Mach4 UK02p21 Release Notes

November 3, 1995

Contents

Overview
Tested Configurations
FTP information
Details

Overview

UK02p21 is a public release of the so−called ‘‘Mach4’’ kernel for the Intel x86 processor family. Mach4 is one component of the Flux project at the University of Utah Computer Science Department. The focus of the x86 branch has been to make Mach more usable on PCs, with particular emphasis on ease of building, compatibility with a number of OS environments and additional hardware support. This release reflects the latest development in those areas.

New in UK02p21, since p13:

- Compatibility: support unmodified Linux network drivers in Mach, from Shantanu Goel of Columbia.
- Booting: kernel has been mostly updated to conform to MultiBoot−0.3, BSD boot adapter code passes along root device and command line options from Mach, FreeBSD and NetBSD bootblocks to Mach, and Linux boot adapter passes LILO command line options and root device to Mach.
- Kernel debugger again works (a.out only).
- Remote console support again works.
- Native drivers: added a contributed NE2000 driver; IDE (nhd) device driver uses LBA mode on drives that support it.
- Various other random things. Note the warnings re building Lites and the Linux boot adaptor code.

Tested Configurations

All configurations were built using i386−mach cross−build tools, which can be found at: ftp://flux.cs.utah.edu/flux/binaries/i386.

The kernel was tested with Lites−1.1.u2 running on the following base systems:

FreeBSD 2.0.5
FreeBSD 2.1.0−950726−SNAP
NetBSD−1.0

The system was also booted on a system running UX42 on a CMU Mach base.

Ftp Information

Mach4 and Mach4−i386

Sources
- ftp://flux.cs.utah.edu/flux/mach4−UK02p21.tar.gz
- ftp://flux.cs.utah.edu/flux/mach4−i386−UK02p21.tar.gz
- ftp://flux.cs.utah.edu/flux/patch−UK02p13−UK02p21.gz

Intel x86: BSD bootable Binaries
- ftp://flux.cs.utah.edu/flux/binaries/i386/mach4/Mach
- ftp://flux.cs.utah.edu/flux/binaries/i386/mach4/Mach+linuxdev

Intel x86: Linux bootable Binaries
- ftp://flux.cs.utah.edu/flux/binaries/i386/mach4/zMach
- ftp://flux.cs.utah.edu/flux/binaries/i386/mach4/zMach+linuxdev

---

Details

Linux Device Drivers:

Shantanu Goel (goel@cs.columbia.edu) has modified Mach to accept mostly unchanged Linux network device drivers. He has work in progress to support SCSI host adapters and all block devices, which should be available "soon." All this work is described in the paper ‘‘Linux Device Driver Emulation in Mach’’ by Shantanu Goel and Dan Duchamp of the Columbia Mobile Computing Laboratory. The paper will be presented at 1996 USENIX Annual Technical Conference. The paper will be available by November 16, 1995, at which time we’ll add a link to it here.

To enable the Linux device driver support, use ‘‘−−enable−linuxdev’’ when configuring Mach4. You cannot mix native network drivers with the Linux network drivers.

Be aware that since the drivers are from Linux, they are all covered by the GPL.

Testing status

The drivers that have been tested at Columbia are:

Western Digital 8003 and clones
SMC Ultra
NE2000 and clones

AMD LANCE based boards
3Com 3c503 (Etherlink II)

We have tested the linuxdev code with HP PCLAN Plus and NE2000 clones, and while they work, we have had problems on fast machines and suggest that you use the native Mach ethernet drivers if you can. Please report your experiences with other hardware.

No PCI boards work yet, since there is currently no support for PCI in Mach. (That will also be provided "soon" via more linuxdev work.)

Complete list of Linux network drivers in the source tree:

de4x5.c: A DIGITAL DE425/DE434/DE435/DE500 ethernet driver for Linux.
depca.c: A DIGITAL DEPCA & EtherWORKS ethernet driver for linux.
ewrk3.c: A DIGITAL EtherWORKS 3 ethernet driver for linux.
3c501.c: A 3Com 3c501 ethernet driver for linux.
3c503.c: A shared−memory NS8390 ethernet driver for linux.
3c507.c: An EtherLink16 device driver for Linux.
3c509.c: A 3c509 EtherLink3 ethernet driver for linux.
8390.c: A general NS8390 ethernet driver core for linux.
ac3200.c: A driver for the Ansel Communications EISA ethernet adaptor.
apricot.c: An Apricot 82596 ethernet driver for linux.
at1700.c: A network device driver for the Allied Telesis AT1700.
atp.c: Attached (pocket) ethernet adapter driver for linux.
e2100.c: A Cabletron E2100 series ethernet driver for linux.
eepro.c: Intel EtherExpress Pro/10 device driver for Linux.
eexpress.c: Intel EtherExpress device driver for Linux.
hp−plus.c: A HP PCLAN/plus ethernet driver for linux.
hp.c: A HP LAN ethernet driver for linux.
lance.c: An AMD LANCE ethernet driver for linux.
ne.c: A general non−shared−memory NS8390 ethernet driver for linux.
seeq8005.c: A network driver for linux.
smc−ultra.c: A SMC Ultra ethernet driver for linux.
tulip.c: A DEC 21040 ethernet driver for linux.
wd.c: A WD80x3 ethernet driver for linux.
znet.c: An Zenith Z−Note ethernet driver for linux.
Linux ethernet device driver for the 3Com Etherlink Plus (3C505)
Linux driver for the D−Link DE−600 Ethernet pocket adapter.
Linux driver for the D−Link DE−620 Ethernet pocket adapter.
hp100.c: Hewlett Packard HP10/100VG ANY LAN ethernet driver for Linux.
net−3−driver for the NI5210 card (i82586 Ethernet chip)
ni6510 (am7990 ’lance’ chip) driver for Linux−net−3 by MH
Schneider & Koch G16 Ethernet Device Driver for
AT&T GIS (nee NCR) WaveLAN card.

Booting:

BSD and Mach

Booting from Mach, NetBSD and FreeBSD bootblocks should now work as you'd expect. The kernel uses the bootdev and boothowto fields that are passed from the bootblocks, so now you can boot from an alternate root/partition without hardcoding it in setroot.c. Any command line arguments are likewise passed along to the kernel and server.

The old Mach bootblocks don't support loading NMAGIC kernels, which is a problem since that's usually what the Mach boot image is. So, if you're going to be booting from the original Mach bootblocks, you can either build ZMAGIC or OMAGIC kernels, or you'll can make a small change to the Mach bootblocks to recognize NMAGIC binaries:

```
*** boot.c      Wed Sep  1 17:14:33 1993
--- newboot.c       Fri Nov  3 03:34:52 1995
***************
*** 137,143 ****
        read(&head, sizeof(head));
!       if (head.a_magic == 0407)
                poff = 32;
        else if (head.a_magic == 0413 ) {
                poff = 0;
                head.a_text += sizeof(struct exec);
        } else {
--- 137,143 ----
        read(&head, sizeof(head));
!       if (head.a_magic == 0407 || head.a_magic == 410)
                poff = 32;
        else if (head.a_magic == 0413) {
                poff = 0;
                head.a_text += sizeof(struct exec);
        } else {
```

Kernel Debugger:

The kernel debugger is once again useful, supporting symbols and backtraces. The problem was that the symbol table for the kernel wasn't getting loaded, which broke a number of things (now at least the kernel and server symbol tables are being loaded, but the bootstrap and emulator's aren't). However, symbols are only loaded if the kernel is a.out. If you feel you have to build your kernel as ELF, then you don't get symbols.

Remote console support:

Remote console support is working again. This is NOT the same as a serial console; basically it just spews out kernel messages to your favorite serial port. This can be very useful for logging the output of the system and using the kernel debugger over the serial line.

To enable the remote console support, you must change a #define in mach4−i386/kernel/bogus/rc.h. RCLINE is the number of the serial device that you want to use, and RCBAUD is the speed. The options for RCLINE are shown below, as taken from autoconf.c, for the ‘‘com’’ device.

options for RCLINE:

```
RCLINE          info
------          ---------------------------
0               (port 0x3f8/irq 4 DOS COM1)
1               (port 0x2f8/irq 3 DOS COM2)
2               (port 0x3e8/irq 5 DOS COM3)
3               (port 0x2e8/irq 9 DOS COM4)
```

NE2000 device driver:

This is a device driver for the ne2000 series of ethernet cards. It was ported from 386BSD by Shang−te Hsu of the Computer System Lab, Dep. of Computer & Information Science National Chiao Tung University in Hsinchu, Taiwan. The driver was sent to us by Terrence Mitchem (mitchem@sctc.com). We’ve only tested the driver with an ne2000 clone card, but the original README said that it supported the ‘‘NE1000, NE2000, DE−100, DE−200, DE−220’’ cards.

The driver is configured to look for cards set up as:

(ne0) port 0x280, memory addr 0xd0000, irq 5
(ne1) port 0x300, memory addr 0xd0000, irq 10

If you need to change this, go into mach4−i386/kernel/i386at/autoconf.c and you’ll find a section that looks like this:

```
#if NNE > 0
 {&nedriver, "ne", 0, neintr, 0x280,0x4000,0xd0000,
      '?',    0,   -1,    -1,    0,   0,        0,   SPL_SIX, 5},
 {&nedriver, "ne", 1, neintr, 0x300,0x4000,0xd0000,
      '?',    0,   -1,    -1,    0,   0,        0,   SPL_SIX, 10},
#endif NNE > 0
```

Change the 4th, 6th and/or the last (port, mem, irq) to match your card and recompile the kernel. If you want to disable the driver or want to add more than one card, look in mach4−i386/kernel/bogus/ne.h and change the #define to either be 0, or greater than one. The number after the ‘‘ne’’ is the number that it will be configured for if found. So, a card at port 0x280 irq 5 will be configured as ne0.

Other Changes, in no particular order:

- To build Lites with the include files that come with this release, you'll have at least one problem. EFL_USER_SET is no longer exported in include/mach/machine/eflags.h, so compiling lites/emulator/i386/e_exception.c will give you an error. Since the kernel always sets the efl state to be EFL_USER_SET, Lites shouldn't bother with setting it. The fix is to just replace EFL_USER_SET with 0.
- Use the .code16 directive of gas instead of AS86, LD86. This mostly affects the Linux boot adapter code. You **MUST** have a recent snapshot of gas to compile the kernel because of this. Check ftp://flux.cs.utah.edu/flux/gas−950428.tar.gz.
- Small addition to trap.c to support NetBSD−current and Linux system call traps (for Lites).
- Modifications to compile with gcc 2.7.0.
- Support for building ELF kernels (use ‘‘−−enable−elf’’ to turn this on). Note that the kernel debugger and symbols currently don't work for ELF kernels.
- Now uses libmach_exec, which supports ELF and a.out. This is part of the ELF support, so that we can exec an ELF kernel/bootstrap.
- Cleaned up the use of EXT/LEXT/ENTRY macros in the kernel.
- Only have to specify ‘‘−−with−mach4=’’ if mach4 is in a strange place.
- Various other makefile changes, too numerous to detail...
- Other bug fixes, of course.

Stephen Clawson <*sclawson@cs.utah.edu*>
November 3, 1995