invoke while running Linux. There is no configuration file.]

At the LILO prompt, simply type in a new kernel name and arguments:

```
LILO boot: vmlinux.se root=/dev/sda2 usbfix
```

Then press enter when you’re done. The next step is for LILO to bring the kernel image into memory. While LILO is doing this it displays a rotating bar (the helicopter) indicating forward progress. At any time during this process, you can interrupt loading by pressing a key. You enter the ’Interactive Mode’ again.

If you want to leave the program, you need to go into interactive mode and then, press CTRL-C, CTRL-D or Escape (Esc) and you will go back to the EFI Shell prompt.

Once LILO is ready to launch kernel execution, by default it will prompt you for a keypress. You can avoid this by specifying the -q option. As of today, you cannot interrupt execution at this point.

In case the specified kernel image cannot be found, LILO will fall back to a failsafe mode and will try to load ’vmlinux’ with nor arguments.

If you don’t want this behavior, then use the -f option.

In order for users to figure out which kernel was booted with which options LILO ALWAYS adds an argument to pass to the kernel. It is the BOOT_IMAGE option. It contains the name of the kernel and arguments. For instance, from the example above, you would get:

```
BOOT_IMAGE=vmlinux.se root=/dev/sda2
```

This is a useful feature during system installation to figure out which kernel was booted.

8.4 The alternate kernel image

Often times when debugging kernels you want to reboot the machine once with your test kernel and, if something goes wrong, you want to fall back to a more stable kernel. In addition you want to be able to do this from remote. Many things can go wrong when doing kernel debugging. It could be that you don’t even reach user-mode. In this case however, you still want to fall back to a stable kernel. The feature you’d like from a bootloader is ’boot once and then fall back to safe kernel’.

This version of LILO offers this feature and it’s called ’alternate kernel image’. You can configure LILO to load a kernel only once and then whatever happens the next reboot falls back to a different kernel hopefully more stable.

To do this, LILO relies on a external file called ’lilo.alt’. The content of this file is very simple: it is whatever you would normally specify on the command line. The file must be present where you have your LILO.

[IMPORTANT: current implementation IMPOSES all the arguments to be on a single line.]

The general theory of operations is that LILO looks for a file ’lilo.alt’. If it finds it then, it takes its content and figures out which kernel and which arguments to use. then it deletes the file, loads the kernel and launch it. Next time, you reboot you fall back to a different kernel unless you recreate the ’lilo.alt’ file.

Let’s take a simple example, the default kernel is still vmlinux with no arguments. The development kernel is vmlinux.se and it needs one argument: usbfix. Then the content of the lilo.alt file must be:

```
# cat lilo.alt

vmlinux.se usbfix
```

You should copy the file where LILO is, i.e. on the EFI partition, using mcopy or cp if filesystem is mounted.

Next, you reboot the system and at the EFI Shell prompt simply type:

```
fs0:\> lilo

LILO for EFI/IA-64 XX

lilo: found alternate file lilo.alt

lilo: kernel is ’vmlinux.se’

lilo: arguments are ’usbfix’

Press ENTER to load or ANY other key to cancel

vmlinux.se autoboot in 5 seconds
```

Even though you went through alternate kernel image, you can still interrupt the loading just like we described earlier.

Using the -a option:

It is very common to have some autoboot sequence where you would specify a default kernel image with some arguments, i.e. like the root filesystem. Then we need to figure out some way of ’prioritizing’ what LILO is going to do in case you have command line arguments and also a lilo.alt file. The rule that we decided to follow is that:

- the arguments specified at the LILO command line take precedence over the alternate file search

This makes sense because you explicitly specify arguments, so why would the next thing you do simply override them?

However it is a problem in a system with autoboot + arguments where you want to do kernel debugging. To overcome the problem we have introduced the -a option which says that LILO MUST always check for the presence of the ’lilo.alt’ file, i.e. override priority.

In this case the behavior is as follows:

- if no lilo.alt file is present, LILO follows the command line arguments
- if lilo.alt file is present, it takes precedence over command line arguments.

8.5 Autobooting the machine

Once you’re satisfied with your machine setup, it is good to install an autoboot procedure. Using EFI, this is easily accomplished using a ’Shell’ script with a special name: ’startup.nsh’.

When the system boots, it looks for bootable partitions and if it finds a ’startup.nsh’ file in ANY of these it will jumpstart execution from it.

So the typical way of autobooting your Linux/ia64 system is to simply create such a file with the following content:

```
# cat /boot/startup.nsh

lilo -a vmlinux root=/dev/sda2
```

Here we show a configuration where the EFI partition is mounted under /boot.

8.6 Credits

Intel Corp.
Stephane Eranian <eranian@hpl.hp.com>
David Mosberger <davidm@hpl.hp.com>
Johannes Erdfelt <jerdfelt@valinux.com>

8.7 Bug reports

This is still beta software and contains some limitations and bugs. Please contact <eranian@hpl.hp.com> if you have any questions.

8.8 Reference

EFI v0.92 specifications are available from the following web site:
http://developer.intel.com/technology/efi/

9 ELI README

This is the EFI Linux boot loader. It currently only supports ia64, but presumably could be modified to handle ia32 as well.

The program will boot kernel images *only* off of the filesystem the executable resides on. This is currently because of the added complexity of supporting the shells notion of filesystem maps.

Images can be specified via the eli.cfg file, or on the command line.

The eli.cfg file is very similar to the configuration file LILO uses. However, it only supports a subset of commands. This is because many of the commands LILO supports do not apply to EFI. Also, some commands which may apply to EFI may not be supported.

The commands which are supported are:

- delay

- timeout

- root (both globally and per image)

- read-only

- read-write

- image

- label

- append

These commands work exactly like they do under LILO.

If an image is specified on the command line, arguments can also be specified. If no arguments are specified, then the config file global options will be used.

Johannes Erdfelt (jerdfelt@valinux.com)

10 Kernel Debugger (KDB) Help -- 4/30/01

```
Kernel panic: VFS: Unable to mount root fs on 08:02

Entering kdb (current=0xe000000000408000, pid 1) on processor 0 due to
panic


[0]kdb> help
Command         Usage                Description
----------------------------------------------------------
md              <vaddr>              Display Memory Contents
mdr             <vaddr> <bytes>      Display Raw Memory
mds             <vaddr>              Display Memory Symbolically
mm              <vaddr> <contents>   Modify Memory Contents
id              <vaddr>              Display Instructions
go              [<vaddr>]            Continue Execution
rd                                   Display Registers
rm              <reg> <contents>     Modify Registers
ef              <vaddr>              Display exception frame
bt              [<vaddr>]            Stack traceback
btp             <pid>                Display stack for process <pid>
bta                                  Display stack all processes
ll              <first-element> <lin Execute cmd for each element in
linked list
env                                  Show environment variables
set                                  Set environment variables
help                                 Display Help Message
?                                    Display Help Message
cpu             <cpunum>             Switch to new cpu
ps                                   Display active task list
reboot                               Reboot the machine immediately
sections                             List kernel and module sections
sr              <key>                Magic SysRq key
bp              [<vaddr>]            Set/Display breakpoints
bl              [<vaddr>]            Display breakpoints
bpa             [<vaddr>]            Set/Display global breakpoints
bph             [<vaddr>]            Set hardware breakpoint
bpha            [<vaddr>]            Set global hardware breakpoint
bc              <bpnum>              Clear Breakpoint
be              <bpnum>              Enable Breakpoint
bd              <bpnum>              Disable Breakpoint
ss              [<#steps>]           Single Step
ssb                                  Single step to branch/call
irr                                  Show interrupt registers
itm                                  Set new ITM value
init                                 Send INIT to cpu
[0]kdb>
```

11 The Linux Devfs (Device File System)

From a slide show by Richard Gooch.

11.1 Introduction to conventional /dev

- Drivers provide access via device nodes
  - Preserves “everything is a file” philosophy
- Device Numbers
  - Major number
  - Minor number
- Linux Implementation
  - Allocated 8 bits each for major and minor
  - Device drivers register “driver operation methods”
  - Methods stored in the “major table”
  - The open(2) system call indexes into the table
  - Minor number is used to determine hardware instance

11.2 The Existing Problem

- Minor numbers often allocated as devices are found
  - Musical devices
- User-space communicates with drivers through two layers of indirection
  - Device number adds a layer
  - Larger device numbers will require a search
- Device nodes are static
  - Have thousands (need millions) of nodes
  - Choice between efficiency and ease of management (still limited)
- Central Authority required for:
  - Device number allocation
  - Device name allocation

11.3 The Existing Problem - The Return

- Information Duplication
- No generic mechanism to search for devices
  - /dev contains no information
- No generic mechanism to handle hot-plug

- Only one way to see the hardware
    - Usually the wrong way
- Running out of device numbers
    - Increase size
    - Don't break libc <6
- We're not good on large systems

11.4 The Solution

- A virtual FS like /proc for devices
- Allow drivers to create device nodes themselves
    - Removes device number indirection
    - Can ignore device numbers and use device instance structures
    - Effectively increases device number space
    - Hot-plug support comes free
    - Easy and natural scanning of devices
- Provide a structured name-space
- Provide multiple name-spaces, exposing:
    - True physical location of devices (vaporware)
    - Driver topology (partly exposes physical location)
    - Classification by device types
    - Consistent references to roaming devices (vaporware)
    - We need them all

11.5 Advantages

- Better naming schemes
- Works at boot
- Works for read-only systems
- Even non-Unix root filesystem
- Preserves Unix “everything is a file” philosophy
- Uniform device scanning at last
- Supports a device management daemon (devfsd)
    - Allows module autoloading
    - Advanced permissions management (even NIS)
    - Access point for hot-plug device management
    - Unified device management for all device types
    - Custom name-spaces can be created

11.6 Advantages - Part II

- Lightweight and unified mechanism
  - 72 bytes per entry
  - Uniform API for drivers
- Vendors can independently manage their name-space
  - Control your own sub-tree
  - Synchronize driver and application releases
- Scalable
  - Small memory footprint on small systems
  - Larger systems have more RAM anyway
- /dev reflects available hardware
- /dev is automatically sized

11.7 Future Work

- Volume name-space
- WWN name-space
- Physical name-space
- Hot-plug user-space device management
- Overhaul block and SCSI layers
  - We need to support more devices at once
- Use dcache for devfs internal database

11.8 Other Alternatives

- Why note just pass device create/remove events to a daemon?
- Just implement a better scsidev
- Put /dev on a ramdisc

12 The New Joy of Init Scripts

From a slide show by Richard Gooch.

12.1 Introduction to Init Scripts

- After kernel has mounted /, init(8) is started
- Init(8) is responsible for:
  - Running init scripts
  - Spawning gettys (login processes)
  - Reaping orphaned processes
  - Catching special signals (i.e. SIGINT for shutdown)
- Two main versions of init(8) available for Linux
  - simpleinit (< 500 lines of code)
  - SysVInit (~5000 lines of bloat)
- Two main schemes available for init scripts:
  - BSD-style
  - SysV-style

12.2 BSD-style Init Scripts

- Master script /etc/rc which orchestrates everything
  - Sequencing explicitly controlled in script
  - May use some helper scripts in /etc/rc.d
  - Local additions placed in /etc/rc.local
- Easy to understand
  - Just read the script to see sequence
  - Hopefully comments will mention dependencies
  - Neat and simple
  - Efficient
- Not scalable
  - Third-party vendors need to edit a script
  - Consequently additions are fragile
  - Package removal is playing with fire
  - No provision for stopping or restarting services

12.3 SysV-style Init Scripts

- Work is performed by many mini-scripts

    - Usually stored in /etc/init.d
    - Each script starts or stops a particular service (daemon)
    - In theory no master script is needed
- Sequencing is done by running some scripts before others
    - Create other directory with symbolic links to real scripts
    - Links are prefixed by “S” and a number
    - Scripts with a lower starting number are started first
    - We've regressed to BASIC line numbers
    - Scripts support starting and stopping services
- Ideal master script:
    - Scans symlink directory and runs each script in turn
    - Shell expansion rules ensure lower numbers are run first
    - Small trap with “S9” and “S10”: need “S0#”
    - In Real Life[tm] “special” work is done inside script

12.4 SysV-style Init Scripts - Part II

- Third-party support:
    - Install mini-script in /etc/init.d
    - Easy to check for name conflicts
    - Create symlink with appropriate (and unused!) number
- Runlevels
    - Boot the system into different states
    - Implemented with runlevel directories
    - Each runlevel directory contains symlinks into /etc/init.d
    - /etc/rc.d/rc0.d is runlevel 0
    - /etc/rc.d/rc1.d is runlevel 1
    - Init(8) passes runlevel to master script
    - Master script scans each directory up to specified runlevel

12.5 SysV-style Init Scripts - Part III

- Limitations
    - How to find “last” script run?
    - Run out of free numbers between existing scripts
    - Runlevels are intrinsically meaningless
    - It's just so damn ugly
    - High confusion factor
    - Linux has turned average computer users in Unix sysadmins

12.6 An Alternative

- Want these features:
  - Scalability of SysV: this means mini-scripts
  - Simplicity of BSD
  - Readability of BSD
  - Enforced declaration of dependencies
- Solution:
  - Abolish master script
  - Make init(8) run all mini-scripts in parallel
  - Dependencies are declared by each script with need(8)
  - Dependencies are resolved by need(8) and init(8)
  - Scripts block in need(8) until what they need is available
  - Pass “runlevel” name to init(8) and select directory
- Results:
  - Fast booting
  - Dependencies are obvious
  - Clean design
  - We can scale without sucking

12.7 Advantages

- We can see what services a script depends on
- Dependency information is localized
  - No need for system integrator to orchestrate everything
- Third-party scripts can fine-tune their dependencies
  - Depend on exactly what you need and no more
- Third-party script installers don't need to know global sequencing
- Some services will start in parallel
  - Can reduce total boot time
  - Need to watch out for more disc thrashing

12.8 Implementation

- Took simpleinit(8) from util-linux and added a database of services
  - Added need(8) which talks to init(8) through FIFO and signals
  - Configure init(8) to “run” a directory
  - All scripts will be run, in parallel
  - Scripts call need(8) and will block until ready

  - Changes rolled back into util-linux
  - < 1000 lines of code in simpleinit(8) now
  - Added other nice but unrelated features
- Implemented sample boot scripts
  - Placed in /sbin/init.d
  - Production quality: I use them all the time
  - Can roll your own, looking at the samples for guidance
- Single-user and runlevels implementation:
  - Based on meaningful names, not silly numbers
  - Can be a script or a directory

12.9 Implementation - Part II

- Reduce disc seek activity by prefetching scripts in FS order
- Support rollback:
  - Can stop services in reverse order they were started
  - Can stop until a script is reached: partial rollback
  - Roll-forward again by running a script or scripts
- Support multiple providers:
  - Scripts call provide(8) to declare what they want to provide
  - Init(8) ensures only one provider is given permission
  - If provider fails, another is given permission
  - Useful for multiple providers of a generic service
    - “mta” provided by sendmail and qmail
    - “httpd” provided by apache and zeus

12.10 Example Script - nfs-export

```
#! /bin/sh
# /sbin/init.d/nfs-export

case "$1" in
  start)
    need portmap || exit 1
    rpc.mountd
    rpd.nfsd
    ;;
  stop)
    killall rpc.nfsd && killall rpc.mountd
    ;;
esac
# End
```

12.11 Example Script - sendmail

```
#! /bin/sh
# /sbin/init.d/sendmail

case "$1" in
  start)
    if [ ! -f /etc/mail/sendmail.cf ]; then exit 2; fi
    provide mta || exit 2
    need portmap
    /usr/sbin/sendmail -bd -q15m
    ;;
  stop)
    killall sendmail
    ;;
esac
# End
```

12.12 Future Work

- Keep full dependency history
  - Could stop a service and stop all that depend on it
  - Could restart a service and all those that depended on it
  - Small increase in complexity
  - Is it worth it?
- Media-order prefetching
  - Kernel hack to record block numbers accessed
  - Turn on recording, extract list when boot completes
  - Re-order block numbers and save in file
  - Next boot, prefetch blocks
  - One single, fast seek and read
  - No more disc accesses

13 Reset Console Output

The following console output is from a P0 (early prototype) SN-IA with 1 C brick and 1 I brick. Two CPUs were enabled, and two CPUs were not connected to power.

```
>> reset
Resetting the system...
INFO: console subchannel changed:  001c24 CPU0
========== Platform errors outstanding A: early main() ==========
NASID 0x0, cpu 0
 0x4000000000000000 PI_INT_PEND1

HARDWARE ERROR STATE:
End Hardware Error State
Booting prom with L1 in PROTOCOL MODE.
INFO: console subchannel changed:  001c24 CPU1
Booting prom with L1 in PROTOCOL MODE.
INFO: console subchannel changed:  001c24 CPU0
NOTICE:  ecc enabled even though CONFIG requested disabled
Starting PROM Boot process
I-brick attached to module 173503.
HUB at 0x0 attached as widget 0xb
xbridge_xbow_init_arbitrate entry. rtc= 0x1b7c01
  iio_link_up 1
Check_master: link 11 is master
get_local_console : 791 -- hub_base 1800000
Check_master: link 11 is master
HUB at 0x0 attached as widget 0xb
diag_pcibus_sanity: /hw/module/001c01/xtalk/15: FAILED
SGI IP37 SAL version 0.01, built 07:07:09 PM Jan 16, 2001 by root
  built for bedrock rev. 1.1 or greater
Local master CPU A revision: 0
INFO: console subchannel changed:  001c24 CPU1
Local slave CPU B revision: 10000
INFO: console subchannel changed:  001c24 CPU0
flash count: 0
Configured bedrock clock: 200.0 MHz
Status of local IO: 0x1, IIO_LLP_CSR 0x3fc03ff640a
Bedrock Rev: 2, Module: 0 (001c00) from PROM backup
Configuring memory
mdir_init done
WARNING:  RAS to CAS Delay set to 3 cycles per promlog variable
MemCfgRCD3.
Local memory configured: 512 MB (standard)
mdir_config done
   MEMORY_CONFIG after mdir_late_init: 0x302b100000000033
Nasid 0 has no 256MBit DRAMs: slowing refresh
*** Diag level set to None (2)
A Found Synergy 1.0 parts
INFO: console subchannel changed:  001c24 CPU1
B Found Synergy 1.0 parts
INFO: console subchannel changed:  001c24 CPU0
before reading NIC
Hub NIC: 0x4ad92ae7
SR1 set to 0x0000040008041000
SR0 set to 0x000000004ad92ae7
Testing/Initializing memory ..............  DONE
```

```
****** SHADOWING DISABLED *******

       ****** ENTER main_cont() *******
INFO: console subchannel changed:  001c24 CPU1
B - DEBUG FEATURES (B0): 0xf44
INFO: console subchannel changed:  001c24 CPU0
A - DEBUG FEATURES (B0): 0xf44
INFO: console subchannel changed:  001c24 CPU1
B - PROC  FEATURES (B0): 0x0100060000000000, (imp 0xbf00c60000000000,
con 0xbf00c00000000000)
INFO: console subchannel changed:  001c24 CPU0
A - PROC  FEATURES (B0): 0x0100060000000000, (imp 0xbf00c60000000000,
con 0xbf00c00000000000)
INFO: console subchannel changed:  001c24 CPU1
B - BUS   FEATURES (B0): 0x0000000020000000, (imp 0xbdc0000060000000,
con 0xbd80000040000000)
INFO: console subchannel changed:  001c24 CPU0
A - BUS   FEATURES (B0): 0x0000000020000000, (imp 0xbdc0000060000000,
con 0xbd80000040000000)
Initializing kldir.
Done initializing kldir.
Initializing klconfig.
init_klcfg: nasid 0 start 30000 size 10000
Done initializing klconfig.
testing init ip27
doing io discover
Discovering local IO ......................  Check_master: link 11 is
master
Check_master: link 11 is master
DONE
done with io discover
Discovering NUMAlink connectivity .........
Local hub NUMAlink is down (0x0).
discover: diag_mode=2
*** Local network link down
DONE
Found 1 objects (1 hubs, 0 routers) in 10810 usec
Waiting for peers to complete discovery....  Discovery results:
ENTRY 0: HUB(4ad92ae7)
    NASID=-1 Mod=0 Flg=0x500000 PROM=0.1 Route=N/A
    MODULE=001c00 PARTITION=0 SPACE=RESET
    Port 1 connection: Not connected
    Port status: NF
DONE
network discovery done
number of nodes = 1
No other nodes present; becoming global master
Global master is entry 0, NIC 0x4ad92ae7, /hw/node/module/0/
Global master is /hw/node/module/0/
Global barrier (line 3326)Global barrier passed.
Global barrier (line 3348)Global barrier passed.
Master System Topology Graph (pre-nasid_assign):
INFO: console subchannel changed:  001c24 CPU1
Local Slave B : Waiting for my NASID ...
INFO: console subchannel changed:  001c24 CPU0
```

```
ENTRY 0: HUB(4ad92ae7)
    NASID=-1 Mod=0 Flg=0x500000 PROM=0.1 Route=N/A
    MODULE=001c00 PARTITION=0 SPACE=RESET
    Port 1 connection: Not connected
    Port status: NF
Calculating NASIDs
num_routers is 0
Master System Topology Graph:
ENTRY 0: HUB(4ad92ae7)
    NASID=0 Mod=0 Flg=0x500000 PROM=0.1 Route=N/A
    MODULE=001c00 PARTITION=0 SPACE=RESET
    Port 1 connection: Not connected
    Port status: NF
Distributing routing tables
Distributing NASIDs
A Changing node ID to 0
INFO: console subchannel changed:  001c24 CPU1
B Changing node ID to 0
INFO: console subchannel changed:  001c24 CPU0
Global barrier (line 3682)Global barrier passed.
About to set real NASIDs

       ****** ENTER main_cont_two() *******
Real NASIDs have been set
CPU A initialized subnode
Global barrier (line 3870)Global barrier passed.
Nasids in partition:  +0
Regions in partition:  +0
CPU A : Other subnode headless. Initializing....
A init_headless_subnode: nasid 0, subnode 1
DONE
Global barrier (line 3937)Global barrier passed.
Checking partitioning information .........  DONE
No other nodes present; becoming partition master
*** After partitioning ***

ENTRY 0: HUB(4ad92ae7)
    NASID=-1 Mod=0 Flg=0x500000 PROM=0.1 Route=N/A
    MODULE=001c00 PARTITION=0 SPACE=RESET
    Port 1 connection: Not connected
    Port status: NF
Calculating NASIDs
num_routers is 0
Master System Topology Graph:
ENTRY 0: HUB(4ad92ae7)
    NASID=0 Mod=0 Flg=0x500000 PROM=0.1 Route=N/A
    MODULE=001c00 PARTITION=0 SPACE=RESET
    Port 1 connection: Not connected
    Port status: NF
Distributing routing tables
Distributing NASIDs
A Changing node ID to 0
INFO: console subchannel changed:  001c24 CPU1
B Changing node ID to 0
INFO: console subchannel changed:  001c24 CPU0
Global barrier (line 3682)Global barrier passed.
About to set real NASIDs
```

```
       ****** ENTER main_cont_two() *******
Real NASIDs have been set
CPU A initialized subnode
Global barrier (line 3870)Global barrier passed.
Nasids in partition:  +0
Regions in partition:  +0
CPU A : Other subnode headless. Initializing....
A init_headless_subnode: nasid 0, subnode 1
DONE
Global barrier (line 3937)Global barrier passed.
Checking partitioning information .........  DONE
No other nodes present; becoming partition master
*** After partitioning ***
ENTRY 0: HUB(4ad92ae7)
    NASID=0 Mod=0 Flg=0x500000 PROM=0.1 Route=N/A
    MODULE=001c00 PARTITION=0 SPACE=RESET
    Port 1 connection: Not connected
    Port status: FE
Erecting partition fences ................                        DONE
Update config for routers connected to hubs
Update config for hubs and hubless routers
A Master Shadowing PAL to 0x40000, size 0x39f00. New PAL call addr
0x48010
Global barrier (line 4122) INFO: console subchannel changed:  001c24
CPU1
B Slave Shadowing PAL to 0x40000, size 0x39f00. New PAL call addr
0x48010
INFO: console subchannel changed:  001c24 CPU0
INFO: console subchannel changed:  001c24 CPU1
Local slave entering slave loop
INFO: console subchannel changed:  001c24 CPU0
Global barrier passed.
Loading EFI .......................  PROM - in load_exec_ia64_elf,
calling fit_compt_get
PROM - got entry - index is 1, address 0x80000000ffc08000
PROM - got entry - index is 2, address 0xffc95b40
PROM - got entry - index is 3, address 0xff800a00
PROM - got entry - index is 4, address 0xff8fe4d0
PROM - calling load_binary_image; EFI fit addr is 0xff8fe4d0
Loading image (uncompressed) from 0xff8fe4d0 -> 0x200000 (0x297950
bytes)
PROM - calling segldr_load_ia64
entry = 0xfc1e350, loadaddr = 0xfc00000, length = 0x234db0, memlength =
0x3294d8, type = 0x1
entry = 0xfc1e350, loadaddr = 0xff294d8, length = 0x94d8, memlength =
0x94d8, type = 0x1GDA at 0x2400
  cnode 0, nasid 0, lb 0x30070
    mb 0x30be0
      bank 0, paddr 0x0, size 0x10000000
      bank 1, paddr 0x40000000, size 0x10000000
PROM - calling segldr_jump_ia64
INFO: console subchannel changed:  001c24 console
DONE

2 CPUs on 1 nodes found.
nslv_nvram = 0
```

```
NVRAM checksum is incorrect: reinitializing.
nslv_nvram = 0
*************KLMALLOC TABLE****************
index   base    limit   current size
0       0x30070 0x30970 0x303b0 2304
1       0x30970 0x33f70 0x31810 13824
2       0x37570 0x3d510 0x38330 24480
3      0x34000    0x37560    0x34000    13664
NIC INFO START = 0x3f5a8
ROUTER NIC INFO PTR = 0x3f418
Initializing PDAs...
Initializing malloc...
Initializing intrs...
Initializing spb...
Calling cpu_reset...
initlock ....
Initializing libsc_private...
Initializing init_prom_graph ...
Installing PROM Device drivers ............
FIXME: kl_usb_install
FIXME: qlfc_attach: diagnostics disabled
Probing Fibre Adapter 2: 1+ 2+ = 2 device(s) found

Probing ieee1394 Adapter 5
can’t do businfo read to id 0, adr = 0xfffff0000400, status = 0x7
can’t do businfo read to id 0, adr = 0xfffff0000400, status = 0x7
can’t do businfo read to id 0, adr = 0xfffff0000400, status = 0x7
probe to controller  had 1 nodes fail
Initializing init_prom_graph ... DONE
Initializing memdescs...
Initializing load space...
Initializing alarm...
Initializing restart block
Initializing eiobs...
Initializing devices...
Initializing PROM Device drivers .........              DONE
Initializing fs...
fatinit
Initializing mbufs...
Initializing sockets...
Initializing boot env...
Initializing stdio...
Checking hardware inventory ...............
Warning: Inventory table ID value is 0. Check Midplane NIC
Writing 3 records... DONE
Updated new configuration. Wrote 3 records.

Dumpping klconfig Nasid 0:
    IP37, Nasid 0, Module 001c01, widget 0x0, partition 0, NIC
0x4ad92ae7 lboard 0x30070 type  2, flags 0x0001, diagval   0, physid
0, virtid  0: BEDROCK
 type 37, flags 0x0001, diagval   0, physid    0, virtid 4294967295:
SYNERGY
 type 37, flags 0x0001, diagval   0, physid    0, virtid 4294967295:
SYNERGY
 type  3, flags 0x0001, diagval   0, physid  255, virtid 4294967295:
```

```
MEMBANK
 type 17, flags 0x0031, diagval   0, physid    0, virtid  0: Hub-Uart
 type  1, flags 0x0001, diagval   0, physid    0, virtid  0: CPU
 type  1, flags 0x0001, diagval   0, physid    1, virtid  1: CPU
 type  1, flags 0x0000, diagval  32, physid    2, virtid  2: CPU
 type  1, flags 0x0000, diagval  32, physid    3, virtid  3: CPU
    IOBRICK_XBOW, Nasid 0, Module 001c01, widget 0x0, partition 1, NIC
0xffffffffffffffff lboard 0x30140
 type  4, flags 0x0001, diagval   0, physid  255, virtid  0: XBOW
    PBRICK, Nasid 0, Module 001c01, widget 0xe, partition 1, NIC
0x30e3fd lboard 0x30210 type  5, flags 0x0001, diagval   0, physid   14,
virtid 14: BRIDGE
    PBRICK, Nasid 0, Module 001c01, widget 0xf, partition 1, NIC
0x30e3fd lboard 0x302e0 type  5, flags 0x0001, diagval   0, physid   15,
virtid 15: BRIDGE
 type 32, flags 0x0031, diagval   0, physid    3, virtid  2: Qlogic
Fibre Channel
 type  6, flags 0x0031, diagval   0, physid    4, virtid  0: IOC3
 type 34, flags 0x0031, diagval   0, physid    5, virtid 4294967295: USB
(OHCI interface)
 type 33, flags 0x0031, diagval   0, physid    6, virtid  5:
IEEE-1394/Firewire
 type  0, flags 0x0031, diagval   0, physid    7, virtid  0: ID


EIEdpfPEFI emulator ver .1
Build flags:

KLSTRUCT_QLFIBRE: ctlr 2, unit 3
KLSTRUCT_1394:    ctlr 5, unit 6
PlInstallOtherDevices: fixme
RtSetVariable-English: fixme
PlInstallSignalHandlers: fixme
Device mapping table
  fs0  : Fibre(2)/HD(Part1,Sig0)
  blk0 : Fibre(1)
  blk1 : Fibre(2)
  blk2 : Fibre(2)/HD(Part1,Sig0)
  blk3 : Fibre(2)/HD(Part2,Sig0)
  blk4 : Fibre(2)/HD(Part3,Sig0)
  blk5 : Fibre(2)/HD(Part4,Sig0)
  blk6 : Fibre(2)/HD(Part4,Sig0)/HD(Part1,Sig0)
  blk7 : Fibre(2)/HD(Part4,Sig0)/HD(Part2,Sig0)
Shell>
```

14 RPM Installation/Upgrade

1. Telnet to dist.engr.sgi.com,and login as guest.
2. cd to correct directory (see below)
3. ftp to target SN-IA
4. On the SN-IA target machine, run this command in directory where rpm was ftp'ed into:

   rpm -ivh kernel-snia-2.4.2-5SGI_131.ia64.rpm

5. cd /boot/efi
6. cp vmlinux-2.4.2-5SGI_131snia vmlinux.lbs

   Note: vmlinux.lbs is what is in boot.nsh)

7. reboot

From an e-mail:

```
I grabbed the rpms via ftp and put them in /tmp.
>
> I configured up the network interface to do this -- this was a fresh
> root with no valid network config...
>
> [root@duron /etc]# ifconfig eth0 128.162.20.55 netmask 255.255.255.0
> broadcast 1 28.162.20.255 up
> [root@duron /etc]# route add default gw 128.162.20.254
> [root@duron /etc]# cd /tmp
> [root@duron /tmp]# ftp subway
>
> ****
> [ftp'd the kernel-snia-2.4.2-5SGI_131.ia64.rpm and
>  arraysvcs-3.5-2.ia64.rpm to /tmp]
> ****
>
> [root@duron /tmp]# ls -l
> total 3560
> -rw-r--r--   1 root     root           48 Apr 27 14:02 arrayd.out
> -rw-r--r--   1 root     root       654890 Apr 27 14:05
arraysvcs-3.5-2.ia64.rpm
> -rw-r--r--   1 root     root      2975540 Apr 27 14:03
kernel-snia-2.4.2-5SGI_131.ia64.rpm
>
> ****
> [Here I install the two RPMs...]
> ****
>
> [root@duron /tmp]# rpm -Uvh arraysvcs-3.5-2.ia64.rpm
> arraysvcs
##################################################
> kernel unaligned access to 0xe000000000b018e6, ip=0xe000000000642550
> arrayd: no process killed
> [root@duron /tmp]# chkconfig array on
```

```
>
> [root@duron /tmp]# rpm -Uvh kernel-snia-2.4.2-5SGI_131.ia64.rpm
> kernel-snia
##################################################
> depmod: *** Unresolved symbols in
/lib/modules/2.4.2-5SGI_131snia/kernel/drivers/xscsi/sbp2_mod.o
> depmod: ieee1394_wait_for_response_Rsmp_8389a7fb
> depmod: ieee1394_transaction_data_request_Rsmp_5891d489
> depmod: ieee1394_get_dest_id_Rsmp_350464d5
> depmod: ieee1394_register_protocol_Rsmp_d03a6e2e
> depmod: ieee1394_connect_to_node_Rsmp_afb605c8
> depmod: ieee1394_gain_interest_Rsmp_d8471064
> depmod: ieee1394_get_speed_Rsmp_3417a633
> depmod: ieee1394_force_busreset_Rsmp_05ce2107
> removal of /boot/efi/vmlinux-2.4.2-5SGI_118snia failed: No such file or
directory
>
> ****
> [Here I check the current boot.nsh file as it needs to be updated]
> ****
>
> [root@duron /tmp]# cd /boot/efi
> [root@duron efi]# cat boot.nsh
> lilo -t 0 -q vmlinux.lbs root=/dev/sda2 fsb1
>
> ****
> [vi boot.nsh to add the fpswa support]
> [After editing, I catted the new version -- here it is]
> ****
> [root@duron efi]# cat boot.nsh^
> fs0:
> load fpswa.efi
> lilo -t 0 -q vmlinux.lbs root=/dev/sda2 fsb1
>
> ****
> [OK - the kernel rpm doesn’t install the kernel as vmlinux.lbs by
default -
>  so we need to copy the new one over vmlinux.lbs]
> ****
>
> [root@duron efi]# ls -l vmlinux.lbs
> -rwxr-xr-x   1 root     root      8335462 Apr 25  2001 vmlinux.lbs
> [root@duron efi]# ls -l vmlinux-2*
> -rwxr-xr-x   1 root     root      8578682 Apr 26  2001
vmlinux-2.4.2-5SGI_131snia
> [root@duron efi]# cp vmlinux-2.4.2-5SGI_131snia vmlinux.lbs
>
> ****
> [OK - so now the kernel is copied in to place, the startup script calls
the
>  right stuff -- ready to halt -- note I got a general exception here
but
>  that isn’t what we care about right now so I’ll skip it here
>  I send a reset.  The rest of this shows the bootup messages  and EFI
>  interaction]
> ****
>
```

```
>
>
>
>
> 001c01-L1>rst
>
> returning to console mode  001c01 console, <CTRL_T> to escape to L1
> INFO: console subchannel changed:  001c05 CPU0
> ========== Platform errors outstanding A: early main() ==========
> NASID 0x0, cpu 0
> DRAM_UE_ERROR2_00xffffffffffffffff
> DRAM_UE_ERROR2_10xffffffffffffffff
> DRAM_CE_ERROR2_00xffffffffffffffff
> DRAM_CE_ERROR2_10xffffffffffffffff
> PI_INT_PEND1    0x4000000000000000
>
> Warning: kl_error_show_node() - klconfig not properly initialized!
> HARDWARE ERROR STATE:
> End Hardware Error State
> INFO: console subchannel changed:  001c05 CPU2
> ========== Platform errors outstanding C: early main() ==========
> NASID 0x0, cpu 2
> DRAM_UE_ERROR2_00xffffffffffffffff
> DRAM_UE_ERROR2_10xffffffffffffffff
> DRAM_CE_ERROR2_00xffffffffffffffff
> DRAM_CE_ERROR2_10xffffffffffffffff
>
> Warning: kl_error_show_node() - klconfig not properly initialized!
> HARDWARE ERROR STATE:
> End Hardware Error State
> INFO: console subchannel changed:  001c05 CPU0
> NOTICE:  ecc enabled even though CONFIG requested disabled
> Starting PROM Boot process
> SGI IP37 SAL version 0.02, built 04:58:36 PM Mar 13, 2001 by jalal
> WARNING:  RAS to CAS Delay set to 3 cycles per promlog variable
MemCfgRCD3.
> Nasid 0 has no 256MBit DRAMs: slowing refresh
> *** Diag level set to None (2)
> Testing/Initializing memory ...............DONE
> ========== Platform errors outstanding A: (in main.c - where shadowing
is enabled) ==========
> NASID 0x0, cpu 0
> SYN_ERROR       0x0000000400020008
> SYSAD_ERROR1    0x00000001aa1c397d
> PI_ERR_INT_PEND 0x0000000000200000
> PI_ERR_STATUS0_A0xc002008c01600026
> PI_ERR_STATUS1_A0x0010080000000000
>
> Warning: kl_error_show_node() - klconfig not properly initialized!
> HARDWARE ERROR STATE:
> End Hardware Error State
> ========== Platform errors outstanding A: main_cont() ==========
> NASID 0x0, cpu 0
> PI_ERR_INT_PEND 0x0000000000200000
>
> Warning: kl_error_show_node() - klconfig not properly initialized!
> HARDWARE ERROR STATE:
```

```
> End Hardware Error State
> Discovering local IO .....................INFO: console subchannel
changed:  001c01 CPU0
> A 000 001c01: ========== Platform errors outstanding A: early main()
==========
> A 000 001c01: NASID 0x0, cpu 0
> A 000 001c01: DRAM_UE_ERROR2_00xffffffffffffffff
> A 000 001c01: DRAM_UE_ERROR2_10xffffffffffffffff
> A 000 001c01: DRAM_CE_ERROR2_00xffffffffffffffff
> A 000 001c01: DRAM_CE_ERROR2_10xffffffffffffffff
> A 000 001c01: PI_INT_PEND1    0x6000000000000000
> A 000 001c01:
> A 000 001c01: Warning: kl_error_show_node() - klconfig not properly
initialized!
> A 000 001c01: HARDWARE ERROR STATE:
> A 000 001c01: End Hardware Error State
> INFO: console subchannel changed:  001c01 CPU2
> C 000 001c01: ========== Platform errors outstanding C: early main()
==========
> C 000 001c01: NASID 0x0, cpu 2
> C 000 001c01: DRAM_UE_ERROR2_00xffffffffffffffff
> C 000 001c01: DRAM_UE_ERROR2_10xffffffffffffffff
> C 000 001c01: DRAM_CE_ERROR2_00xffffffffffffffff
> C 000 001c01: DRAM_CE_ERROR2_10xffffffffffffffff
> C 000 001c01:
> C 000 001c01: Warning: kl_error_show_node() - klconfig not properly
initialized!
> C 000 001c01: HARDWARE ERROR STATE:
> C 000 001c01: End Hardware Error State
> INFO: console subchannel changed:  001c01 CPU0
> A 000 001c01: NOTICE:  ecc enabled even though CONFIG requested
disabled
> A 000 001c01: Starting PROM Boot process
> A 000 001c01: I/O brick attached to module 001c01.
> A 000 001c01: HUB at 0x0 attached as widget 0xb
> A 000 001c01: xbridge_xbow_init_arbitrate entry. rtc= 0xbd759
> A 000 001c01:   iio_link_up 1
> A 000 001c01: Check_master: link 11 is master
> A 000 001c01: get_local_console : 793 -- hub_base 1800000
> A 000 001c01: Check_master: link 11 is master
> A 000 001c01: HUB at 0x0 attached as widget 0xb
> A 000 001c01: discover_hub_console : 895
> A 000 001c01: get_local_console : 811 -- uart_base 0x800002000f820178
> A 000 001c01: SGI IP37 SAL version 0.02, built 04:58:36 PM Mar 13, 2001
by jalal
> A 000 001c01:   built for bedrock rev. 1.1 or greater
> A 000 001c01: Local master CPU A revision: 0
> INFO: console subchannel changed:  001c01 CPU2
> C 000 001c01: Local slave CPU C revision: 20000
> INFO: console subchannel changed:  001c01 CPU0
> A 000 001c01: flash count: 0
> A 000 001c01: Configured bedrock clock: 200.0 MHz
> A 000 001c01: Status of local IO: 0x1, IIO_LLP_CSR 0x3fc0fff640a
> A 000 001c01: Bedrock Rev: 2, Module: 1 (001c01) from Sys Ctlr
> A 000 001c01: Configuring memory
> A 000 001c01: mdir_init done
> A 000 001c01: WARNING:  RAS to CAS Delay set to 3 cycles per promlog
variable MemCfgRCD3.
> A 000 001c01: Local memory configured: 512 MB (standard)
> A 000 001c01: mdir_config done
> A 000 001c01:    MEMORY_CONFIG after mdir_late_init: 0x302b100000000033
> A 000 001c01: Nasid 0 has no 256MBit DRAMs: slowing refresh
> A 000 001c01: *** Warning: System controller debug switches are
non-zero (0x4)
> A 000 001c01: *** Diag level set to None (2)
> A 000 001c01: *** Info level set to verbose
> A 000 001c01: A Found Synergy 2.0 parts
> INFO: console subchannel changed:  001c01 CPU2
> C 000 001c01: C Found Synergy 2.0 parts
> INFO: console subchannel changed:  001c01 CPU0
> A 000 001c01: before reading NIC
> A 000 001c01: Hub NIC: 0x4e465e67
> A 000 001c01: SR1 set to 0x0000080010041000
> A 000 001c01: SR0 set to 0x000000004e465e67
> A 000 001c01: Testing/Initializing memory ...............DONE
> A 000 001c01:
> A 000 001c01:        ****** ENTER main_cont() *******
> A 000 001c01: A - PROC  FEATURES (B3): 0x0100060000000000, (imp
0xbf00c60000000000, con 0xbf00c00000000000)
> INFO: console subchannel changed:  001c01 CPU2
> C 000 001c01: C - PROC  FEATURES (B3): 0x0100060000000000, (imp
0xbf00c60000000000, con 0xbf00c00000000000)
> INFO: console subchannel changed:  001c01 CPU0
> A 000 001c01: A - BUS   FEATURES (B3): 0x0000000020000000, (imp
0xbdc00000e0000000, con 0xbd80000040000000)
> INFO: console subchannel changed:  001c01 CPU2
> C 000 001c01: C - BUS   FEATURES (B3): 0x0000000020000000, (imp
0xbdc00000e0000000, con 0xbd80000040000000)
> INFO: console subchannel changed:  001c01 CPU0
> A 000 001c01: A - DEBUG FEATURES (B3): 0x2104
> INFO: console subchannel changed:  001c01 CPU2
> C 000 001c01: C - DEBUG FEATURES (B3): 0x2104
> INFO: console subchannel changed:  001c01 CPU0
> A 000 001c01: Initializing kldir.
> A 000 001c01: Done initializing kldir.
> A 000 001c01: Initializing klconfig.
> A 000 001c01: init_klcfg: nasid 0 start 30000 size 10000
> A 000 001c01: Done initializing klconfig.
> A 000 001c01: testing init ip27
> A 000 001c01: doing io discover
> A 000 001c01: Discovering local IO ......................Check_master:
link 11 is master
> A 000 001c01: Check_master: link 11 is master
> A 000 001c01: io_discover: Copying parents module_id 0xfffffffe to IO
Board 0x3fffe28
> A 000 001c01: DONE
> A 000 001c01: done with io discover
> A 000 001c01: Discovering NUMAlink connectivity .........discover:
diag_mode=2
> A 000 001c01: get_chipid: path=0x0 status=0x0000000023012049 r=0
> A 000 001c01: discover_hub: path=0x0 stat=0x0000000023012049 from=0/0
> A 000 001c01: discover_hub: nic=0x4e465e24
> A 000 001c01: discover_hub: sr0=0x000000004e465e24
sr1=0x0000280010060002
> A 000 001c01: discover_hub: running diags
> A 000 001c01: discover_hub: sr1=0x0000280010060002, flags=0x400000,
promvers=0
> A 000 001c01: DONE
> A 000 001c01: Found 2 objects (2 hubs, 0 routers) in 75692 usec
> A 000 001c01: Waiting for peers to complete discovery....Discovery
results:
> A 000 001c01: ENTRY 0: HUB(4e465e67)
> A 000 001c01:     NASID=-1 Mod=1 Flg=0x500000 PROM=0.2 Route=N/A
> A 000 001c01:     MODULE=001c01 PARTITION=0 SPACE=RESET
> A 000 001c01:     Port 1 connection: Entry 1 Hub 4e465e24
> A 000 001c01:     Port status: NF
> A 000 001c01: ENTRY 1: HUB(4e465e24)
> A 000 001c01:     NASID=-1 Mod=5 Flg=0x400000 PROM=0.2 Route=0x0
> A 000 001c01:     MODULE=001c05 PARTITION=0 SPACE=RESET
> A 000 001c01:     Port 1 connection: Entry 0 Hub 4e465e67
> A 000 001c01:     Port status: NF
> A 000 001c01: DONE
> A 000 001c01: network discovery done
> A 000 001c01: number of nodes = 2
> A 000 001c01: Global master NIC is 0x4e465e67
> A 000 001c01: Global master is entry 0, NIC 0x4e465e67,
/hw/node/module/1/
> A 000 001c01: Global master is /hw/node/module/1/
> A 000 001c01: Global barrier (line 3315) \
> A 000 001c01: \
> A 000 001c01: \
> A 000 001c01: Global barrier (line 3337) \
> A 000 001c01: \
> A 000 001c01: \
> A 000 001c01: Master System Topology Graph (pre-nasid_assign):
> INFO: console subchannel changed:  001c01 CPU2
> C 000 001c01: Local Slave C : Waiting for my NASID ...
> INFO: console subchannel changed:  001c01 CPU0
> A 000 001c01: ENTRY 0: HUB(4e465e67)
> A 000 001c01:     NASID=-1 Mod=1 Flg=0x500000 PROM=0.2 Route=N/A
> A 000 001c01:     MODULE=001c01 PARTITION=0 SPACE=RESET
> A 000 001c01:     Port 1 connection: Entry 1 Hub 4e465e24
> A 000 001c01:     Port status: NF
> A 000 001c01: ENTRY 1: HUB(4e465e24)
> A 000 001c01:     NASID=-1 Mod=5 Flg=0x400000 PROM=0.2 Route=0x0
> A 000 001c01:     MODULE=001c05 PARTITION=0 SPACE=RESET
> A 000 001c01:     Port 1 connection: Entry 0 Hub 4e465e67
> A 000 001c01:     Port status: NF
> A 000 001c01: Calculating NASIDs
> A 000 001c01: num_routers is 0
> A 000 001c01: Master System Topology Graph:
> A 000 001c01: ENTRY 0: HUB(4e465e67)
> A 000 001c01:     NASID=0 Mod=1 Flg=0x500000 PROM=0.2 Route=N/A
> A 000 001c01:     MODULE=001c01 PARTITION=0 SPACE=RESET
> A 000 001c01:     Port 1 connection: Entry 1 Hub 4e465e24
> A 000 001c01:     Port status: NF
> A 000 001c01: ENTRY 1: HUB(4e465e24)
> A 000 001c01:     NASID=1 Mod=5 Flg=0x400000 PROM=0.2 Route=0x0
> A 000 001c01:     MODULE=001c05 PARTITION=0 SPACE=RESET
> A 000 001c01:     Port 1 connection: Entry 0 Hub 4e465e67
> A 000 001c01:     Port status: NF
> A 000 001c01: Distributing routing tables
> A 000 001c01: Set up Hub: index 1, nic 0x4e465e24, nasid 1
> A 000 001c01: Distributing NASIDs
> A 000 001c01: > Notifying Hub: index 1, nic 0x4e465e24, nasid 1
> A 000 001c01: A Changing node ID to 0
> INFO: console subchannel changed:  001c01 CPU2
> C 000 001c01: C Changing node ID to 0
> INFO: console subchannel changed:  001c01 CPU0
> A 000 001c01: Global barrier (line 3671) \
> A 000 001c01: \
> A 000 001c01: \
> A 000 001c01: About to set real NASIDs
> A 000 001c01: Global barrier (line 3713) \
> A 000 001c01: \
> A 000 001c01: \
> A 000 001c01:        ****** ENTER main_cont_two() *******
> INFO: console subchannel changed:  001c01 CPU2
> C 000 001c01: CPU C initialized subnode
> INFO: console subchannel changed:  001c01 CPU0
> C 000 001c01: CPU C initialized subnode
>         ****** ENTER main_cont_two() *******
> C 000 001c01: CPU C initialized subnode
> A 000 001c01: CPU A initialized subnode
> A 000 001c01: Global barrier (line 3831) \
> A 000 001c01: \
> A 000 001c01: \
> A 000 001c01: Nasids in partition:  +0 +1
> A 000 001c01: Regions in partition:  +0 +1
> A 000 001c01: Global barrier (line 3898) \
> A 000 001c01: \
> A 000 001c01: \
> Local slave entering slave loop
> INFO: console subchannel changed:  001c01 CPU0
> \
> A 000 001c01: Checking partitioning information .........DONE
> A 000 001c01: Partition master NIC is 0x4e465e67
> A 000 001c01: *** After partitioning ***
> A 000 001c01: ENTRY 0: HUB(4e465e67)
> A 000 001c01:     NASID=0 Mod=1 Flg=0x500000 PROM=0.2 Route=N/A
> A 000 001c01:     MODULE=001c01 PARTITION=0 SPACE=RESET
> A 000 001c01:     Port 1 connection: Entry 1 Hub 4e465e24
> A 000 001c01:     Port status: NF
> A 000 001c01: ENTRY 1: HUB(4e465e24)
> A 000 001c01:     NASID=1 Mod=5 Flg=0x400000 PROM=0.2 Route=0x0
> A 000 001c01:     MODULE=001c05 PARTITION=0 SPACE=RESET
> A 000 001c01:     Port 1 connection: Entry 0 Hub 4e465e67
> A 000 001c01:     Port status: NF
> A 000 001c01: System console is module 001c01.
> A 000 001c01: hub_partition_master : set master console nasid to 0
> A 000 001c01: Erecting partition fences ................
DONE
> A 000 001c01: Update config for routers connected to hubs
> A 000 001c01: Update config for hubs and hubless routers
> A 000 001c01: update_brd_module: Setting module_id brd_module 0x30070
to 001c01
```

```
> A 000 001c01: update_brd_module: Setting module_id brd_module 0x30140
to 001c01
> A 000 001c01: update_brd_module: Setting module_id brd_module 0x30210
to 001c01
> A 000 001c01: update_brd_module: Setting module_id brd_module 0x302e0
to 001c01
> A 000 001c01: update_brd_module: Setting module_id brd_module
0x200030070 to 001c05
> A 000 001c01: update_brd_module: Setting module_id brd_module
0x200030140 to 001c05
> A 000 001c01: update_brd_module: Setting module_id brd_module
0x200030210 to 001c05
> A 000 001c01: update_brd_module: Setting module_id brd_module
0x2000302e0 to 001c05
> A 000 001c01: update_brd_module: Setting module_id brd_module
0x2000303b0 to 001c05
> A 000 001c01: update_brd_module: Setting module_id brd_module
0x200030480 to 001c05
> A 000 001c01: update_brd_module: Setting module_id brd_module
0x200030550 to 001c05
> A 000 001c01: update_brd_module: Setting module_id brd_module
0x200030620 to 001c05
> A 000 001c01: A Master Shadowing PAL to 0x40000, size 0x3a100. New PAL
call addr 0x48010
> A 000 001c01: Global barrier (line 4137) INFO: console subchannel
changed:  001c01 CPU2
> C Slave Shadowing PAL to 0x40000, size 0x3a100. New PAL call addr
0x48010
> INFO: console subchannel changed:  001c01 CPU0
> A 000 001c01: \
> Local slave entering slave loop
> INFO: console subchannel changed:  001c01 CPU0
> A 000 001c01: > Waiting for slave set: index 1, nic 0x4e465e24, nasid 1
> A 000 001c01: \
> A 000 001c01: \
> Region mask for partition 0 is 0x3
> pttn_init_node:  Setting up memory fences on nasid 1, rm=0x3
> pttn_init_node: nasid(1) slot(0) 0x200000000 --> 0x210000000
> INFO: console subchannel changed:  001c01 CPU0
> \
> A 000 001c01: Region mask for partition 0 is 0x3
> A 000 001c01: pttn_init_node:  Setting up memory fences on nasid 0,
rm=0x3
> A 000 001c01: pttn_init_node: nasid(0) slot(0) 0x0 --> 0x10000000
> A 000 001c01: pttn_init_node: nasid(0) slot(1) 0x40000000 -->
0x50000000
> A 000 001c01: Loading EFI ......................INFO: console
subchannel changed:  001c01 console
> DONE
>
> 4 CPUs on 2 nodes found.
> nslv_nvram = 0
> nslv_nvram = 0
> **************KLMALLOC TABLE****************
> index   base    limit   current size
> 0       0x30070 0x30970 0x303b0 2304
> 1       0x30970 0x33f70 0x31810 13824
> 2       0x37570 0x3d510 0x38330 24480
> 3      0x34000    0x37560    0x34000    13664
> NIC INFO START = 0x3f4d8
> ROUTER NIC INFO PTR = 0x3f418
> Initializing PDAs...
> Initializing malloc...
> Initializing intrs...
> Initializing spb...
> Calling cpu_reset...
> initlock ....
> Initializing libsc_private...
> Initializing init_prom_graph ...
> node 0: 4<->55 edge cpu2 add error 1
> init_board_graph:init_ip27_graph error code 0x0,node 1
> Installing PROM Device drivers ...........
> FIXME: kl_usb_install
> FIXME: qlfc_attach: diagnostics disabled
> Mailbox timeout: intr_retry_cnt 200, m0 0x4000, isr 0x8020, bus_sema
0x2, hccr 0x80
> qlfc_reset_interface failed (15)
> qlfcInitIsp failed (4)
> qlfcInit failed (2)
> FIXME: qlfc_attach: diagnostics disabled
> Probing Fibre Adapter 2: 1+ 2+ = 2 device(s) found
>
> Probing ieee1394 Adapter 5
> Detected Vendor id/Product TEAC CD-540E
> Initializing init_prom_graph ... DONE
> Initializing memdescs...
> Initializing load space...
> Initializing alarm...
> Initializing restart block
> Initializing eiobs...
> Initializing devices...
> Initializing PROM Device drivers ..........             DONE
> Initializing fs...
> fatinit
> Initializing mbufs...
> Initializing sockets...
> Initializing boot env...
> Initializing stdio...
> Checking hardware inventory ...............
> Warning: Inventory table ID value is 0. Check Midplane NIC
> Writing 10 records.......... DONE
> Updated new configuration. Wrote 10 records.
>
> Dumpping klconfig Nasid 0:
>     IP37, Nasid 0, Module 001c01, widget 0x0, partition 0, NIC
0x4e465e67 lboard 0x30070
>         type  2, flags 0x0001, diagval   0, physid    0, virtid  0:
BEDROCK
>         type 37, flags 0x0001, diagval   0, physid    0, virtid
4294967295: SYNERGY
>         type 37, flags 0x0001, diagval   0, physid    0, virtid
4294967295: SYNERGY
>         type  3, flags 0x0001, diagval   0, physid  255, virtid
4294967295: MEMBANK
```

```
>     type 17, flags 0x0031, diagval   0, physid    0, virtid  0:
Hub-Uart
>     type  1, flags 0x0001, diagval   0, physid    2, virtid  1: CPU
>     type  1, flags 0x0000, diagval  32, physid    3, virtid  5: CPU
>     type  1, flags 0x0001, diagval   0, physid    0, virtid  0: CPU
>     type  1, flags 0x0000, diagval  32, physid    1, virtid  4: CPU
>   IOBRICK_XBOW, Nasid 0, Module 001c01, widget 0x0, partition 0, NIC
0xffffffffffffffff lboard 0x30140
>     type  4, flags 0x0001, diagval   0, physid  255, virtid  0:
XBOW
>   IBRICK, Nasid 0, Module 001c01, widget 0xe, partition 0, NIC
0xffffffffffffffff lboard 0x30210
>     type  5, flags 0x0001, diagval   0, physid   14, virtid 14:
BRIDGE
>     type 32, flags 0x0031, diagval   0, physid    2, virtid  4:
Qlogic Fibre Channel
>   IBRICK, Nasid 0, Module 001c01, widget 0xf, partition 0, NIC
0xffffffffffffffff lboard 0x302e0
>     type  5, flags 0x0001, diagval   0, physid   15, virtid 15:
BRIDGE
>     type 32, flags 0x0031, diagval   0, physid    3, virtid  2:
Qlogic Fibre Channel
>     type  6, flags 0x0031, diagval   0, physid    4, virtid  0:
IOC3
>     type 34, flags 0x0031, diagval   0, physid    5, virtid
4294967295: USB (OHCI interface)
>     type 33, flags 0x0031, diagval   0, physid    6, virtid  5:
IEEE-1394/Firewire
>
> Dumpping klconfig Nasid 1:
>   IP37, Nasid 1, Module 001c05, widget 0x0, partition 0, NIC
0x4e465e24 lboard 0x200030070
>     type  2, flags 0x0001, diagval   0, physid    0, virtid  1:
BEDROCK
>     type 37, flags 0x0001, diagval   0, physid    0, virtid
4294967295: SYNERGY
>     type 37, flags 0x0001, diagval   0, physid    0, virtid
4294967295: SYNERGY
>     type  3, flags 0x0001, diagval   0, physid  255, virtid
4294967295: MEMBANK
>     type 17, flags 0x0031, diagval   0, physid    1, virtid  1:
Hub-Uart
>     type  1, flags 0x0001, diagval   0, physid    2, virtid  3: CPU
>     type  1, flags 0x0000, diagval  32, physid    3, virtid  7: CPU
>     type  1, flags 0x0001, diagval   0, physid    0, virtid  2: CPU
>     type  1, flags 0x0000, diagval  32, physid    1, virtid  6: CPU
>   IOBRICK_XBOW, Nasid 1, Module 001c05, widget 0x0, partition 0, NIC
0xffffffffffffffff lboard 0x200030140
>     type  4, flags 0x0001, diagval   0, physid  255, virtid  0:
XBOW
>   PBRICK, Nasid 1, Module 001c05, widget 0x8, partition 0, NIC
0xffffffffffffffff lboard 0x200030210
>     type  5, flags 0x0001, diagval   0, physid    8, virtid  8:
BRIDGE
>     type  0, flags 0x0031, diagval   0, physid    1, virtid  0: ID
>   PBRICK, Nasid 1, Module 001c05, widget 0x9, partition 0, NIC
0xffffffffffffffff lboard 0x2000302e0
```

```
>         type  5, flags 0x0001, diagval   0, physid    9, virtid  9:
BRIDGE
>     PBRICK, Nasid 1, Module 001c05, widget 0xc, partition 0, NIC
0xffffffffffffffff lboard 0x2000303b0
>         type  5, flags 0x0001, diagval   0, physid   12, virtid 12:
BRIDGE
>     PBRICK, Nasid 1, Module 001c05, widget 0xd, partition 0, NIC
0xffffffffffffffff lboard 0x200030480
>         type  5, flags 0x0001, diagval   0, physid   13, virtid 13:
BRIDGE
>     PBRICK, Nasid 1, Module 001c05, widget 0xe, partition 0, NIC
0xffffffffffffffff lboard 0x200030550
>         type  5, flags 0x0001, diagval   0, physid   14, virtid 14:
BRIDGE
>     PBRICK, Nasid 1, Module 001c05, widget 0xf, partition 0, NIC
0xffffffffffffffff lboard 0x200030620
>         type  5, flags 0x0001, diagval   0, physid   15, virtid 15:
BRIDGE
>         type 32, flags 0x0031, diagval   0, physid    1, virtid
4294967295: Qlogic Fibre Channel
>
>
>
> dpfEFI emulator ver .1
> Build flags:
> EFI64 EFI_BRINGUP BRINGUP IP37
> EFI Initialization begins ...
> dks5d0s0: blksz = 2048 (expected 512)‘
> PlInstallOtherDevices: fixme
> RtSetVariable-English: fixme
> PlInstallSignalHandlers: fixme
> Device mapping table
>   fs0   : Pci(1|3)/Fibre(1)/HD(Part1,Sig0)
>   fs1   : Pci(1|3)/Fibre(2)/HD(Part1,Sig0)
>   blk0  : Pci(1|3)/Fibre(1)
>   blk1  : Pci(1|3)/Fibre(1)/HD(Part1,Sig0)
>   blk2  : Pci(1|3)/Fibre(1)/HD(Part2,Sig0)
>   blk3  : Pci(1|3)/Fibre(1)/HD(Part3,Sig0)
>   blk4  : Pci(1|3)/Fibre(1)/HD(Part4,Sig0)
>   blk5  : Pci(1|3)/Fibre(1)/HD(Part4,Sig0)/HD(Part1,Sig0)
>   blk6  : Pci(1|3)/Fibre(1)/HD(Part4,Sig0)/HD(Part2,Sig0)
>   blk7  :
Pci(1|3)/Fibre(1)/HD(Part4,Sig0)/HD(Part2,Sig0)/HD(Part1,Sig0)
>   blk8  : Pci(1|3)/Fibre(2)
>   blk9  : Pci(1|3)/Fibre(2)/HD(Part1,Sig0)
>   blkA  : Pci(1|3)/Fibre(2)/HD(Part2,Sig0)
>   blkB  : Pci(1|3)/Fibre(2)/HD(Part3,Sig0)
>   blkC  : Pci(1|3)/Fibre(2)/HD(Part4,Sig0)
>   blkD  : Pci(1|3)/Fibre(2)/HD(Part4,Sig0)/HD(Part1,Sig0)
>   blkE  : Pci(1|3)/Fibre(2)/HD(Part4,Sig0)/HD(Part2,Sig0)
>   blkF  :
Pci(1|3)/Fibre(2)/HD(Part4,Sig0)/HD(Part2,Sig0)/HD(Part1,Sig0)
>   blk10 : Pci(3|1)/1394(0)
> Shell> fs0:
> fs0:\> boot
> +boot> load fpswa.efi
> FPSWA: Image handle 24FFC6508.  Installed Success
> FPSWA: FpswaInterface = 24FFB8010  Fpswa = 24FFB8018
> load: image fs0:%s loaded at 24FF64000. returned Success
> +boot> lilo -t 0 -q vmlinux.lbs root=/dev/sda2 fsb1
> LILO for EFI/IA-64 1.1
> lilo: ignoring trailing 45 characters on command line
> kernel     is  'vmlinux.lbs'
> arguments  are 'root=/dev/sda2 fsb1'
> Booting 'vmlinux.lbs' with args='root=/dev/sda2 fsb1'
> Press any key to interrupt
> Loading kernel |


\
>
>
-
>
> \
> Starting kernel at 0xE00000000061C1E0
> final arguments  are 'BOOT_IMAGE=vmlinux.lbs root=/dev/sda2 fsb1'
> lilo: couldn't find ACPI table, default to no ACPI infos
> INFO: console subchannel changed:  001c01 CPU0
> A 000 001c01: A PROM RTS_TRACE:  (sal_mc_set_params)
> A 000 001c01: A PROM RTS_TRACE:  (sal_mc_set_params)
> A 000 001c01: A PROM RTS_TRACE:  (sal_set_vectors)
> A 000 001c01: A PROM RTS_TRACE:  (sal_set_vectors)
> A 000 001c01: A PROM RTS_TRACE:  (sal_set_vectors)
> A 000 001c01: A PROM RTS_TRACE:  (sal_freq_base)
> A 000 001c01: A PROM RTS_TRACE:  (sal_get_console_nasid)
> INFO: console subchannel changed:  001c01 console
> Master console is set to nasid 0nLinux version 2.4.2-5SGI_131snia
(root@lothar) (gcc version 2.96-ia64-000717 snap 001117) #1 SMP Thu Apr
26 20:31:31 PDT 2001
> Warning: EFI system table major version mismatch: got 0.91, expected
1.00
> EFI v0.91 by SGI: MPS=0x0 ACPI=0x1800b0 SALsystab=0x180000
> NODEMEM_S info ....
> Node         start                end
> 0      0xe000000000400000   0xe00000004fffc000
> Holes -> 0xe00000000fc00000 0xe00000004fffc000 0xffffffffffffffff
0xffffffffffffffff 0xffffffffffffffff 0xffffffffffffffff
0xffffffffffffffff 0xffffffffffffffff
> 1      0xe000000200100000   0xe00000024fffc000
> Holes -> 0xe000000210000000 0xe00000024fffc000 0xffffffffffffffff
0xffffffffffffffff 0xffffffffffffffff 0xffffffffffffffff
0xffffffffffffffff 0xffffffffffffffff
> CNODE INFO ....
> 0 1
> BOOT MEM INFO ....
> Node   Start                LowPfn               BootmemMap
> 0      0x0000000000000000   0x0000000000013fff   0xe000000000f8c000
> 1      0x0000000200000000   0x0000000000093fff   0xe000000200100000
> SAL v2.09: oem=SGI, product=SN1
> sal[0] - entry: pal_proc=0x48010, sal_proc=0xff86fa50
> sal[1] - wakeup type 0, 0x12
> SAL: AP wakeup using external interrupt vector 0x12
> CPU 0: 51 virtual and 44 physical address bits
> ACPI: SGI SN1 1.1
>       CPU 0 (0000:0000): Available.
>       CPU 1 (0001:0000): Disabled.
>       CPU 2 (0002:0000): Available.
>       CPU 3 (0003:0000): Disabled.
>       CPU 4 (0000:0001): Available.
>       CPU 5 (0001:0001): Disabled.
>       CPU 6 (0002:0001): Available.
>       CPU 7 (0003:0001): Disabled.
>       4 CPUs available, 8 CPUs total
> ia64_mca_init : begin
> ia64_mca_init : registered mca rendezvous spinloop and wakeup mech.
> ia64_mca_init : correctable mca vector setup done
> ia64_mca_init : registered os mca handler with SAL
> ia64_mca_init : os init handler at 64cde0
> ia64_mca_init : registered os init handler with SAL
> ia64_mca_init : platform-specific mca handling setup done
> Mca related initialization done
> sn1_setup: setting master_node_bedrock_address to 0xc0000b0030000000
> On node 0 totalpages: 32255
> Allocate mem_map: 0xe000000000f90028 - 0xe000000001a30028, entries
81920, size 0xaa0000
> zone(0): 81919 pages.
> zone(1): 0 pages.
> zone(2): 0 pages.
> hpage=0xe0000000011a7828, hpageend=0xe000000001810028
> On node 1 totalpages: 32676
> Allocate mem_map: 0xe000000200104058 - 0xe000000200ba4058, entries
81920, size 0xaa0000
> zone(0): 81919 pages.
> zone(1): 0 pages.
> zone(2): 0 pages.
> hpage=0xe000000200324058, hpageend=0xe000000200984058
> NODE DATA ....
> Node, Start, Size, MemMap, BitMap, StartP, Mapnr, Size, Id
> 0, 0xe000000000000000, 0x4fffc000, 0xe000000000f90028,
0xe000000000405180, 0x0, 0x1d4b5, 0x13fff, 0
> 1, 0xe000000200000000, 0x4fffc000, 0xe000000200104058,
0xe000000200ba9180, 0x200000000, 0x3c3e25b, 0x13fff, 1
> Kernel command line: BOOT_IMAGE=vmlinux.lbs root=/dev/sda2 fsb1
> fpswa interface at 24ffb8010
> CPU 0: base freq=100.000MHz, ITC ratio=14/2, ITC freq=700.000MHz
> Calibrating delay loop... 696.92 BogoMIPS
> Memory: 1019344k/1012016544k available (5676k code, 19552k reserved,
2133k data, 144k init)
> show_mem: fixme
> kdb version 1.8 by Scott Lurndal, Keith Owens. Copyright SGI, All
Rights Reserved
> perfmon: version 0.2
> perfmon: Interrupt vectored to 238
> perfmon: Counters are 32 bits
> perfmon: Maximum counter value 0xffffffff
> perfmon: 4 PMC/PMD pairs
> perfmon: 18 PMCs, 14 PMDs
> perfmon: Sampling format v1
> Dentry-cache hash table entries: 65536 (order: 6, 1048576 bytes)
> Buffer-cache hash table entries: 65536 (order: 5, 524288 bytes)
> Page-cache hash table entries: 65536 (order: 5, 524288 bytes)
> vmdump: setting dump_execute() as dump_function_ptr ...
> vmdump: bank 0 -- 0..10000000
> vmdump: bank 1 -- 40000000..50000000
> vmdump: bank 2 -- 200000000..210000000
> vmdump: bank 3 -- 240000000..250000000
> Inode-cache hash table entries: 65536 (order: 6, 1048576 bytes)
> VFS: Diskquotas version dquot_6.5.0 initialized
> POSIX conformance testing by UNIFIX
> SMP: starting up secondaries.
> CPU 1: 51 virtual and 44 physical address bits
> INFO: console subchannel changed:  001c01 CPU2
> C 000 001c01: C PROM RTS_TRACE:  (sal_freq_base)
> INFO: console subchannel changed:  001c01 console
> CPU 1: base freq=100.000MHz, ITC ratio=14/2, ITC freq=700.000MHz
> Unexpected irq vector 0x112 on CPU 1!
> Calibrating delay loop... 694.36 BogoMIPS
> CPU 2: 51 virtual and 44 physical address bits
> CPU 2: base freq=100.000MHz, ITC ratio=14/2, ITC freq=700.000MHz
> Unexpected irq vector 0x212 on CPU 2!
> Calibrating delay loop... 694.36 BogoMIPS
> CPU 3: 51 virtual and 44 physical address bitsINFO: console subchannel
changed:  001c05 CPU2
> INFO: console subchannel changed:  001c01 console
>
> CPU 3: base freq=100.000MHz, ITC ratio=14/2, ITC freq=700.000MHz
> Unexpected irq vector 0x312 on CPU 3!
> Calibrating delay loop... 694.36 BogoMIPS
> SMP: Total of 4 processors activated (2778.72 BogoMIPS).
> sn_mp_setup(): Allocating backing store for *Nodepdaindr[ 2]
> WARNING : io_module_init: No serial number found.
> synergy_perf_init(), counting is initially disabled
> WARNING: hwgraph_char_device_add() not supported .. use
hwgraph_register.
> WARNING: hwgraph_char_device_add() not supported .. use
hwgraph_register.
> WARNING: hwgraph_char_device_add() not supported .. use
hwgraph_register.
> WARNING: hwgraph_char_device_add() not supported .. use
hwgraph_register.
> PCI: Probing PCI hardware
> Linux NET4.0 for Linux 2.4
> Based upon Swansea University Computer Society NET3.039
> Initializing RT netlink socket
> PAL Information Facility v0.4
> Starting kswapd v1.8
> pty: 256 Unix98 ptys configured
> block: queued sectors max/low 673280kB/542208kB, 1984 slots per queue
> RAMDISK driver initialized: 16 RAM disks of 6000K size 1024 blocksize
> Uniform Multi-Platform E-IDE driver Revision: 6.31
> ide: Assuming 33MHz system bus speed for PIO modes; override with
idebus=xx
> ide0: ports already in use, skipping probe
> ide1: ports already in use, skipping probe
> ide2: ports already in use, skipping probe
> ide3: ports already in use, skipping probe
> ide4: ports already in use, skipping probe
> ide5: ports already in use, skipping probe
> loop: loaded (max 8 devices)
> SGI L1 Protocol Serial driver version 2.0 (2001-04-19) with no serial
options enabled
> ttyS00 at 0x0000 (irq = 65) is a 16550A
> Using PHY 1, vendor 0x20005c1, model 1, rev 0.
> eth0:  MII transceiver found at MDIO address 1, config 3100 status
786d.
> IOC3 SSRAM has 128 kbyte.
> Ethernet address is 08:00:69:11:be:95.
> SCSI subsystem driver Revision: 1.00
> qlogicfc0 : Loop Reinitialized
> qlogicfc0 : Link is Up
> qlogicfc1 : Loop Reinitialized
> qlogicfc1 : Link is Up
> qlogicfc2 : Loop Reinitialized
> qlogicfc2 : Link is Up
> scsi0 : QLogic ISP2200 SCSI on PCI bus 00 device 18 irq 72 base
0xc00002000f700000
> scsi1 : QLogic ISP2200 SCSI on PCI bus 01 device 10 irq 328 base
0xc00002000e600000
> scsi2 : QLogic ISP2200 SCSI on PCI bus 02 device 08 irq 586 base
0xc00002020f400000
> qlogicfc0 : Port Database
> wwn: 210000e08b01729b  scsi_id: 0  loop_id: 0
> wwn: 21000020371410eb  scsi_id: 1  loop_id: 1
> wwn: 5006094670407b12  scsi_id: 2  loop_id: 2
> qlogicfc0 : completion status 0x15; scsi status 0x800; state flags
0x1c28; status flags 0x8000
>   Vendor: SEAGATE   Model: ST19171FC         Rev: SG11
>   Type:   Direct-Access                      ANSI SCSI revision: 02
> qlogicfc0 : completion status 0x15; scsi status 0x800; state flags
0x1c28; status flags 0x8000
>   Vendor: IBM       Model: DRVL18L  CLAR18   Rev: SG03
>   Type:   Direct-Access                      ANSI SCSI revision: 02
> qlogicfc1 : Port Database
> wwn: 210000e08b0166f0  scsi_id: 0  loop_id: 0
> wwn: 21000020375afb26  scsi_id: 1  loop_id: 52
> wwn: 21000020375afc73  scsi_id: 2  loop_id: 53
> wwn: 21000020375afbe2  scsi_id: 3  loop_id: 54
> wwn: 21000020375afc72  scsi_id: 4  loop_id: 55
> wwn: 21000020375af792  scsi_id: 5  loop_id: 62
> wwn: 21000020374b1975  scsi_id: 6  loop_id: 63
> wwn: 21000020375af546  scsi_id: 7  loop_id: 64
> wwn: 21000020374baf28  scsi_id: 8  loop_id: 65
> qlogicfc1 : completion status 0x15; scsi status 0x800; state flags
0x1c28; status flags 0x8000
>   Vendor: SEAGATE   Model: ST318203FC        Rev: 0004
>   Type:   Direct-Access                      ANSI SCSI revision: 02
> qlogicfc1 : completion status 0x15; scsi status 0x800; state flags
0x1c28; status flags 0x8000
>   Vendor: SEAGATE   Model: ST318203FC        Rev: 0004
>   Type:   Direct-Access                      ANSI SCSI revision: 02
> qlogicfc1 : completion status 0x15; scsi status 0x800; state flags
0x1c28; status flags 0x8000
>   Vendor: SEAGATE   Model: ST318203FC        Rev: 0004
>   Type:   Direct-Access                      ANSI SCSI revision: 02
> qlogicfc1 : completion status 0x15; scsi status 0x800; state flags
0x1c28; status flags 0x8000
>   Vendor: SEAGATE   Model: ST318203FC        Rev: 0004
>   Type:   Direct-Access                      ANSI SCSI revision: 02
> qlogicfc1 : completion status 0x15; scsi status 0x800; state flags
0x1c28; status flags 0x8000
>   Vendor: SEAGATE   Model: ST318203FC        Rev: 0004
>   Type:   Direct-Access                      ANSI SCSI revision: 02
> qlogicfc1 : completion status 0x15; scsi status 0x800; state flags
0x1c28; status flags 0x8000
>   Vendor: SEAGATE   Model: ST318203FC        Rev: 0004
>   Type:   Direct-Access                      ANSI SCSI revision: 02
> qlogicfc1 : completion status 0x15; scsi status 0x800; state flags
0x1c28; status flags 0x8000
>   Vendor: SEAGATE   Model: ST318203FC        Rev: 0004
>   Type:   Direct-Access                      ANSI SCSI revision: 02
> qlogicfc1 : completion status 0x15; scsi status 0x800; state flags
0x1c28; status flags 0x8000
>   Vendor: SEAGATE   Model: ST318203FC        Rev: 0004
>   Type:   Direct-Access                      ANSI SCSI revision: 02
> qlogicfc2 : Port Database
> wwn: 210000e08b01d5c2  scsi_id: 0  loop_id: 2
> wwn: 21000020375af6e9  scsi_id: 1  loop_id: 0
> wwn: 21000020375afed5  scsi_id: 2  loop_id: 1
> wwn: 210000203765d05f  scsi_id: 3  loop_id: 3
> wwn: 21000020375aff9e  scsi_id: 4  loop_id: 4
> wwn: 21000020375aff2c  scsi_id: 5  loop_id: 72
> wwn: 2100002037187363  scsi_id: 6  loop_id: 73
> wwn: 21000020371873c2  scsi_id: 7  loop_id: 74
> wwn: 2100002037187607  scsi_id: 8  loop_id: 75
> qlogicfc2 : completion status 0x15; scsi status 0x800; state flags
0x1c28; status flags 0x8000
>   Vendor: SEAGATE   Model: ST318203FC        Rev: 0004
>   Type:   Direct-Access                      ANSI SCSI revision: 02
> qlogicfc2 : completion status 0x15; scsi status 0x800; state flags
0x1c28; status flags 0x8000
>   Vendor: SEAGATE   Model: ST318203FC        Rev: 0004
>   Type:   Direct-Access                      ANSI SCSI revision: 02
> qlogicfc2 : completion status 0x15; scsi status 0x800; state flags
0x1c28; status flags 0x8000
>   Vendor: SEAGATE   Model: ST318304FC        Rev: FA4A
>   Type:   Direct-Access                      ANSI SCSI revision: 03
> qlogicfc2 : completion status 0x15; scsi status 0x800; state flags
0x1c28; status flags 0x8000
>   Vendor: SEAGATE   Model: ST318203FC        Rev: 0004
>   Type:   Direct-Access                      ANSI SCSI revision: 02
> qlogicfc2 : completion status 0x15; scsi status 0x800; state flags
0x1c28; status flags 0x8000
>   Vendor: SEAGATE   Model: ST318203FC        Rev: 0004
>   Type:   Direct-Access                      ANSI SCSI revision: 02
> qlogicfc2 : completion status 0x15; scsi status 0x800; state flags
0x1c28; status flags 0x8000
>   Vendor: SEAGATE   Model: ST318203FC        Rev: 0004
>   Type:   Direct-Access                      ANSI SCSI revision: 02
> qlogicfc2 : completion status 0x15; scsi status 0x800; state flags
0x1c28; status flags 0x8000
>   Vendor: SEAGATE   Model: ST318203FC        Rev: 0004
>   Type:   Direct-Access                      ANSI SCSI revision: 02
> qlogicfc2 : completion status 0x15; scsi status 0x800; state flags
0x1c28; status flags 0x8000
>   Vendor: SEAGATE   Model: ST318203FC        Rev: 0004
>   Type:   Direct-Access                      ANSI SCSI revision: 02
> Attached scsi disk sda at scsi0, channel 0, id 1, lun 0
> Attached scsi disk sdb at scsi0, channel 0, id 2, lun 0
> Attached scsi disk sdc at scsi1, channel 0, id 1, lun 0
> Attached scsi disk sdd at scsi1, channel 0, id 2, lun 0
> Attached scsi disk sde at scsi1, channel 0, id 3, lun 0
> Attached scsi disk sdf at scsi1, channel 0, id 4, lun 0
> Attached scsi disk sdg at scsi1, channel 0, id 5, lun 0
> Attached scsi disk sdh at scsi1, channel 0, id 6, lun 0
> Attached scsi disk sdi at scsi1, channel 0, id 7, lun 0
> Attached scsi disk sdj at scsi1, channel 0, id 8, lun 0
> Attached scsi disk sdk at scsi2, channel 0, id 1, lun 0
> Attached scsi disk sdl at scsi2, channel 0, id 2, lun 0
> Attached scsi disk sdm at scsi2, channel 0, id 3, lun 0
> Attached scsi disk sdn at scsi2, channel 0, id 4, lun 0
> Attached scsi disk sdo at scsi2, channel 0, id 5, lun 0
> Attached scsi disk sdp at scsi2, channel 0, id 6, lun 0
> Attached scsi disk sdq at scsi2, channel 0, id 7, lun 0
> Attached scsi disk sdr at scsi2, channel 0, id 8, lun 0
> SCSI device sda: 17783112 512-byte hdwr sectors (9105 MB)
> Partition check:
>  /dev/scsi/host0/bus0/target1/lun0: p1 p2 p3 p4 < p5 p6 >
> SCSI device sdb: 35546876 512-byte hdwr sectors (18200 MB)
>  /dev/scsi/host0/bus0/target2/lun0: p1 p2 p3 p4 < p5 p6 >
> SCSI device sdc: 35566480 512-byte hdwr sectors (18210 MB)
>  /dev/scsi/host1/bus0/target1/lun0: p1
> SCSI device sdd: 35566480 512-byte hdwr sectors (18210 MB)
>  /dev/scsi/host1/bus0/target2/lun0: p1 p2 p3 p4 < p5 p6 p7 >
> SCSI device sde: 35566480 512-byte hdwr sectors (18210 MB)
>  /dev/scsi/host1/bus0/target3/lun0: p1
> SCSI device sdf: 35566480 512-byte hdwr sectors (18210 MB)
>  /dev/scsi/host1/bus0/target4/lun0: p1
> SCSI device sdg: 35566480 512-byte hdwr sectors (18210 MB)
>  /dev/scsi/host1/bus0/target5/lun0: p1
> SCSI device sdh: 35566480 512-byte hdwr sectors (18210 MB)
>  /dev/scsi/host1/bus0/target6/lun0: p1
> SCSI device sdi: 35566480 512-byte hdwr sectors (18210 MB)
>  /dev/scsi/host1/bus0/target7/lun0: p1
> SCSI device sdj: 35566480 512-byte hdwr sectors (18210 MB)
>  /dev/scsi/host1/bus0/target8/lun0: p1
> SCSI device sdk: 35566480 512-byte hdwr sectors (18210 MB)
>  /dev/scsi/host2/bus0/target1/lun0: p1
> SCSI device sdl: 35566480 512-byte hdwr sectors (18210 MB)
>  /dev/scsi/host2/bus0/target2/lun0: p1
> SCSI device sdm: 35964301 512-byte hdwr sectors (18414 MB)
>  /dev/scsi/host2/bus0/target3/lun0: p1
> SCSI device sdn: 35566480 512-byte hdwr sectors (18210 MB)
>  /dev/scsi/host2/bus0/target4/lun0: p1
> SCSI device sdo: 35566480 512-byte hdwr sectors (18210 MB)
>  /dev/scsi/host2/bus0/target5/lun0: p1
> SCSI device sdp: 35566480 512-byte hdwr sectors (18210 MB)
>  /dev/scsi/host2/bus0/target6/lun0: p1
> SCSI device sdq: 35566480 512-byte hdwr sectors (18210 MB)
>  /dev/scsi/host2/bus0/target7/lun0: p1
> SCSI device sdr: 35566480 512-byte hdwr sectors (18210 MB)
>  /dev/scsi/host2/bus0/target8/lun0: p1
> NET4: Linux TCP/IP 1.0 for NET4.0
> IP Protocols: ICMP, UDP, TCP, IGMP
> IP: routing cache hash table of 16384 buckets, 256Kbytes
> TCP: Hash tables configured (established 131072 bind 65536)
> NET4: Unix domain sockets 1.0/SMP for Linux NET4.0.
> devfs: v0.102 (20000622) Richard Gooch (rgooch@atnf.csiro.au)
> devfs: devfs_debug: 0x0
> devfs: boot_options: 0x0
> VFS: Mounted root (ext2 filesystem) readonly.
> Mounted devfs on /dev
> Freeing unused kernel memory: 144kB freed
> INIT: version 2.78 booting
> Started device management daemon for /dev
> hostname: mankato
> Checking root filesystems.
> Parallelizing fsck version 1.19 (13-Jul-2000)
> [/sbin/fsck.ext2 -- /] fsck.ext2 -a -C0 /dev/sda2
> /dev/sda2 was not cleanly unmounted, check forced.
> /dev/sda2: |==                                                        |
3.5%   /dev/sda2: |====
/  7.0%   /dev/sda2: |======
- 10.5%   /dev/sda2: |=======


=
>
>
>                                                 \ 14.0%   /dev/sda2:
|==========                                               | 17.5%
/dev/sda2: |============                                          /
21.0%   /dev/sda2: |==============



>
>
>              - 24.5%  /dev/sda2:
Deleted inode 113943 has zero dtime.  FIXED.
> /dev/sda2: |================                                        \
28.0%   /dev/sda2: |==================
| 31.5%  /dev/sda2:
Inode 1


46
>
>
> 515, i_blocks is 16, should be 8.  FIXED.
> /dev/sda2: |====================                                   /
35.0%   /dev/sda2: |======================
- 38.5%   /dev/sda2: |========================
\ 42.0%   /dev/sda2: |=======
==
>
>
> ================                               | 45.5%   /dev/sda2:
|===========================                             / 49.0%
/dev/sda2:
Inode 225730, i_blocks is 96
,
>
>
> should be 40.  FIXED.
> /dev/sda2: Inode 225737, i_blocks is 264, should be 256.  FIXED.
> [really long line of nulls deleted]
> Checking root filesystem quotas
> Turning on user and group quotas for root filesystem
> modprobe: Can’t locate module /dev/hda
> modprobe: Can’t locate module /dev/hdc
> Remounting root filesystem in read-write mode.
> Finding module dependencies... depmod: *** Unresolved symbols in
/lib/modules/2.4.2-5SGI_131snia/kernel/arch/ia64/sn/io/pciba.o
> depmod: *** Unresolved symbols in
/lib/modules/2.4.2-5SGI_131snia/kernel/drivers/net/3c59x.o
> depmod: *** Unresolved symbols in
/lib/modules/2.4.2-5SGI_131snia/kernel/drivers/net/acenic.o
> depmod: *** Unresolved symbols in
/lib/modules/2.4.2-5SGI_131snia/kernel/drivers/net/eepro100.o
> depmod: *** Unresolved symbols in
/lib/modules/2.4.2-5SGI_131snia/kernel/drivers/scsi/qla1280.o
> depmod: *** Unresolved symbols in
/lib/modules/2.4.2-5SGI_131snia/kernel/drivers/scsi/qla2x00.o
> depmod: *** Unresolved symbols in
/lib/modules/2.4.2-5SGI_131snia/kernel/drivers/xscsi/sbp2_mod.o
> done
> Checking filesystems.
> Parallelizing fsck version 1.19 (13-Jul-2000)
> Checking all file systems.
> [/sbin/fsck.ext2 -- /extra] fsck.ext2 -a -C0 /dev/sda5
> /dev/sda5 was not cleanly unmounted, check forced.
> /dev/sda5: |=====                                                   |
8.8%   /dev/sda5: |==========
/ 17.5%   /dev/sda5: |================
- 26.2%   /dev/sda5: |=========
>
>
> ===========                                    \ 35.0%   /dev/sda5:
|=========================                               | 43.8%
/dev/sda5: |=============================                           /
52.5%   /dev/sda5: |==============================
>
>
> ====                      - 61.2%   /dev/sda5:
|========================================                \ 70.0%
/dev/sda5: |=========================================               |
73.0%   /dev/sda5: |==========================================

>
>
```

```
>   / 75.7%  /dev/sda5: |============================================
- 79.2%  /dev/sda5: |==============================================
\ 81.6%  /dev/sda5: |===============================================
| 84.6%   /dev


/s
>
>
> da5: |=================================================       / 87.0%
/dev/sda5: |==================================================      -
88.8%  /dev/sda5: |==================================================
\ 90.0%   /dev/sda5: |=============


==
>
>
> =====================================    | 92.0%   /dev/sda5:
|=====================================================   / 95.0%
/dev/sda5: |=====================================================   -
95.3%   /dev/sda5: |==================================


==
>
>
> ==================  \ 96.2%   /dev/sda5:
|========================================================|100.0%
/dev/sda5: 1537/128768 files (0.0% non-contiguous), 9790/257032 b


lo
>
>
> cks
> [/sbin/fsck.ext2 -- /scratch] fsck.ext2 -a -C0 /dev/sda6
> /dev/sda6 was not cleanly unmounted, check forced.
> Checking filesystem quotas
> /etc/rc.d/rc.sysinit: line 199:    34 Segmentation fault
/sbin/quotacheck -v -R -a
> Mounting local filesystems.
> modprobe: Can’t locate module /dev/hda
> modprobe: Can’t locate module /dev/hdc
> modprobe: modprobe: Can’t locate module nls_cp437
> modprobe: modprobe: Can’t locate module nls_iso8859-1
> modprobe: modprobe: Can’t locate module nls_iso8859-1
> modprobe: modprobe: Can’t locate module nls_iso8859-1
> Configuring system to save crash dumps
> modprobe: Can’t locate module /dev/vmdump
> modprobe: Can’t locate module /dev/vmdump
> modprobe: Can’t locate module /dev/vmdump
> modprobe: Can’t locate module /dev/vmdump
> vmdump: dump device opened: 0x803
> Saving crash dump data (if any)
> Error: could not save crash dump.
> Setting up swapspace version 1, size = 1052819456 bytes
> Adding Swap: 1028128k swap-space (priority -1)
> modprobe: Can’t locate module /dev/tty0
> Couldnt get a file descriptor referring to the console
> Turning on user and group quotas for local filesystems
> Setting clock: modprobe: Can’t locate module /dev/rtc
> modprobe: Can’t locate module /dev/tty1
> Cannot access the Hardware Clock via any known method.
> Use the --debug option to see the details of our search for an access
method.
> Sun Apr 27 14:13:16 PDT 2003
> Initializing random number generator...
> INIT: Entering runlevel: 3
> Starting portmapper: kernel unaligned access to 0xe000000000b1892a,
ip=0xe000000000642710
> kernel unaligned access to 0xe000000000b188ea, ip=0xe000000000642710
> kernel unaligned access to 0xe000000000b1ac6a, ip=0xe000000000642710
> kernel unaligned access to 0xe000000000b1898a, ip=0xe000000000642710
> portmap
> Mounting remote filesystems.
> Starting CSA job accounting
> CSA job accounting is not available.
>    System Error(38): Function not implemented.
> Starting system loggers: syslogd klogd
> Setting system clock from remote host zeus.americas.sgi.com
> 27 Apr 14:14:39 ntpdate[215]: step time server 128.162.8.103 offset
-18446744069477656239.231129 sec
> Starting cron daemon: crond
> Starting INET services: inetd
> Starting sshd: sshd
> Using /lib/modules/2.4.2-5SGI_131snia/kernel/drivers/misc/array.o
> Starting Array services:
> Starting gpm mouse services: (no mouse is configured)
> Performance Co-Pilot starting PMCD (logfile is
/var/log/pcp/pmcd/pmcd.log) ...
> Performance Co-Pilot starting archive loggers ...
> Installing support for jobs
> paggctl(PAGG_JOB, JOB_SETHID, ...): Function not implemented
> kernel unaligned access to 0xe000000000b18f5a, ip=0xe000000000642710
>
> TurboLinux release 010307 English Frontier (Beta 3)
> Kernel 2.4.2-5SGI_131snia on an ia64 (
> )
> VC: l
>
> login:
> telnet> quit
> Connection closed.
>
> erikj@subway
> exit
>
> script done on Fri Apr 27 16:15:01 2001
```

15 XSCSI

Before running the new xscsi kernel, the /etc/fstab should have similar changes to the following made:

```
--- /etc/fstab.scsi     Thu Apr 19 01:50:53 2001
+++ /etc/fstab  Thu Apr 19 01:51:04 2001
@@ -1,11 +1,11 @@
-/dev/sda2  /             ext2         defaults        1 1
+/dev/xscsi/pci00.01.0/target1/lun0/part2  /             ext2
defaults        1 1
 proc       /proc         proc         defaults        0 0
 none       /var/shm      shm          defaults        0 0
 #none       /dev/pts      devpts       gid=5,mode=620  0 0
 #none       /proc/bus/usb usbdevfs     defaults        0 0
 /dev/hda   /mnt/floppy   vfat         user,noauto     0 0
 /dev/hdc   /mnt/cdrom    iso9660      ro,user,noauto  0 0
-/dev/sda1  /boot/efi     vfat         defaults        0 0
-/dev/sda3  swap          swap         defaults        0 0
-/dev/sda5  /var          ext2         defaults        0 2
-/dev/sda6  /home         ext2         defaults        0 2
+/dev/xscsi/pci00.01.0/target1/lun0/part1  /boot/efi     vfat
defaults        0 0
+/dev/xscsi/pci00.01.0/target1/lun0/part3  swap          swap
defaults        0 0
+/dev/xscsi/pci00.01.0/target1/lun0/part5  /var          ext2
defaults        0 2
+/dev/xscsi/pci00.01.0/target1/lun0/part6  /home         ext2
defaults        0 2
```

Note: Getting this wrong leaves you with an unbootable system, so make sure it is correct, especially the pci00.01.0 part. The first 2-digit string (00) is the PCI bus #, the second (01) is the slot number of the QL2200A card.

Also, you'll need a new boot script:

```
[root@sniac-edl /root]# cat /boot/efi/bootqlfc.nsh
fs0:
load fpswa.efi
lilo -t 0 -q vmqlfc root=/dev/xscsi/pci00.01.0/target1/lun0/part2
```

Note: If you decide to go back to the qlogicfc driver (current standard kernels), then you have to put the /etc/fstab back.

16 EFI Shell Commands

16.1 Command Status

The following list contains the status of the EFI Shell> commands as of May 8, 2001:

```
fs0:\> help
help [internal command]        - works
guid [sname]                   - works
set [-d] [sname] [value]          - works per-boot, working on permanence
alias [-d] [sname] [value]     - works
dh [-p prot_id] | [handle]     - works
map [-dvr] [sname[:]] [handle] - works
cd [path]                      - works
echo [[-on | -off] | [text]     - not sure this works even on Intel boxes
if [not] condition then        - untested
endif                          - untested
goto label                     - untested
for var in <set>               - untested
endfor                         - untested
pause                          - untested
pod                                      - currently trying to fix this
ls [dir] [dir] ...             - works
mkdir dir [dir] ...                    - works, with fix from my workarea
mode [row col]                 - get works, set untested
cp [-r] file [file] ... [dest]         - works, with fix from my workarea
comp file1 file2               - works
rm file/dir [file/dir]                 - works, with fix from my workarea
memmap                         - works
type [-a] file                 - partly works
dmpstore                       - works
load driver_name                         - works for fpswa and kernel
ver                            - works
err [level]                    - get works, set untested
time [hh:mm:ss]                 - works in EFI, but kernel doesn’t see it
date [mm/dd/yyyy]               - works in EFI, but kernel doesn’t see it
stall microseconds             - "works"  (not implemented)
reset [/warm] [reset string]               - currently trying to fix this
```

```
vol fs [Volume Label]          - get works, set untested

cls [background color]          - untested

edit [file name]                      - last time I tried it, EFI crashed

dblk device [Lba] [Blocks]     - works

mm Address [Width] [;Type]     - untested

mem [Address] [size] [;MMIO]    - works -- do not use 0x before addresses

pci [bus dev] [func]      - runs, but with many errors and no useful info

bcfg -?                        - untested

attrib [-b] [+/- rhs] [file]   - untested
```

There's a kernel / EFI / SAL plumbing problem right now, and I expect that fixing it will get pod, reset, time, and date all working again.

16.2 Information from EFI Shell> Command Test -- April 12, 2001

Note: Contains outdated information, but some of the output may be useful.

```
* Shell> guid
*   DevIo           : AF6AC311-84C3-11D2-8E3C-00A0C969723B
*   diskio          : CE345171-BA0B-11D2-8E4F-00A0C969723B
*   blkio           : 964E5B21-6459-11D2-8E39-00A0C969723B  *
*   txtin           : 387477C1-69C7-11D2-8E39-00A0C969723B
*   txtout          : 387477C2-69C7-11D2-8E39-00A0C969723B  *
*   fs              : 964E5B22-6459-11D2-8E39-00A0C969723B
*   load            : 56EC3091-954C-11D2-8E3F-00A0C969723B
*   image           : 5B1B31A1-9562-11D2-8E3F-00A0C969723B  *
*   varstore        : F088CD91-A046-11D2-8E42-00A0C969723B
*   unicode         : 1D85CD7F-F43D-11D2-9A0C-0090273FC14D
*   LegacyBoot      : 376E5EB2-30E4-11D3-BAE5-0080C73C8881
*   serialio        : BB25CF6F-F1D4-11D2-9A0C-0090273FC1FD
*   pxebc           : 03C4E603-AC28-11D3-9A2D-0090273FC14D
*   net             : A19832B9-AC25-11D3-9A2D-0090273FC14D
*   VgaClass        : 0E3D6310-6FE4-11D3-BB81-0080C73C8881
*   TxtOutSplit     : 56D830A0-7E7A-11D3-BBA0-00A0C969723B
*   ErrOutSplit     : F0BA9039-68F1-425E-AA7F-D9AAF91B82A1
*   TxtInSplit      : F9A3C550-7FB5-11D3-BBA0-00A0C969723B
*   dpath           : 09576E91-6D3F-11D2-8E39-00A0C969723B  *
*   ShellInt        : 47C7B223-C42A-11D2-8E57-00A0C969723B
*   SEnv            : 47C7B224-C42A-11D2-8E57-00A0C969723B
*   ShellProtId     : 47C7B226-C42A-11D2-8E57-00A0C969723B
*   ShellDevPathMap : 47C7B225-C42A-11D2-8E57-00A0C969723B
*   ShellAlias      : 47C7B227-C42A-11D2-8E57-00A0C969723B
*   G0              : 00000000-0000-0000-0000-000000000000
*   Efi             : 8BE4DF61-93CA-11D2-AA0D-00E098032B8C
*   GenFileInfo     : 09576E92-6D3F-11D2-8E39-00A0C969723B
*   FileSysInfo     : 09576E93-6D3F-11D2-8E39-00A0C969723B
*   PcAnsi          : E0C14753-F9BE-11D2-9A0C-0090273FC14D
*   Vt100           : DFA66065-B419-11D3-9A2D-0090273FC14D
*   ESP             : C12A7328-F81F-11D2-BA4B-00A0C93EC93B
*   GPT MBR         : 024DEE41-33E7-11D3-9D69-0008C781F39F
```

```
 *   Unknown Device  : CF31FAC5-C24E-11D2-85F3-00A0C93EC93B
 * Shell> echo zzz aaa 999 laskjdlkslkaj
 * zzz aaa 999 laskjdlkslkaj
 * Shell> memmap
 *
 * Type       Start            End                 # Pages
Attributes
 * RT_data    0000000000000000-00000000003FFFFF  0000000000000400
8000000000000008
 * available  0000000000400000-000000000FBFFFFF  000000000000F800
0000000000000008
 * RT_data    000000000FC00000-000000000FFFFFFF  0000000000000400
8000000000000008
 * available  0000000040000000-000000004FF81FFF  000000000000FF82
0000000000000008
 * BS_data    000000004FF82000-000000004FF9BFFF  000000000000001A
0000000000000008
 * RT_data    000000004FF9C000-000000004FFA1FFF  0000000000000006
8000000000000008
 * BS_data    000000004FFA2000-000000004FFA2FFF  0000000000000001
0000000000000008
 * RT_data    000000004FFA3000-000000004FFB3FFF  0000000000000011
8000000000000008
 * BS_data    000000004FFB4000-000000004FFFBFFF  0000000000000048
0000000000000008
 * RT_data    000000004FFFC000-000000004FFFCFFF  0000000000000001
8000000000000008
 * BS_data    000000004FFFD000-000000004FFFEFFF  0000000000000002
0000000000000008
 * RT_data    000000004FFFF000-000000004FFFFFFF  0000000000000001
8000000000000008
 *
 *   BS_data   :     101 Pages (413,696)
 *   RT_data   :   2,073 Pages (8,491,008)
 *   available : 128,898 Pages (527,966,208)
 * Total Memory: 512 MB (536,870,912) Bytes
 * Shell> dmpstore
 *
 * Dump NVRAM
 * Variable RT+BS ’Efi:LangCodes’ DataSize = 3
 *   00000000: 65 6E 67                                         *eng*
 * Shell> ver
 * EFI Specification Revision      1.2
 *   EFI Vendor        = SGI
 *   EFI Revision      = 000C0021
 *
 * SAL Specification Revision      2. 9
 *   SAL_A Revision    = 0. 1
 *   SAL_B Revision    = 0. 1
 *
 * PAL_A Revision        66.18
 * PAL_B Revision        66.18
 *
 * Other modules mentioned in FIT (Firmware Interface Table)
 * FIT_Entry Type   0, Revision    2.60
 * FIT_Entry Type  16, Revision    0.70
 * FIT_Entry Type  17, Revision    0.70
 * FIT_Entry Type  31, Revision    0. 1
 *
 *  SalProc Entry 00000000FFE3ECC0 and GP 00000000FFEB5BE0
 *  PalProc Entry 0000000000048010 IO Port Base 000000004FF91F08
 *  Cache Enabled
 * Shell> err
 *
 * EFI ERROR 0000000080000000
 *     00000001  D_INIT
 *     00000002  D_WARN
 *     00000004  D_LOAD
 *     00000008  D_FS
 *     00000010  D_POOL
 *     00000020  D_PAGE
 *     00000040  D_INFO
 *     00000100  D_VAR
 *     00000200  D_PARSE
 *     00000400  D_BM
 *     00001000  D_BLKIO
 *     00002000  D_BLKIO_ULTRA
 *     00004000  D_NET
 *     00008000  D_NET_ULTRA
 *     80000000  D_ERROR
 * Shell> time
 * 05:27:48
 * Shell> date
 * 04/13/2001
 * Shell> mem 0x4ff8c0c0 10
 *   Memory Address 0000000000000000 10 Bytes
 *   00000000: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
*................*
 *
 * Shell> dh -p diskio
 * Handle dump by protocol 'diskio'
 *   B: diskio blkio DevPath(Fibre(1))
 *  30: diskio blkio fs DevPath(Fibre(1)/HD(Part1,Sig00000000))
 *  31: diskio blkio DevPath(Fibre(1)/HD(Part2,Sig00000000))
 *  32: diskio blkio DevPath(Fibre(1)/HD(Part3,Sig00000000))
 *  33: diskio blkio DevPath(Fibre(1)/HD(Part4,Sig00000000))
 *  34: diskio blkio DevPath(..00000)/HD(Part1,Sig00000000))
 *  35: diskio blkio DevPath(..00000)/HD(Part2,Sig00000000))
 *  36: diskio blkio DevPath(..00000)/HD(Part1,Sig00000000))
 *   C: diskio blkio DevPath(1394(0))
* Shell> dh -p blkio
 * Handle dump by protocol 'blkio'
 *   B: diskio blkio DevPath(Fibre(1))
 *  30: diskio blkio fs DevPath(Fibre(1)/HD(Part1,Sig00000000))
 *  31: diskio blkio DevPath(Fibre(1)/HD(Part2,Sig00000000))
 *  32: diskio blkio DevPath(Fibre(1)/HD(Part3,Sig00000000))
 *  33: diskio blkio DevPath(Fibre(1)/HD(Part4,Sig00000000))
 *  34: diskio blkio DevPath(..00000)/HD(Part1,Sig00000000))
 *  35: diskio blkio DevPath(..00000)/HD(Part2,Sig00000000))
 *  36: diskio blkio DevPath(..00000)/HD(Part1,Sig00000000))
 *   C: diskio blkio DevPath(1394(0))
 * Shell> dh -p txtin
 * Handle dump by protocol 'txtin'
 *   5: txtin DevPath(I2O(0))
 * Shell> dh -p txtout
 * Handle dump by protocol 'txtout'
 *    6: txtout DevPath(I2O(1))
 * Shell> dh -p fs
 * Handle dump by protocol 'fs'
 *   30: diskio blkio fs DevPath(Fibre(1)/HD(Part1,Sig00000000))
 * Shell> dh -p load
 * Handle dump by protocol 'load'
 *   13: load DevPath(..4000000992B5A390046000000A9))
 * Shell> dh -p image
 * Handle dump by protocol 'image'
 *    7: Image(PcAnsi)
 *    8: Image(vt100)
 *    9: Image(ConSpliter)
 *    A: Image(BiosDisk)
 *    D: Image(dskio)
 *    E: Image(pblkio)
 *   10: Image(fat)
 *   11: Image(SimpleNetwork)
 *   12: Image(PxeBaseCode)
 *   14: Image(shellenv)
 *   15: Image(ls)
 *   16: Image(mkdir)
 *   17: Image(mode)
 *   18: Image(cp)
 *   19: Image(comp)
 *   1A: Image(rm)
 *   1B: Image(memmap)
 *   1C: Image(type)
 *   1D: Image(dmpstore)
 *   1E: Image(load)
 *   1F: Image(ver)
 *   20: Image(err)
 *   21: Image(time)
 *   22: Image(date)
 *   23: Image(stall)
 *   24: Image(reset)
 *   25: Image(vol)
 *   26: Image(cls)
 *   27: Image(edit)
 *   28: Image(lilo)
 *   29: Image(dblk)
 *   2A: Image(mm)
 *   2B: Image(dmem)
 *   2C: Image(pci)
 *   2D: Image(bcfg)
 *   2E: Image(attrib)
 *   2F: Image(nshell) ShellInt
 * Shell> dh -p varstore
 * Handle dump by protocol 'varstore'
 *    1: varstore
 *    3: varstore
 * Shell> dh -p unicode
 * Handle dump by protocol 'unicode'
 *    4: unicode
 * Shell> dh -p LegacyBoot
 * Handle dump by protocol 'LegacyBoot'
 * Shell> dh -p dpath
 * Handle dump by protocol 'dpath'
 *   2: DevIo DevPath()
 *   5: txtin DevPath(I2O(0))
 *   6: txtout DevPath(I2O(1))
 *   B: diskio blkio DevPath(Fibre(1))
 *   C: diskio blkio DevPath(1394(0))
 *  13: load DevPath(..4000000992B5A390046000000A9))
 *  30: diskio blkio fs DevPath(Fibre(1)/HD(Part1,Sig00000000))
 *  31: diskio blkio DevPath(Fibre(1)/HD(Part2,Sig00000000))
 *  32: diskio blkio DevPath(Fibre(1)/HD(Part3,Sig00000000))
 *  33: diskio blkio DevPath(Fibre(1)/HD(Part4,Sig00000000))
 *  34: diskio blkio DevPath(..00000)/HD(Part1,Sig00000000))
 *  35: diskio blkio DevPath(..00000)/HD(Part2,Sig00000000))
 *  36: diskio blkio DevPath(..00000)/HD(Part1,Sig00000000))
 * Shell> dh -p ShellInt
 * Handle dump by protocol 'ShellInt'
 *  2F: Image(nshell) ShellInt
 * Exit status code: Invalid Parameter
 * Shell> dh -p DevIo
 * Handle dump by protocol 'DevIo'
 *   2: DevIo DevPath()
 * Shell> vol fs
 * fs is not a mapped short name
 * Exit status code: Invalid Parameter
 * Shell> vol
 * (null) is not a mapped short name
 * Exit status code: Invalid Parameter
 * Shell> ls fs0:
 * ls: could not list file fs0:. - No mapping
 * Shell> dh -p fs
 * Handle dump by protocol 'fs'
 *  30: diskio blkio fs DevPath(Fibre(1)/HD(Part1,Sig00000000))
 * Shell>
```

17 LEDs

Information on the SN-IA L1>leds command output.

(If the PROM hangs, one can drop into the L1 console by typing Ctrl-T.)

Note: This information is subject to change.

```
/***********************************************************************
**
* *
* File: prom_leds.h *
* *
* Contains the led values displayed at various phases in the *
* PROM startup sequence, and on boot failures. *
* *
* Copyright (C) 2000 Silicon Graphics, Inc. *
* *
* These coded instructions, statements, and computer programs contain *
* unpublished proprietary information of Silicon Graphics, Inc., and *
* are protected by Federal copyright law. They may not be disclosed *
* to third parties or copied or duplicated in any form, in whole or *
* in part, without the prior written consent of Silicon Graphics, Inc. *
* *
************************************************************************
*/
#ident "$Revision: 1.6 $"
#ifndef _IP35_PROM_LEDS_H_
#define _IP35_PROM_LEDS_H_
/*
* Progress LED (PLED) values
*
* Warning: if you change these, you will want to also update the
* document IP35prom/doc/ip35prom.html. Avoid big changes if possible!
* Also be sure to preserve the special values of PLED_NOT_USED_55
* and PLED_BIST_BASE.
*/
#define PLED_RESET 0x00
#define PLED_INMAIN 0x09
#define PLED_SPEEDUP 0x0a
#define PLED_SPEEDUPOK 0x0b
#define PLED_UARTINIT 0x0e
#define PLED_UARTINITDONE 0x0f
#define PLED_PODLOOP 0x10
#define PLED_PODPROMPT 0x11
#define PLED_PODMODE 0x12
#define PLED_LOCALARB 0x13
#define PLED_BARRIER 0x14
#define PLED_MAKESTACK 0x15
#define PLED_BARRIEROK 0x1a
#define PLED_STACKRAMOK 0x20
#define PLED_MDIRINIT 0x25
#define PLED_MDIRCONFIG 0x26
#define PLED_IODISCOVER 0x27
#define PLED_HUB_CONFIG 0x28
#define PLED_ROUTER_CONFIG 0x29
#define PLED_INITIO 0x2a
```

```
#define PLED_CONSOLE_GET 0x2b
#define PLED_CONSOLE_GET_OK 0x2c
#define PLED_INITIODONE 0x2d
#define PLED_STASH2 0x2e
#define PLED_STASH3 0x2f
#define PLED_STASH4 0x30
#define PLED_IODISCOVER_DONE 0x31
#define PLED_TEST_INTS 0x33
/* BRINGUP - need to reorder the LED values */
#define PLED_FLUSH_L4_TO_MEMORY 0x40
#define PLED_PRE_SET_NASIDS 0x41
#define PLED_POST_SET_NASIDS 0x42
#define PLED_NOT_USED_55 0x55 /* Avoid using this pattern */
#define PLED_BIST_BASE 0x60 /* OR-in the dimm,bank */
#define PLED_SAL_ENTRY 0x70
#define PLED_SAL_ENTRY_RESET 0x71
#define PLED_SAL_RESET_OK 0x72
#define PLED_SAL_RESET_WAIT_MASTER 0x73
#define PLED_SAL_RESET_CMAIN 0x74
#define PLED_SAL_INIT 0x75
#define PLED_SAL_CHECK 0x76
#define PLED_SAL_RESET_EARLY_INIT 0x77
#define PLED_APOS_HANDOFF 0x7f
/*
* Failure LED (FLED) values
*
* Warning: if you change these, you will want to also update the
* document IP35prom/doc/ip35prom.html. Avoid big changes if possible!
* Also be sure to preserve the special value of PLED_NOT_USED_AA
*
* Note: avoid using 0x80 - this is frequently the power-on led value
*/
#define FLED_SYN_ERRORS 0x81
#define FLED_SYN_TAGTESTFAILED 0x82
#define FLED_SYN_L4DATAERROR 0x83
#define FLED_SYN_BISTERROR 0x84
#define FLED_SYN_L4ERROR 0x85
#define FLED_SYNCONT_TAGTESTFAILED 0x86
#define FLED_SYNCONT_L4ERROR 0x87
#define FLED_BADREV_SYNERGY 0x88 /* Unsupported Synergy Rev */
#define FLED_HUBLOCAL 0x89 /* Hub local failed */
#define FLED_PREM_DIR_REQ 0x8a /* Some node not premium */
#define FLED_NOMEM 0x8b /* No local memory */
#define FLED_DISABLED 0x8c /* CPU is disabled by envvar */
#define FLED_HUB_CONFIG 0x8f /* Hub klconfig failed */
#define FLED_ROUTER_CONFIG 0x90 /* Router klconfig failed */
#define FLED_HUBIO_INIT 0x91 /* hub io init failed */
#define FLED_CONFIG_INIT 0x92 /* node klconfig failed */
#define FLED_LLP_FAIL 0x94 /* LLP failed after reset */
#define FLED_LLP_NORESET 0x95 /* LLP never up after reset */
#define FLED_NET_DISCOVER 0x96 /* Network discovery failed */
#define FLED_NASID_CALC 0x97 /* NASID calculation failed */
#define FLED_ROUTE_CALC 0x98 /* Route calculation failed */
#define FLED_ROUTE_DIST 0x99 /* Route distribution failed */
#define FLED_NASID_DIST 0x9a /* NASID distribution failed */
#define FLED_NO_NASID 0x9b /* Master assigned no NASID */
#define FLED_NOT_USED_AA 0xaa /* Avoid using this pattern */
```

```
#define FLED_SAL_ENTRY_CODE 0xc0
#define FLED_SAL_RESET_CODE 0xc1
#define FLED_PAL_EARLY_PTP_FAIL 0xc2
#define FLED_PAL_LATE_PTP_FAIL 0xc3
#define FLED_SAL_RESET_CMAIN 0xc4
#define FLED_SAL_RESET_PANIC 0xc5
#define FLED_SAL_RESET_STUB0 0xc8
#define FLED_SAL_RESET_STUB1 0xc9
#define FLED_FLASHING_IN_PROGRESS 0xca /* flashes on remote node */
#define FLED_FLASHING_COMPLETE 0xcb /* flashes on remote node is done */
#define FLED_PROM_VERSION_MISMATCH 0xcf /* PROM VERSION MISMATCH DETECTED
, HALTING THE NODE WITH THE BAD VERSION */
#define PLED_PAL_TEST_PROC_ENTRY 0xe0
#endif /* _IP35_PROM_LEDS_H_ */
```

18 SMP Detection

Q: How can I detect an SMP system in user space even when running a UP kernel?
(Things like 'cat /proc/cpuinfo' don't count.)

A: `usr/bin/getconf _NPROCESSORS_ONLN`

19 Synergy L4 Cache Errors Output

```
> SNIA system
> test cell prosys2-4 crashed running OS high volts.
> Attached is the error info output from console.
>
>
> --------------61363EC0761E609554F4A564
>
>
> re-entering system console mode (002c10 console), <CTRL_T> to escape to
L2^M
> SYNERR: time 3889615, sapicid 0101, val 0x400^M^M
> ^M^M
> MCA - NASID 1, cpu 1^M^M
>  Bus_Check corrected continuable isolated moreInfo^M^M
> MCA Processor State: 0x20000000fff611a0^M^M
> MinState area at 0x8000000200008600^M^M
>   Control registers^M^M
>     IIP 0x400000000001cdf0    IPSR 0x0000101309026038    IFS:
0x800000000000030d^M^M
>     XIP 0x400000000001cdd0    XPSR 0x0000121309026038    XFS:
0x000000000000030d^M^M
>     B0  0x400000000001ac10    PRED 0x0000000000028211    RSC:
0x000000000000000f^M^M
>     ISR 0x0000080400000000    IIPA 0x400000000001cde0   ITIR:
0x0000000000000670^M^M
>     IFA 0xc0000e0000000340     NaT:0x0000000000000000^M^M
>     BSR:0x00030000001e8e00    LIBC 0x00000000ffcf7858    ELSC
0x0000000200005600^M^M
>     PAL:0x0000000000048010^M^M
>   General registers GR0 .. GR31 (bank 1)^M^M
>    GR0   0x0000000000000000  0x6000000000008fe8  0x0000000000000003
0x0000000000000003^M^M
>    GR4   0x0000000000000000  0x80000000fff94640  0x80000b00100000c8
0x00000000000000c0^M^M
>    GR8   0x0000000000004624  0x80000ffffd3ff830  0x00000000000044f1
0x80000ffffd3ff838^M^M
>    GR12  0x80000ffffd3ff810  0x80000ffffd3ffa60  0x6000000000018da8
0x6000000000018da0^M^M
>    GR16  0x0000000000002010  0x0000000000000090  0x400000000001dd60
0x0000000000000003^M^M
>    GR20  0x0000000000002010  0x0000000000000090  0x80000ffffd3ff600
0x0000000000004624^M^M
>    GR24  0x0000000000002010  0x0000000000000090  0x0000000000000090
0x0000000000000090^M^M
>    GR28  0x80000ffffd3ff608  0xe65757bcda0d9640  0x20000000e916f128
0xe65757bcda0d9640^M^M
>   General registers  GR16 .. GR31 (bank 0)^M^M
>    GR16  0xe000000800701288  0xc00000000000030d  0x0000000001480000
0x0000000000000003^M^M
>    GR20  0x0000080400000000  0x20000000003de1a0  0xe000000800701140
0x80000ffffd200000^M^M
>    GR24  0x0000000000000000  0x0000000000000000  0xc00000000000030d
0x000000000000000f^M^M
```

```
>     GR28   0x400000000001cdd0   0x0000121309026038   0x800000000000030d
0x0000000000028211^M^M
>   Error map      : 0x0000000001002000^M^M
>     edc01 - ss   : 0x080000b70000bd16^M^M
>     edc01 - ta   : 0x000000020f42edc0^M^M
>     ebh00 - ss   : 0x1800000000000369^M^M
> ========== Platform errors outstanding B: MCA ==========^M^M
> NASID 0x1, cpu 1^M^M
>         SYN_ERROR               0x00000004000a0200^M^M
```

bit 9 is Uncorrectable ECC error detected on L4 data Superbank 1.

```
>         DRAM_UE_ERROR1_1        0x8000020000392dd0^M^M
```

UCE Failing Address is 0x392dd0, Way 0x2.

```
>         DRAM_UE_ERROR2_1        0xaaaa88aaaaaaaaaa^M^M
```
This is the doubleword associated with the error.

```
>         DRAM_CE_ERROR1_1        0xc0041600001178e0^M^M
```

There was also a correctable error at address 0x001178e0, way 0x6.

```
>         DRAM_CE_ERROR2_1        0x1087010000000040^M^M
```

This is the doubleword associated with the error.

```
> ^M^M
> HARDWARE ERROR STATE:^M^M
> End Hardware Error State^M^M
> Branching to OS_MCA Handler^M^M
> SYNERR: time 14396772, sapicid 0101, val 0x4000a0200^M^M
> ^M^M
>
> MCA - NASID 1, cpu 1^M^M
>  Bus_Check Cache_Check MulErr isolated uncontained moreInfo^M^M
> MCA Processor State: 0x28000000fff21330^M^M
> MinState area at 0x8000000200008600^M^M
>   Control registers^M^M
>     IIP 0x400000000001cf30    IPSR 0x00001013090a6018    IFS:
0x800000000000030d^M^M
>     XIP 0x400000000001cf30    XPSR 0x00001013090a6018    XFS:
0x000000000000030d^M^M
>     B0  0x400000000001ac10    PRED 0x0000000000020111    RSC:
0x000000000000000f^M^M
>     ISR 0x0000080400000000    IIPA 0x400000000001cf20   ITIR:
0x0000000000000670^M^M
>     IFA 0xc0000e0000000340     NaT:0x0000000000000000^M^M
>     BSR:0x00030000001e8e00    LIBC 0x00000000ffcf7858    ELSC
0x0000000200005600^M^M
>     PAL:0x0000000000048010^M^M
>   General registers GR0 .. GR31 (bank 1)^M^M
>    GR0   0x0000000000000000   0x6000000000008fe8   0x0000000000004624
0x00000000000044ea^M^M
>    GR4   0x0000000000000000   0x80000000fff94640   0x80000b00100000c8
0x00000000000000c0^M^M
```

```
>    GR8   0x80000ffffd3ff838  0x0000000000004624  0x0000000000004624
0x80000ffffd3ff838^M^M
>    GR12  0x80000ffffd3ff810  0x80000ffffd3ffa60  0x19a8a84325f269bf
0x20000000cf294df0^M^M
>    GR16  0x0000000000004624  0x80000ffffd3ff838  0x8643be9e85812f4f
0x00000000000044ea^M^M
>    GR20  0x80000ffffd3ff850  0x80000ffffffff250  0x80000ffffd3ff838
0x0000000000002010^M^M
>    GR24  0x0000000000000090  0x0000000000000090  0x0000000000000090
0x0000000000000090^M^M
>    GR28  0x80000ffffd3ff608  0xe65757bcda0d9640  0x20000000e916f128
0xe65757bcda0d9640^M^M
>   General registers  GR16 .. GR31 (bank 0)^M^M
>    GR16  0xe000000800701288  0xc00000000000030d  0x0000000001480000
0x0000000000000003^M^M
>    GR20  0x0000080400000000  0x20000000003de1a0  0xe000000800701140
0x80000ffffd200000^M^M
>    GR24  0x0000000000000000  0x0000000000000000  0xc00000000000030d
0x000000000000000f^M^M
>    GR28  0x400000000001cf30  0x00001013090a6018  0x800000000000030d
0x0000000000020111^M^M
>   Error map      : 0x0000000001002000^M^M
>     edc01 - ss   : 0x080000b700009d16^M^M
>     ebh00 - ss   : 0x1800000000000369^M^M
> ========== Platform errors outstanding B: MCA ==========^M^M
> NASID 0x1, cpu 1^M^M
>        SYN_ERROR              0x0000000400020008^M^M
>        SYSAD_ERROR1           0x00000001fa1c2d6a^M^M
>        DRAM_UE_ERROR1_1       0xc000020000392dd0^M^M
>        DRAM_UE_ERROR2_1       0xaaaa88aaaaaaaaaa^M^M
>        DRAM_CE_ERROR1_1       0xc0041600001178e0^M^M
>        DRAM_CE_ERROR2_1       0x1087010000000040^M^M
> ^M     PI_ERR_INT_PEND        0x000000001b000000^M^M
>        PI_ERR_STATUS0_A       0xc0904e4b7d8a0006^M^M
>        PI_ERR_STATUS1_A       0x0120000000000000^M^M
> ^M     PI_ERR_STATUS0_B       0xc0904e4b60800006^M^M
>        PI_ERR_STATUS1_B       0x0000040000000000^M^M
>        MD_MEM_ERROR           0x3300ff0041392df0^M^M
```

The Bedrock MD detected bad ECC, but the syndrom is 0xFF, indicating that the data was bad before being written back to memory.

```
> ^M^M^M
> ^M
> HARDWARE ERROR STATE:^M^M
> ia64_log_get: retrieved 0 bytes of error informationEnd Hardware Error
State^M^M
> ^MBranching to OS_MCA Handler^M^M
> ^M
> B PROM ERROR: Unimplemented SAL call (sal_get_state_info)^M^M
> +BEGIN HARDWARE ERROR STATE [MCA PROCESSOR]B PROM ERROR: Unimplemented
SAL call (sal_get_state_info)^M^M
> ^M^M
> + Cache check info[0]^M^M
> +  Level: L0 ,Index: 0, ,Operation: Unknown,^M^M
> + Cache check info[1]^M^M
> +  Level: L0 ,Index: 0, ,Operation: Unknown,^M^M
> + Cache check info[2]^M^M
> +  Level: L0 ,Index: 0, ,Operation: Unknown,^M^M
> + Cache check info[3]^M^M
> +  Level: L0 ,Index: 0, ,Operation: Unknown,^M^M
> + Cache check info[4]^M^M
> +  Level: L0 ,Index: 0, ,Operation: Unknown,^M^M
> + Cache check info[5]^M^M
> +  Level: L0 ,Index: 0, ,Operation: Unknown,^M^M
> + TLB Check Info [0]^M^M
> +^M^M
> + TLB Check Info [1]^M^M
> +^M^M
> + TLB Check Info [2]^M^M
> +^M^M
> + TLB Check Info [3]^M^M
> +^M^M
> + TLB Check Info [4]^M^M
> +^M^M
> + TLB Check Info [5]^M^M
> +^M^M
> B PROM ERROR: Unimplemented SAL call (sal_clear_state_info)^M^M
> + BUS Check Info [0]B PROM ERROR: Unimplemented SAL call
(sal_clear_state_info)^M^M
> ^M^M
> + Status Info: 0 ,Severity: 0 ,Transaction Type: 0 ,Transaction Size:
0^M^M
> +END HARDWARE ERROR STATE [MCA PROCESSOR]^M^M
> ^M^M
> ia64_mca_log_get : Clearing processor log failed^M^M
> Kernel panic: Uncorrected MCA abort^M^M
> SYNERR: time 14398468, sapicid 0101, val 0x400020008^M^M
> ^M^M
> Entering kdb (current=0xe000000800700000, pid 2672) on processor 2 due
to panic^M^M
> [2]kdb> ^M
>
>
> --------------61363EC0761E609554F4A564--
```

20 SN-Itanium Preliminary Spares (05/08/01)

Note: Preliminary / Under Construction. SPO or GPS has not yet reviewed this Document for finalization.

The following information includes only those spares that are unique to SN-Itanium C bricks.

SN-Itanium Processor Spares

Part Number	Description	Qty Per Brick	MTBF Per Spare	Est Unit Cost	FE or FR	Depot Spare	Cty Spare
013-3423-001	Itanium Processor Assy 733MHz 2MB	2-4	909,091		FR	x	x
013-3424-001	Itanium Processor Assy 800MHz 2MB	4	909,091		FR	x	x
013-3425-001	Itanium Processor Assy 800MHz 4MB	4	909,091		FR	x	x

SN-Itanium Unique Maintenance Tools

Part Number	Description	Qty Per Kit	MTBF Per Spare	Est Unit Cost	FE or FR	Depot Spare	Cty Spare
042-0979-001	Tool IP37 PCA Handle	1	N/A		FE		
	Pin Checker	1	N/A		FE		
	Pin Straightner	1	N/A		FE		

Standard SN-Itanium C-Brick Spares

Part Number	Description	Qty Per Brick	MTBF Per Spare	Est Unit Cost	FE or FR	Depot Spare	Cty Spare
013-3407-001	Logic Carrier Assy S/M Itanium	1			FE		
015-0316-001	Harness Assy Power Pod	1-2	N/A	6	FE		
030-1437-002	PCA Subassy IP37 4CPU Itanium-1	1	140,418	4738	FR	x	x
060-0066-001	Power Supply 48V Power Pod	2-4	1,000,000	376	FR		

21 Links

- SNIA SW Platform Group's Home Page
  http://ddt1.engr.sgi.com/
- SN-Itanium Project Page (Daryl Coulthart)
  http://wwwmn.americas.sgi.com/snia/
- SN-Itanium Bringup Home Page
  http://gout.engr.sgi.com/sn1-ia/
- Livelink SN-Itanium Page
  https://sginfo.corp.sgi.com/livelink/livelink.exe/
- DevInfo SN-Itanium Page
  http://cujo.americas.sgi.com/devinfo/Menus/P04/index.html
- SN1-Itanium Diagnostic Software Packages
  http://wwwcf.americas.sgi.com/PUBLIC/diags/release/sn1ia/index.html
- SGI Intel Web Site
  http://intelweb.esd.sgi.com/default.htm
- devfs and hwgraph
  http://www-devtoolbox.engr.sgi.com/linux/documents/sgilinux/devfs.html
- Linux: simple example of using devfs
  http://www-devtoolbox.engr.sgi.com/linux/src/exampleCode/drivers/devfs/
- IRIX-Linux Differences
  http://www-devtoolbox.engr.sgi.com/linux/documents/sgilinux/ildifs.html
- Driver Kernel Interface (DKI) In IRIX and Linux
  http://www-devtoolbox.engr.sgi.com/linux/documents/sgilinux/dki.html
- The PCI System in Linux 2.2.5
  http://www-devtoolbox.engr.sgi.com/linux/documents/sgilinux/PCIforX85.html
- SGI Linux Base Software Build Procedures
  http://wwwccn.americas.sgi.com/~ljgm/linux/linuxbuilds/
- SGI Linux Base Software (LBS) 1.5
  http://info.engr.sgi.com/propack/lbs15/
- Service Information Homepage
  http://servinfo.csd.sgi.com