

Custom Debian installer with Areca support

Download the official netinst ISO image and mount it:

```
mkdir /sarge; mount debian-31r0a-i386-netinst.iso /sarge -oloop
```

Now create custom kernels. The installer sports a 2.4.27 and a 2.6.8 kernel, so I had to make 2 kernels. The following procedure is for 2.6 but needs to be repeated for the 2.4 one.

Install the kernel source package:

```
apt-get install kernel-source-2.6.8
cd /usr/src; tar xvfj kernel-source-2.6.8.tar.bz2
```

Unpack the kernel-package from the sarge tree and copy the configuration to your source tree:

```
dpkg -x /sarge/pool/main/k/kernel-image-2.6.8-i386/kernel-image-2.6.8-2-386_2.6.8-16_i386.deb /tmp
cp /tmp/boot/config-2.6.8-2-386 /usr/src/kernel-source-2.6.8/.config
```

Now customize, configure, build and install your kernels (for the Areca module it was necessary to symlink the kerneltree to /usr/src/linux otherwise it would not build):

```
cd /usr/src/
ln -s kernel-source-2.6.8 linux
cd linux
[add modules/patches]
make menuconfig
fakeroot make-kpkg --initrd --append-to-version="-2-386" --
revision="2.6.8.Areca.1.20.0X.08" kernel_image
dpkg -i kernel-image-2.6.8-2-386_2.6.8.Areca.1.20.0X.08_i386.deb
```

NOTE

During my attempt to build an amd64 kernel, it turned out that the kernel on the stock ISO was compiled using gcc-3.4. Since by default that installer install gcc-3.3, the module would not load because of a wrong 'version magic'. You should check this by using the "modinfo" command. I had to install gcc-3.4 and g++-3.4, and change the symlinks in /usr/bin.

Create a working copy of the installation cdrom:

```
mkdir /new-sarge
rsync -av -H --delete --exclude=TRANS.TBL /sarge /new-sarge
```

Get debian-installer sources for 2.4 and 2.6 kernels:

```
cd; mkdir installer; cd installer;
apt-get source linux-kernel-di-i386 linux-kernel-di-i386-2.6
```

For each kernel version, add the module to the appropriate list and create kernel udebs:

```
cd linux-kernel-di-i386-2.6-1.02
echo "drivers/scsi/arcmsr/arcmsr.o" >> modules/i386/scsi-extra-modules
fakeroot debian/rules binary
```

This will create a lot of so-called "udeb's". A udeb is a small Debian package containing a set of kernel modules. The udeb that contains our driver is in this case scsi-extra-modules-2.6.8-2-386-di_1.02_i386.udeb. We now replace the stock udeb with this new one:

```
cp installer/scsi-extra-modules-2.6.8-2-386-di_1.02_i386.udeb /new-
sarge/pool/main/l/linux-kernel-di-i386-2.6/
```

Now copy your new kernel debs (the big one, not the split stuff) also to the pool and remove the stock kernel:

```
cp /usr/src/kernel-image-2.6.8-2-386_2.6.8.Areca.1.20.0X.08_i386.deb /new-
sarge/pool/main/k/kernel-image-2.6.8-386/
rm /new-sarge/pool/main/k/kernel-image-2.6.8-i386/ kernel-image-2.6.8-2-386_2.6.8-
16_i386.deb
```

For the installer to recognize the controller without manually modprobing, the initrd image for both kernels needs to be altered. First the module needs to be available, and also the PCI database that 'discover' uses may need to be edited to make it recognize the new controller.

The initrd-image is a compressed gzip file and can be mounted for editing:

```
cd /new-sarge/install/2.6
gunzip initrd.gz
mount initr /initrd -oloop
cd /initrd
cp -Rp /lib/modules/2.6.8-2-386/kernel/drivers/scsi/arcmsr/ lib/modules/2.6.8-2-
386/kernel/drivers/scsi/
depmod -a -b .
cd usr/share/discover
```

Now edit the "pci.lst" file and add your new hardware to the list.

You have to know the PCI ID's of the board you want to support.

In my case this meant adding these:

```
17d3 Areca Technology Corp.
  17d31110      scsi   arcmsr  ARC-1110 4-Port PCI-X to SATA RAID Controller
  17d31120      scsi   arcmsr  RC-1120 8-Port PCI-X to SATA RAID Controller
  17d31130      scsi   arcmsr  ARC-1130 12-Port PCI-X to SATA RAID Controller
  17d31160      scsi   arcmsr  ARC-1160 16-Port PCI-X to SATA RAID Controller
  17d31170      scsi   arcmsr  ARC-1170 24-Port PCI-X to SATA RAID Controller
  17d31210      scsi   arcmsr  ARC-1210 4-Port PCI-Express to SATA RAID
Controller
  17d31220      scsi   arcmsr  ARC-1220 8-Port PCI-Express to SATA RAID
Controller
  17d31230      scsi   arcmsr  ARC-1230 12-Port PCI-Express to SATA RAID
Controller
  17d31260      scsi   arcmsr  ARC-1260 16-Port PCI-Express to SATA RAID
Controller
```

Please keep in mind that discover is very picky and allows ONLY TABS as delimiters!! If you make them spaces, the discover scripts skip the entries and your hardware is NOT recognized. You have been warned ;)

```
cd -
```

```
umount initrd
cat initrd | gzip -9 > initrd.gz
```

Reboot into your 2.4.27 kernel and repeat these steps for 2.4.27.

After all this is done, we need to fix the md5sums, names, and sizes of the deb's we modified. In my case, these files were:

```
pool/main/l/linux-kernel-di-amd64-2.6/scsi-extra-modules-2.6.8-11-amd64-generic-
di_0.22_amd64.udeb
pool/main/k/kernel-image-2.6.8-amd64/kernel-image-2.6.8-11-amd64-
generic_2.6.8.Areca.1.20.00.08_amd64.deb
```

(You could find these by issuing "find -type f -mtime 0" in your new sarge install tree.

This info is put into multiple files:

```
dists/sarge/main/debian-installer/binary-i386/Packages      (contains description of the
scsi-extra-modules package)
dists/sarge/main/debian-installer/binary-i386/Packages.gz   (contains description of the
scsi-extra-modules package)
dists/sarge/main/binary-i386/Packages                      (contains description of the
kernel package)
dists/sarge/main/binary-i386/Packages.gz                  (contains description of the
kernel package)
```

Edit the plain text version of these files and make sure the proper values are in place.
After that creat a compressed copy in place:

```
gzip -9 -c Packages > Packages.gz
```

Keep in mind that the installer will not work if the checksums are not correct!

Once the packages files are updated, you need to calculate checksums of each and every
file on the installer. Fortunately this can be done very easy:

```
cd /new-sarge
md5sum `find -follow -type f` > md5sum.txt
```

Now create the ISO image:

```
cd /new-sarge
mkisofs -o /root/new-sarge.iso -r -J -no-emul-boot -boot-load-size 4 -boot-info-table -b
isolinux/isolinux.bin -c isolinux/boot.cat .
```