The 3Com Etherlink Plus (3c505) driver.

This driver now uses DMA. There is currently no support for PIO operation.
The default DMA channel is 6; this is _not_ autoprobed, so you must
make sure you configure it correctly. If loading the driver as a
module, you can do this with "modprobe 3c505 dma=n". If the driver is
linked statically into the kernel, you must either use an "ether="
statement on the command line, or change the definition of ELP_DMA in 3c505.h.

The driver will warn you if it has to fall back on the compiled in
default DMA channel.

If no base address is given at boot time, the driver will autoprobe
ports 0x300, 0x280 and 0x310 (in that order). If no IRQ is given, the driver
will try to probe for it.

The driver can be used as a loadable module.

Theoretically, one instance of the driver can now run multiple cards,
in the standard way (when loading a module, say "modprobe 3c505
io=0x300,0x340 irq=10,11 dma=6,7" or whatever). I have not tested
this, though.

The driver may now support revision 2 hardware; the dependency on
being able to read the host control register has been removed. This
is also untested, since I don't have a suitable card.

Known problems:
 I still see "DMA upload timed out" messages from time to time. These
seem to be fairly non-fatal though.
 The card is old and slow.

To do:
 Improve probe/setup code
 Test multicast and promiscuous operation

Authors:
 The driver is mainly written by Craig Southeren, email
 <craigs@ineluki.apana.org.au>.
 Parts of the driver (adapting the driver to 1.1.4+ kernels,
 IRQ/address detection, some changes) and this README by
 Juha Laiho <jlaiho@ichaos.nullnet.fi>.
 DMA mode, more fixes, etc, by Philip Blundell <pjb27@cam.ac.uk>
 Multicard support, Software configurable DMA, etc., by
 Christopher Collins <ccollins@pcug.org.au>