Adaptec dpti driver

This driver supports the Adaptec I2O RAID and DPT SmartRAID V I2O boards.

Credits

The original linux driver was ported to Linux by Karen White while at Dell Computer. It was ported from Bob Pasteur's (of DPT) original non-Linux driver. Mark Salyzyn and Bob Pasteur consulted on the original driver.

2.0 version of the driver by Deanna Bonds and Mark Salyzyn.

History

The driver was originally ported to linux version 2.0.34

V2.0	Rewrite of driver. Re-architectured based on i2o subsystem. This was the first full GPL version since the last version used i2osig headers which were not GPL. Developer Testing version.
V2.1	Internal testing
V2.2	First released version
V2.3	Changes: • Added Raptor Support • Fixed bug causing system to hang under extreme load with • management utilities running (removed GFP_DMA from kmalloc flags)
V2.4	First version ready to be submitted to be embedded in the kernel Changes: • Implemented suggestions from Alan Cox • Added calculation of resid for sg layer • Better error handling • Added checking underflow conditions • Added DATAPROTECT checking • Changed error return codes • Fixed pointer bug in bus reset routine • Enabled hba reset from ioctls (allows a FW flash to reboot and use the new FW without having to reboot) • Changed proc output

TODO

- Add 64 bit Scatter Gather when compiled on 64 bit architectures
- Add sparse lun scanning
- Add code that checks if a device that had been taken offline is now online (at the FW level) when test unit ready or inquiry command from scsi-core
- Add proc read interface
- busrescan command
- rescan command
- Add code to rescan routine that notifies scsi-core about new devices
- Add support for C-PCI (hotplug stuff)
- Add ioctl passthru error recovery

Notes

The DPT card optimizes the order of processing commands. Consequently, a command may take up to 6 minutes to complete after it has been sent to the board.

The files dpti_ioctl.h dptsig.h osd_defs.h osd_util.h sys_info.h are part of the interface files for Adaptec's management routines. These define the structures used in the ioctls. They are written to be portable. They are hard to read, but I need to use them 'as is' or I can miss changes in the interface.