

Linux/m68k Frequently Asked Questions

Chris Lawrence <faq@linux-m68k.org>

Version 2.2.0, 25 February 1999

This is the Frequently Asked Questions file for the port of the Linux operating system to Motorola 680x0-based systems. It also provides some information about the Linux port for PowerPC-based Amigas, Linux/APUS. This is version 2.2.0 of this FAQ, which documents Linux/m68k and Linux/APUS kernel versions up to and including 2.0.36 (stable) and 2.2.1-pre2 (developmental; new stable release candidate).

Contents

1	What's New in this version of the FAQ	6
2	Introduction	7
2.1	What is this document?	7
2.2	I'm running Linux/ APUS on my Amiga; why should I bother reading this FAQ?	8
2.3	Who Are You? What Do You Want?	8
2.4	Where can I get the latest version of this FAQ?	8
2.5	Is the FAQ available in other formats?	9
2.6	How are FAQ revisions numbered?	9
2.7	What versions of Linux/ m68k are covered by this FAQ?	10
2.8	General Linux Questions and Answers	10
2.8.1	What is Linux?	10
2.8.2	How does Linux consist of Linux and some other stuff?	11
2.8.3	What's the difference between MkLinux and Linux/m68k?	11
2.9	Misconceptions about Linux/ m68k	12
2.9.1	The Linux/ m68k port is "under development" or "experimental"	12
2.9.2	Linux/ m68k isn't popular	12
2.9.3	Porting Linux/ m68k to my system is useless because there is *BSD for my system	12
2.10	A bit about the Linux/ m68k platforms	12
2.10.1	Amiga	12
2.10.2	Atari	13
2.10.3	Macintosh	13
2.10.4	Motorola VMEbus	13
2.10.5	NeXT workstations	14
2.10.6	Other systems	14

2.11	Who's using Linux/m68k?	14
3	A Brief History of Linux and Linux/m68k	15
3.1	MonoCPUism: Linus creates Linux	15
3.2	The Linux/m68k Trinity: Hamish, Roman and Jes	15
3.3	In Their Own Words	15
3.4	The Ghost of Linux Future	16
4	General requirements to run Linux/m68k	16
4.1	Processor	16
4.2	RAM	17
4.3	Hard Disk	17
4.4	Software	18
4.5	Amiga Hardware Support	18
4.6	Atari Hardware Support	20
4.7	VME Hardware Support	21
4.7.1	Motorola MVME166 and MVME167 boards	21
4.7.2	Motorola MVME162 board	22
4.7.3	Motorola MVME147 board	22
4.7.4	Other VME Boards: VME 17x, BVME4000/6000	22
4.8	Macintosh Hardware Support	22
4.9	Sun 3 Workstation Hardware Support	23
4.10	NeXT Workstation Hardware Support	23
4.11	Hewlett-Packard (HP)/Apollo Domain Workstation Hardware Support	24
4.12	HP 9000/300 Workstation Hardware Support	24
4.13	Unsupported 680x0 Hardware	24
4.14	Support for other processors	24
4.15	Definitely Unsupported Hardware: The Hall of Shame	25
5	Future Development	25
5.1	General	25
5.2	Amiga	26
5.3	Atari	27
5.4	VME	27
5.5	Other Systems	27

6	The Kernel	27
6.1	Recompiling Linux	27
6.1.1	Finding The Sources	28
6.1.2	What You Need to Recompile	28
6.1.3	How to Compile	28
6.1.4	Submitting Kernel Changes	29
6.1.5	Bug Reports	30
6.2	Known kernel bugs	30
6.2.1	TT-FPU bug	30
6.2.2	Amiga with GVP 16-bit RAM	30
6.2.3	Zorro II DMA Bug on A3640 rev 3.0	31
6.2.4	MC68060 performance issues	32
7	Common problems	32
7.1	I can't find the man page for XXX	32
7.2	How can I access my SCSI devices?	33
7.3	How do I access my IDE devices?	33
7.3.1	How do I use an IDE CD-ROM?	34
7.3.2	How do I use an IDE tape?	34
7.4	My SCSI bus locks up when the kernel probes for devices	34
7.5	I displayed a binary file, and now my console is totally screwed up	35
7.6	Can I use both ELF and a.out libraries/binaries in my system?	35
7.7	"less" behaves oddly when I press a key	36
7.8	What are the current major/minor device numbers for /dev/xxx?	36
7.9	How can I tell an a.out binary from an ELF one?	36
7.10	GCC complains that it can't find shared libraries while linking	36
7.11	How do I byte-swap an ext2 filesystem?	36
7.12	My kernel hangs at the login prompt	36
7.13	I just upgraded to 2.1.21 (or later) and modules don't work	37
7.14	Where can I get Linux/m68k on CD-ROM?	37
7.15	How do I patch my kernel?	37
7.16	How do I patch a generic Linux kernel tree to work with Linux/m68k?	38
7.17	How can I get my Jaz drive to work with Linux (or another OS...)?	38
7.18	When I use dpkg or tar, I get messages about a "broken pipe"	39

7.19	What is the current status of FPU emulation?	39
7.20	I can't boot from a ramdisk	39
7.21	I can't execute programs in my current directory	40
7.22	Video Questions	40
7.22.1	How do I choose what video mode to use with Linux?	40
7.22.2	When I run X, it complains about invalid modes	40
7.22.3	Ok, I've got fbset, but I can't create any video modes	40
7.22.4	How do I create the framebuffer device nodes?	41
7.22.5	How do I make sure that I do not damage my monitor when running X?	42
7.23	When I try to run the X Window System on 2.0.33, I get an error	42
7.24	Kernel 2.1.X doesn't compile out of the box for me	42
7.25	I get strange crashes with kernel 2.1.X and two IDE drives	43
7.26	Internationalization questions	43
7.26.1	How do I set a xxx keymap? (xxx = German, French, Swedish, ...)	43
7.26.2	How do I create a xxx keymap?	44
7.26.3	How do I set up my shell to use non-ASCII characters?	44
7.26.4	How do I use locales?	44
7.26.5	How do I use my keymap in X?	45
7.27	I installed glibc and now I get errors about undefined references.	45
7.28	ps and top do not display the associated tty numbers, just a '?'.	46
7.29	"make menuconfig" does not work!	46
7.29.1	For Watchtower	46
7.29.2	For Debian	46
7.30	How do I use the loop-back device?	46
7.31	Floppies and modules	47
7.31.1	I have compiled floppy support as a module but it doesn't work	47
7.31.2	Which aliases do I put to /etc/conf.modules?	47
7.32	Why is there so little information on the web about the BrandX port?	47
7.33	How do I use my mouse in X or with GPM?	48
7.34	My English isn't very good. Can I read this FAQ in my native language?	48
7.35	How do I uncompress the pre-compiled kernels outside of Linux?	48
7.36	Where's Netscape for Linux/m68k?	48
7.37	dpkg problems with 2.1/2.2 kernels	48
7.38	Can I install Debian (or Red Hat) over an existing Watchtower installation?	48

7.39	What's the difference between glibc and libc6?	49
7.40	How do I install the X Window System?	50
7.41	How can I save the kernel messages when the system crashes?	50
7.42	Where can I get an MMU or FPU for my computer?	51
8	Amiga-specific questions	51
8.1	Where did all my Amiga's chip memory go?	51
8.2	How do I access Amiga files from Linux (and vice versa)?	51
8.3	My SCSI bus locks up when I want to use my DAT drive	52
8.4	Linux recognizes my Amiga's XXX board, but it doesn't work	52
8.5	Amiboot dies when I start it with VMM running	52
8.6	How do I compile Amiboot and Amiga LILO?	52
8.7	How do I access floppies under Linux?	53
8.7.1	mttools	55
8.7.2	formatting	55
8.8	Can I use an Amiga-formatted partition as my root partition?	55
8.9	Can I run Linux on my Phase5 PowerUP card's PowerPC CPU?	56
8.10	Can I use an IDE doubler with Linux/m68k?	56
8.11	What video modes does my card support?	57
8.12	The Amiga memory device driver (z2ram)	57
8.12.1	What is the z2ram device?	57
8.12.2	Ok, now that I know what the device does, how do I use it?	59
9	Atari-specific questions	60
9.1	I can't use ttyS3 and ttyS4 simultaneously on my Atari	60
9.2	I'm having problems with my Falcon with Afterburner 040. Any tips?	60
9.3	I'm having strange problems with my PAK030 board in my ST!?!	61
9.4	How do I access my Atari SLM laser printer?	61
9.5	How do I access my ACSI drives?	61
10	Macintosh-specific questions	62
10.1	Where the **** is the Mac site?	62
11	Other sources for information, sources and binaries	62
11.1	Installation Guides	62
11.2	Other Documents	62

11.3 Newsgroups	62
11.4 Mailing Lists	63
11.5 WWW sites	63
11.6 FTP sites	64
11.7 Modem	65
11.8 Distributions	65
11.9 IRC (Internet Relay Chat)	66
11.10 Magazines	66
11.11 Books	66
12 Famous last words	66
12.1 Credits	66
12.2 Copyright and License	67

1 What's New in this version of the FAQ

If you've read a previous version of the FAQ, here's what's changed in the past few revisions:

- 2.2.0
 - Jes has released Linux/m68k kernels 2.0.36 and 2.2.1-pre2. The 2.2 series is considered “semi-stable” on m68k, and may not work for all hardware combinations yet.
 - Debian/m68k 2.1 will be released on 2 March 1999.
 - Richard Zidlicky has ported Linux/m68k to the Q40, a 68040-based computer with an ISA bus being made in Germany. Patches relative to 2.2.1-pre1 are at `<ftp://ftp.uni-erlangen.de/pub/Linux/680x0/q40/>`.
 - Added lots of info on the Amiga memory device driver (z2ram) and getting MMUs and FPU's for 680x0-based systems in the appropriate Questions sections.
 - New FTP site mirror: `<ftp://ftp.linuxberg.com/pub/distributions/Linux-m68k/>`.
 - New Linux/m68k mirror: `<http://www.lordsutch.com/linux/>`.
 - Credited the originator of the concept of going home and eating popcorn.
 - Updated the names of the Debian X packages to reflect the latest Great X Reorganization.
 - The FAQ is now licensed under the GPL and an alternate license that provides for distribution of printed versions and the like without including the SGML source.
 - New section listing devices for which we have encountered a non-disclosure agreement that forbids us from including a driver in the kernel for the device. I strongly recommend either (a) boycotting these devices and their manufacturers or (b) contacting the manufacturer and letting them know you will not purchase the device unless we can release a driver for it.

- There is now a driver for the Amiga Oktagon SCSI card. A patch for this support was released to the mailing list.
 - Shawn D’Alimonte has written a driver for the Amiga ADSC dual serial port card. A patch relative to 2.0.33p11 is available on the mailing list; the driver is included in 2.0.36.
 - New warning about compiling Linus’s kernels for m68k added to the kernel compilation section. Bottom line: if you compile a Linus kernel tree, and m68k doesn’t work, don’t complain to Jes.
 - New question on how to patch one of Linus’s Linux kernel trees to the corresponding Linux/m68k version. Should clarify the purpose of `-native` diffs.
 - Pulled out Strunk and White and fixed a number of that/which problems.
 - Modified the changelog format somewhat.
 - Bumped FAQ revision to 2.2.0.
- 2.0.53
 - Richard Hirst is porting Linux/m68k to another VMEbus 680x0 machine: the Tadpole TP34V.
 - Added a section on people using Linux/m68k in commercial or research applications.
 - Really bumped Linux/m68k unstable version to 2.2.0-pre6.
 - 2.0.52: Added information on how to use your console keymap in X.
 - 2.0.51: Included information on how to access the Linux/m68k CVS repository. Thanks to David Kilzer for pointing out the omission and sending me the access instructions.
 - 2.0.50
 - `sunsite.unc.edu` is now `metalab.unc.edu`.
 - Added info on Permedia2 framebuffer (`pm2fb`)
 - Added users of “PowerPC-based Amigas” as a target audience for this FAQ, since most of the Amiga-specific information in the FAQ applies to both Linux/m68k and Linux/APUS.
 - Most references to 2.1 kernels now reference 2.2 kernels as well (since the 2.2 pre-release kernels are in the 2.1 kernel series).
 - Updated VME info to reflect that Richard is now maintaining the VME147 port.
 - Bumped unstable kernel version for m68k (to 2.2.0-pre4).

2 Introduction

2.1 What is this document?

This is a copy of the Linux/m68k Frequently Asked Questions file (or FAQ). Since it probably contains errors (typographical and logical), outdated and missing information, and other significant problems, I ask that you send feedback and corrections to me.

What it is supposed to do is answer common questions about Linux/m68k, in the hope (infinitesimal though it might be) that these questions will not be asked again (since they’ve already been answered here). It is highly advisable to read this FAQ thoroughly before asking questions in the newsgroup, on

the mailing lists, or directly; failure to read this FAQ in its entirety may result in either no response or an extremely hostile response to your questions, depending on whom you ask and what mood that person is in.

This FAQ is not intended to serve as an introduction to Linux; nor is it intended to explain how to administer a Linux-based system. To find out more about these topics, please read the standard Linux books and manuals. I particularly recommend Matt Welsh's two books: *Linux Installation and Getting Started* published by the LDP and available in print from SSC <<http://www.ssc.com/>>, and *Running Linux, Second Edition* available from O'Reilly and Associates <<http://www.oreilly.com/>> (cowritten with Lar Kaufman), which expands greatly on the content of the previous book (if you can only afford one Linux book, get this one). See section 11.11 (Books) for more details.

For other questions about Linux in general, I recommend that you read the Linux INFO-SHEET. This document, along with many others about Linux (including instructions for specific applications), is available at *the Linux Documentation Project's home page* <<http://metalab.unc.edu/LDP/>>.

2.2 I'm running Linux/APUS on my Amiga; why should I bother reading this FAQ?

The short answer is that Jesper Skov (or someone else) will yell at you if you ask a question that's already answered here.

The long answer: While this FAQ is primarily about the Linux/m68k project, it also includes information specific to the Amiga platform that is not included in any other FAQ. Since the APUS project uses the same Amiga hardware support code as the Linux/m68k project, the Amiga-specific information in this FAQ applies to both ports. Accordingly you should read both this FAQ and Jesper's Linux/APUS FAQ.

2.3 Who Are You? What Do You Want?

This FAQ was created and originally maintained by Jörg Mayer <jmayer@telemation.de>. It is now maintained by Chris Lawrence <lawrenc@clark.net>. The current maintainer of the FAQ can always be reached at <faq@linux-m68k.org>.

2.4 Where can I get the latest version of this FAQ?

The latest revision is available in HTML format, suitable for reading with a World Wide Web browser (such as Netscape, Lynx, Arena, etc.), at <<http://www.linux-m68k.org/faq/faq.html>>, <<http://home.olemiss.edu/~clawrenc/linux/faq/faq.html>>, <<http://www.lordsutch.com/linux/faq/faq.html>>, <<http://www.se.linux-m68k.org/faq/faq.html>>, or <<http://www.clark.net/pub/lawrenc/linux/faq/faq.html>>. It is also mirrored daily in Germany at <<http://ftp.uni-erlangen.de/pub/Linux/680x0/FAQ/faq.html>> and in Norway at <<http://amiga.nvg.org/linux/mirrors/lawrenc/faq/faq.html>>.

There is a French translation of the FAQ available at <<http://www.mygale.org/~atari/Linux68k/Faq/>>, translated by Christian Jacolot.

Il y a un traduction français du FAQ; on peut le retrouver à <<http://www.mygale.org/~atari/Linux68k/Faq/>>. Il est traduit par Christian Jacolot.

A pointer to this FAQ is supposed to be posted every two weeks to the Usenet newsgroups *comp.os.linux.m68k*, *comp.unix.amiga*, *maus.os.linux68k*, *comp.arch.bus.vmebus*, *comp.sys.amiga.misc*, *comp.sys.atari.st*, *comp.sys.m68k*, *comp.answers*, and *news.answers*. The entire FAQ is no longer posted to Usenet because it has become ridiculously large.

2.5 Is the FAQ available in other formats?

The FAQ is available in the following formats. Note that many of these files are compressed using the GNU gzip program; you will need a copy of gzip (or gunzip, or another compatible tool) to access these files.

- HTML: `<http://www.linux-m68k.org/faq/faq.html>`.
- Plain Text: `<http://www.clark.net/pub/lawrencc/linux/faq/faq.txt.gz>` (compressed with GNU gzip).
- Printable (US letter size; should also print on A4 if you trick the printer into thinking the paper is letter size):
 - PostScript: `<http://www.clark.net/pub/lawrencc/linux/faq/faq.ps.gz>` (compressed with GNU gzip).
 - Portable Document Format (PDF): `<http://www.clark.net/pub/lawrencc/linux/faq/faq.pdf>` (readable with PDF viewers such as Adobe Acrobat or xpdf).
 - TeX Device Independent (DVI): `<http://www.clark.net/pub/lawrencc/linux/faq/faq.dvi.gz>` (compressed with GNU gzip; use a DVI converter (dvips, dvi2pdf...) to print).
- Source (for these to be useful, you will need a working *SGMLTools 1.0* `<http://www.sgmltools.org/>` installation)
 - SGML (LinuxDoc DTD): `<http://www.clark.net/pub/lawrencc/linux/faq/faq.sgml.gz>` (compressed with GNU gzip).
 - LaTeX (useful for reformatting to A4 or other sizes): `<http://www.clark.net/pub/lawrencc/linux/faq/faq.tex.gz>` (compressed with GNU gzip). You'll need some of the SGMLTools style files to successfully run it through LaTeX.

2.6 How are FAQ revisions numbered?

The version number has three parts, formatted as follows: X.Y.Z

- X.Y is the current stable kernel version number (currently 2.0).
- Z is the revision number of this FAQ since the last stable kernel version change.

For example, FAQ 2.0.43 was the 43rd revision of the FAQ after the 2.0 kernel was released. When the first official 2.2 kernel was released, the next FAQ version was FAQ 2.2.0.

Note that this FAQ is often updated between Usenet posts. Interim versions of the FAQ are made available at the usual places.

2.7 What versions of Linux/m68k are covered by this FAQ?

This FAQ documents the 2.0 and 2.2 series of kernels. If you are currently using a 0.x or 1.2 series kernel, I recommend that you read the *Linux/m68k 2.0 announcement* <<http://www.linux-m68k.org/announce-2.html>> (and all the documents it says to read) and then upgrade to a 2.0 kernel. The 0.x and 1.2 series of kernels are obsolete and completely unsupported.

Most things said about 2.0 kernels apply also to the 2.1 development series and 2.2 semi-stable kernels, unless otherwise noted. I recommend against trying to use the 2.1 and 2.2 kernels until you are accustomed to using Linux, or you need to use hardware that's not supported in a 2.0 kernel.

The latest versions of the kernel and boot loaders, as of this FAQ release:

- Mainstream Linux kernel (released by Linus; current information at <<http://www.cs.helsinki.fi/cgi-bin/linuxversion>>):
 - Stable: 2.0.36
 - Semi-Stable: 2.2.1
- Linux/m68k kernel (released by Jes):
 - Stable: 2.0.36
 - Developmental: 2.2.1-pre2
- Amiboot (AmigaOS bootstrap): 5.6
- Ataboot (Atari bootstrap): 3.1
- Penguin (Mac bootstrap): 16
- Amiga LILO (low-level Amiga bootstrap): 2.2
- MVME LILO (low-level VMEbus bootstrap): 1.0

For current versions of other software, please refer to the Changes document included in the kernel sources (also available at <<http://www.linuxhq.com/>>).

2.8 General Linux Questions and Answers

This is a brief section that answers a few questions about the Linux operating system. It is not intended to replace the real documentation about Linux, however.

2.8.1 What is Linux?

Linux is a freely-distributable kernel and operating system that works virtually the same as UNIX. Unlike all other available truly UNIX-like operating systems (this means those that provide memory protection and virtual memory), it is built from the ground-up from scratch to comply with open standards. Currently, Linux complies with virtually all of the POSIX.1 standard (the only completely vendor-independent standard), and work is underway to finish work on compliance with the System V Interface Definition (SVID) and other commercially-established standards.

Linux was started in 1991 by Linus Torvalds, who at the time was an undergraduate student in Computer Science at the University of Helsinki in Finland. While Linus is no longer a starving college student (he now works for *Transmeta* <<http://www.transmeta.com/>>, a highly-secretive Silicon Valley company), he continues to coordinate the work on the kernel and makes significant contributions of his own, particularly on the Alpha and SMP (symmetric multiprocessing) code. The names of many of the other people who have contributed to the Linux kernel can be found in the CREDITS and MAINTAINERS files that are included with the Linux kernel sources.

More of Linux's history (particularly the history of Linux/m68k) is covered in the next section of the FAQ.

2.8.2 How does Linux consist of Linux and some other stuff?

The Linux kernel is vaguely equivalent to the Kickstart under AmigaOS. It provides basic services to the operating system, but that's about it. Unlike AmigaOS, it requires at least one other program to launch (a shell [command line interpreter] or a special program called `init`). Without another program, you'll never even get to a command prompt.

The Linux operating system is a collection of programs (such as interpreters, shells, utilities, applications, and daemons) and libraries that facilitate user interaction with the system. Much of the Linux operating system is derived from the Free Software Foundation's GNU project and the University of California at Berkeley Source Distribution of Unix (BSD). The Linux OS also includes software from other sources, some of which was written specifically for Linux.

For the most part, I use the term Linux as the generic term for both the operating system that most Linux users use and to refer specifically to the kernel. Others would use "GNU/Linux", or a distribution name (e.g. "Red Hat Linux", "Slackware Linux", or "Debian GNU/Linux"), for the operating system, reserving "Linux" strictly for the kernel. Suffice it to say it's not worth the effort to try to convince me to adopt this alternative terminology (you can start the GNU/Linux/m68k FAQ if you like :-).

Where the distinction between one meaning of Linux and another is unclear, I apologize in advance.

2.8.3 What's the difference between MkLinux and Linux/m68k?

MkLinux <<http://www.mklinux.apple.com/>> is a project sponsored by Apple (in collaboration with the Open Group, née the Open Software Foundation) to build a Microkernel-based Linux kernel for PowerPC (and some other) systems.

Linux/m68k <<http://www.linux-m68k.org/>> is a project to build a monolithic Linux kernel for 680x0 systems. It has no connection with Apple or the OG/OSF (as a matter of fact, Apple, unlike many other manufacturers, has been downright unhelpful with the m68k Linux port).

Unfortunately, the use by some of the term "MacLinux" has added to the confusion and made a lot of people think that MkLinux and Linux/m68k on the Macintosh are the same project. They aren't. Not even close.

2.9 Misconceptions about Linux/m68k

2.9.1 The Linux/m68k port is “under development” or “experimental”

False. Linux/m68k is at least as stable as Linux on Intel, Alpha, PowerPC and Sparc systems (all of which have, like Linux/m68k, at least one “major” distribution available). Furthermore, Linux/m68k was the first stable port of Linux to any other (non-Intel) processor.

There is development on additional hardware drivers and additional machine ports (like implementations for the Macintosh, Apollo and Sun 3), but this is the same “development” that is underway on other platforms.

As an illustration of Linux/m68k’s stability, a recent report on the mailing list said that a 68030-based Amiga 1200 had been running a 2.0.29 kernel for 24 hours a day for 363 days without a system crash, while serving as a web server and providing file, news and mail services to several other machines.

2.9.2 Linux/m68k isn’t popular

False. Over 1600 people have registered using Linux/m68k (and registering is strictly optional; see *the Linux/m68k Registration Site* <<http://www.cs.kuleuven.ac.be/~geert/Linux/m68k>> for up-to-date figures). Many hundreds more are using it on systems without Internet connections.

Linux/m68k’s usage on systems capable of running it is probably equivalent to that of Linux on Intel platforms (on a percentage basis).

Over 330 people participated in the call for votes for the Usenet newsgroup `comp.os.linux.m68k`, which took place in late 1995 (when Linux/m68k was in less widespread use).

2.9.3 Porting Linux/m68k to my system is useless because there is *BSD for my system

False. Linux has many features that make it preferable to NetBSD or OpenBSD. The most impressive feature is that there is virtually no Berkeley code in the kernel: it is written from the ground up to comply with POSIX and other standards (XPG, SVID, etc.), and work is underway to make it a “branded” Unix.

Linux is also highly popular on Intel platforms (to a much greater degree than BSD). This popularity, combined with 99.9% source compatibility, means that virtually any program that runs on Linux/i386 (and doesn’t use non-portable features like SVGAlib) can be compiled and run on Linux/m68k. It also means that you can walk into virtually any bookstore and buy a book specifically about your OS (try that with AmigaOS!).

2.10 A bit about the Linux/m68k platforms

Additional information about a number of 680x0-based systems is available at *Joaquin Menchaca’s hardware pages* <<http://www.angelfire.com/ca2/tech68k/index.html>>

2.10.1 Amiga

The Amiga was the first 680x0-based computer to have Linux ported to it. The first Amiga computer was the Amiga 1000, released in mid-1985. It featured a 68000 processor running at 7.14 MHz, along with 256k

of RAM.

The Amiga line has included quite a few models, including the Amiga 500, 600, 1200, 2000 (and its variants, like the 1500 and 2500), 3000 and 4000. The 3000T and 4000T are tower versions of the 3000 and 4000, respectively.

The Amiga line also includes the CDTV and CD-32 platforms, which are CD-ROM-based Amigas. More recently, clones have appeared, like the DraCo and BoXeR motherboard.

Recent Amigas can be upgraded to use PowerPC 603e and 604e processors in addition to a 680x0 processor using third-party CPU boards.

2.10.2 Atari

The Atari 32-bit series was the second platform to receive an implementation of Linux/m68k. The Atari machines were launched with the release of the ST520 in mid-1985.

The Atari line includes the ST models, TT and Falcon. There have also been a number of Atari clones, including the Medusa.

2.10.3 Macintosh

(revised by David Kilzer)

The Macintosh, introduced in 1984, was the first popular 680x0-based computer. There have been dozens of different 680x0-based Macintoshes.

The port of Linux/m68k to the Macintosh platform is still ongoing, though some systems are usable today with functional SCSI, IDE, Ethernet and console support.

Current gaps in support include FPU-less Macs (the FPU emulator is still a work-in-progress) and most Powerbooks (ADB is not supported yet, though code from the Linux-PPC and MkLinux projects will help greatly).

A fairly comprehensive overview may be found at the Linux/m68k on Macintosh site, <<http://www.mac.linux-m68k.org/>>.

2.10.4 Motorola VMEbus

(written by Richard Hirst)

Motorola has released a number of single-board systems using the 680x0 processors, based on the VME bus standard. More information on these systems is available at *Motorola's web site* <<http://www.mot.com/GSS/MCG/products/boards/>>.

(More information from a later post:)

I have a VME system based on Motorola MVME boards. Follow the links from [www.sleepie.demon.co.uk](http://www.sleepie.demon.co.uk/linuxvme/index.html) <<http://www.sleepie.demon.co.uk/linuxvme/index.html>> to find out more about the boards. The boards I use are basically single board computers, which can be plugged into a VME card cage. The interface to the VME is via a chip called the VMEchip2 which provides programmable address windows between the VME bus and the on-board bus. As part of programming the VMEchip2, you specify the AM (Address Modifier) code to use in VME bus cycles. The AM code specifies A24, A32, etc.

VME is used a lot in industrial applications, with various interface boards for digital i/o, etc, so people using Linux on these boards often want to read/write to specific addresses in the VME address space.

Before anyone asks, these boards are expensive (relative to a good PC) - I got mine from work so didn't have to pay for them.

2.10.5 NeXT workstations

The NeXT workstations were produced by NeXT Computer, Inc., starting in the late 1980s and ending in 1994. The workstations were made in two configurations: the NeXT Cube and NeXTstation (a.k.a. "the slab").

The NeXT Cube came in 68030 and 68040-based configurations, while the slabs were produced later and came with 68040's only. 68040-based models came in 25MHz and 33MHz (Turbo) editions.

The basic NeXTs came with 4-grayscale video (black, white and two shades of gray). Color NeXTs are capable of 12-bit color, or 4096-color video output (16 levels of red, green and blue). NeXT also produced the NeXT Dimension board for the cubes, which was capable of 24-bit color.

NeXTs ran the NeXTstep operating system; however, current versions of that OS (now called OpenStep) no longer support the original ("black") m68k-based hardware; this has made a Linux port to the NeXT particularly attractive. More information can be found at the Li/NeXT web site, <<http://www.black.linux-m68k.org/>>.

2.10.6 Other systems

Any takers?

2.11 Who's using Linux/m68k?

Among the thousands of Linux/m68k installations, there are several organizations which use Linux/m68k in their commercial or research endeavors:

- The *European Synchrotron Research Facility* <<http://www.esrf.fr/>> in Grenoble, France, is using Linux/m68k on several Motorola VME single-board computers with the hope of getting more out of their existing VME installation. You can read more about their projects at their *Linux on VME* <<http://www.esrf.fr/computing/cs/sysadmin/rtk/emlinux/vme.htm>> page. (A synchrotron is a type of particle accelerator.)
- *BVM Ltd.* <<http://www.bvmltd.co.uk/>> is offering Linux/m68k as an alternative operating system to OS/9 on their VME single-board computers.
- Several public web servers are running on Linux/m68k boxes, including <<http://amiga.nvg.org/>> and <<http://www.uk.linux.org/>>.

3 A Brief History of Linux and Linux/m68k

More details on the history of Linux can be found the book *Inside Linux* by Randolph Bentson <bentson@grieg.seaslug.org> (see section 11.11 (Books) for more information about this book).

3.1 MonoCPUism: Linus creates Linux

Linux is a freely available operating system for PCs: to be more precise, it is one of many flavors of Unix. Linux is being developed on the Internet by several thousand people, first and foremost by Linus Torvalds <torvalds@transmeta.com>, who created Linux for the 80386 in 1991. Linux is being tested and used by many more (the total is thought to be in the hundreds of thousands, perhaps millions).

At the time, Linus believed that Linux was inherently Intel-specific. A few Amiga and Atari hackers were determined to prove him wrong.

3.2 The Linux/m68k Trinity: Hamish, Roman and Jes

The fun and success of Linux on Intel-based systems inspired Hamish Macdonald <hamish@border.ocunix.on.ca> and Greg Harp to port it to another platform: the Amiga. The first version released to the general public was 0.0.5. While 0.0.8 was current, a few enthusiasts ported that version to the Atari and the two versions were successfully merged with 0.9pl3 (this reads version 0.9 patchlevel 3).

After releasing 1.2.13pl3, Hamish handed the coordination of Linux/m68k over to Roman Hodek <rnhodek@informatik.uni-erlangen.de>. Starting with Hamish's unfinished 1.3.20 port, Jes Degn Sørensen <Jes.Sorensen@cern.ch> started to work on integrating the m68k stuff into 1.3. With 1.3.94 the majority of the m68k stuff was put into the official kernel tree. Work continues to integrate Linux/m68k with the mainstream kernel. Jes continues to be the maintainer of the kernel.

The present 2.0 series releases are of "production quality" and are suitable for general use on the Amiga, Atari and a number of VME platforms. The versions of the 2.1 and 2.2 series are generally as stable as their counterparts on Intel platforms (probably even more stable, as there is more testing between releases).

3.3 In Their Own Words

Here's the story about the beginning in Hamish's words:

I decided to port Linux to my Amiga for a variety of reasons. I have always had an interest in operating systems (my work is in embedded systems for telecommunications). After finishing my Master's thesis, I needed some project to keep me busy, and justify keeping my Amiga. Linux was just getting popular at the time, and I thought it would be fun to port it to my Amiga. So I did. Greg Harp and a few others had been talking for a while about porting Linux to the Amiga. They'd only got a little way into it when I got involved, bringing the work I'd already done myself...

And here about the port to the Atari in Roman Hodek's words:

I'm an old Atari user, but in some dark age of Atari, I also bought a PC... running only Linux, of course! Some time later, I noticed that there was a Linux for m68k (was a version about 0.06 or so), but learned that it was for Amiga only. Already at that time, I thought that it can't be that hard to port this to the Atari, but after some browsing in the sources, I gave up. I just didn't find a point where to start. But at least I subscribed to the MausNet newsgroup "LINUX68K". Several months later, in April '94, Björn Brauel posted an article there, that he has adapted Linux's head.S so it ran on his Falcon (until the first console output :-). I again was interested and asked Björn for the sources. In the next time, we two built some very basic driver, so we could at least see some output on the screen, and the kernel booted until the "unable to mount root". There was no HD driver yet... So I started to write a SCSI driver, and Björn went to IDE.

At that time, we heard that there was another group working on an Atari port. The most important members of this group were Robert de Vries and Andreas Schwab. They've never announced that they're working or how far they are, so we didn't know about them. And we communicated over the MausNet, not the Internet, so they didn't notice us... So we finally had two versions of an Atari port at the same time. Fortunately, we've mostly worked on different parts, so the merged version 0.01pl3 made a big jump in respect to what drivers were available.

The next story is about Martin Schaller: He also ported Linux to the Atari, starting directly from PC Linux, not from the Amiga version. (He didn't have a modem at that time, so no Internet, not even MausNet...) For that he worked totally on his own, he came very far and did a great job. In fall 1994, a German Atari magazine published an article about Linux/m68k. By this Martin heard that there were some more Atari Linux hackers and joined us.

3.4 The Ghost of Linux Future

In the past few years, various groups have formed to support additional machines. The largest, not surprisingly, has been the Macintosh group, which finally got off the ground (after an apparent vaporware "effort" was dismissed as a hoax) in 1996. Now the Mac port is running stably with both 2.0 and 2.1 kernels.

The Amiga support code in Linux/m68k has also been adapted to work with Amigas using PowerPC processors.

4 General requirements to run Linux/m68k

4.1 Processor

You need a Motorola 680x0 processor *with a programmable memory management unit (PMMU)*. There is *no way* to run Linux/m68k without one. This reduces the list of possible processors to 68020+68851, 68030, 68040, 68LC040, and 68060. This list of processors excludes the 68000, 68HC000, 68008, 68010, 68EC020, 68EC030, and 68EC040. It also excludes the CPU32 processors (683x0 series) and the ColdFire processor. Linux/m68k can *never* run on these processors as they lack a PMMU and an interface for an external one (some 68EC030s do have a functioning PMMU and will run Linux; however, their long-term reliability is questionable since Motorola never tested the PMMU). Consequently, all Linux/m68k software is compiled for 68020 or higher CPUs.

Having said this, there has been an effort to create a Linux port that does not require an MMU; it's called uClinux (or Microcontroller Linux), and apparently does what it does quite well, but it is limited in that it can't support virtual memory or memory protection. uClinux and Linux/m68k binaries are not interchangeable. You can learn more about uClinux at *Paul Coene's site* <<http://www.esat.kuleuven.ac.be/~pcoene/index.html>>.

If you can't figure out whether your computer has the "right" processor built-in, please see the next section of the FAQ.

At the moment, current versions of Linux/m68k require a FPU (floating point unit) to run. While there is some unsupported code that attempts to emulate the FPU (see 7.19 (FPU emulation), it is not reliable and its use is highly discouraged. If you have a 68020 or 68030, you need a 68881 or 68882 coprocessor as well; if you have a 68040 or 68060, you need the full version of the CPU (not the EC or LC version). So until the new FPU emulator is released, the following CPUs (listed by their full Motorola part number) are the only CPUs that support Linux/m68k:

- MC68020 + MC68851 MMU + separate FPU (MC68881/MC68882)
- MC68030 + separate FPU (MC68881/MC68882)
- MC68040
- MC68060

4.2 RAM

It is possible (but very difficult) to boot Linux/m68k with as little as 2 MB. Now you know that the kernel works on your system: that's it. If you want to work with it you should have at least 4 MB (8 MB is probably the working minimum if you want to run an X display on your machine); booting from most ramdisks will require at least 5 MB.

Notes:

- On the Amiga only the size of your Fast RAM is relevant. The Chip RAM is only used internally for graphics, sound and floppy data; it is not used by normal programs.
- If you have an Amiga with 16-bit expansion RAM on a Zorro card, see 6.2.2 (16-bit RAM).
- On Ataris, ST RAM is usable on most systems.

4.3 Hard Disk

If you want to do much more than just boot Linux/m68k you will need 60 MB or more of free space on your hard disk and a supported hard disk controller. Add another 20 MB of disk space to the base requirements to use X.

In addition, you'll need some swap space. Any amount will do (and you will need less if you have more RAM). 16 MB of swap space is a fairly reasonable size for most systems; the kernel is limited to using 128 MB of swap space.

Due to the general law that your files will expand and multiply to fill your available disk space, I recommend getting the largest disk you can afford (and your system can handle reliably).

4.4 Software

Amiga: In order to run amiboot (i.e. to boot Linux) you need AmigaOS 2.0 or higher (expansion.library V36 or later). amiboot works with SetPatch and RsrvMem (part of Emplant) running. Additionally, you can start amiboot with VMM running (see 8.5 (Known Bugs) for how to do this). On a 68060, you will probably need the 68060 libraries that came with your CPU card installed as well. Amiga LILO may have different requirements; consult its documentation.

Macintosh: A minimal copy of System 6.0.X, System 7.X or Mac OS 8.X is required to boot the Mac. After the Mac boots, the Penguin application is run to load Linux/m68k and boot it.

Atari: You need some version of TOS available to boot the system.

4.5 Amiga Hardware Support

All models are supported, with the exception of the DraCo clone; BoXeR motherboards may work, but have not been tested. A3000(T) means A3000 or A3000T tower; A4000(T) means A4000 or A4000T tower. A2000 implies all A2000-derivatives (A1500, A2500, etc.). Note that you will need a supported CPU installed.

The following Amigas have an acceptable processor built-in: A2500, A3000, A3000T, A4000/040 (not the A4000/030, which ships with a 68EC030 processor, unless upgraded with an 040 or 060), A4000T/040 and A4000T/060. There has been a report that the last A4000/040s produced by Commodore were shipped with 68LC040's; these computers will not run Linux/m68k unless you upgrade the CPU itself (swap the chip) or swap the CPU card. All other Amiga models (even the A1000) can have the correct processor (or processors, as needed) added on; see 4.1 (the CPU listing) for a comprehensive list of the CPU/MMU/FPU combinations that are supported by Linux/m68k.

You can check whether you have a working PMMU or not using a program called "lawbreaker" included in the Enforcer package (available from Aminet).

"Cards" generally refer to Zorro cards, which can be installed in any big box Amiga (and in various tower kits for other models). Some cards here are PCMCIA devices, which will only work in the A600 and A1200.

Supported built-in hardware:

- SCSI: A3000(T) (WD33c93), A4000T (53c710)
- IDE: A1200, A4000(T) (Gayle controller). Some IDE doublers are also supported.
- Serial port
- Parallel port
- Mouse (2 or 3 buttons; trackballs that require no special drivers will also work)
- Keyboard (keymaps for several keyboards are included in Debian/m68k kbd package; some others are available at <ftp://ftp.uni-erlangen.de/pub/Linux/680x0/bin/system/keymaps/>)
- OCS, ECS and AGA chipset graphics (amifb)
- Floppy drives (internal and external; double and high density)
- Paula sound

- Real-time clock: A2000, A3000(T) and A4000(T). Add-on clocks for the A500, A600, A1000 and A1200 that emulate the built-in clocks on the other systems should also work.

All accelerator cards with compatible CPUs are supported (this does not mean that all features of the CPU card, like a SCSI controller, will be available, but the CPU itself should work).

All RAM expansions are supported (but see 6.2.2 (16 bit RAM expansions)). Non-AutoConfig expansions may not be recognized under all circumstances without a “memory file” (see 6.2.2 (16 bit RAM expansions) or the amiboot documentation for details).

Supported IDE cards (experimental):

- Individual Computers Buddha
- Individual Computers Catweasel

Supported SCSI cards:

- WD33c93-based cards: A590, A2091 and GVP Series II (including HC+8, A4008 and some 030/040 accelerators)
- Phase5 cards: Cyberstorm I and II add-ons (but not Cyberstorm III/PPC built-in, yet); Blizzard 1230IV and 1260 add-ons; BlizzardPPC 603e+; Blizzard 2060; Fastlane Z3.
- 53c710-based cards: Commodore/DKB A4091 and WarpEngine built-in.

Note: If you have an early A4000/040, read 6.2.3 (the A3640 Zorro II Bug).

Supported Ethernet cards:

- Commodore A2065 (a2065)
- VillageTronic Ariadne (ariadne)
- VillageTronic Ariadne2 (kernels 2.0.36+/2.1.124+; ariadne2)
- BSC AlfaData Hydra (aka Hydranet; hydra)
- CNET40 (PCMCIA; apne)
- LinkSys (PCMCIA; apne)

Supported I/O cards:

- BSC AlfaData MultiFaceCard III (serial/parallel)
- GVP IO-Extender (serial/parallel)
- HiSoft Whippet (PCMCIA serial)
- AD5G dual-serial board (2.0.36+)

Supported graphics cards (with device names for 2.1/2.2 kernels); see also *the Framebuffer HOWTO* <<http://www.tahallah.demon.co.uk/programming/prog.html>>:

- Phase5 Cybervision 64 (cyber)
- Phase5 Cybervision 64/3D (virge: 2.0.33+, with patches)
- Phase5 BVisio PPC and Cybervision PPC (pm2fb): See the *Permedia2 framebuffer page* <<http://www.cs.unibo.it/~nardinoc/pm2fb/>> for the files you need and any updates; the driver only works under Linux/APUS at the moment (but should be fairly easy to get running under Linux/m68k).
- Retina Z3 (retz3)
- Cirrus Logic (clgen): See the *CLGen page* <<http://www.informatik.uni-oldenburg.de/~amigo/clgen.html>> for the files you need and any updates; the drivers should be included in most recent kernels. Note that instructions for using the CLGen driver are *not* included in the kernel sources, so you will need the CLGen source package for the documentation, even with recent kernels.
 - Piccolo
 - Piccolo SD64
 - VillageTronic Picasso II
 - VillageTronic Picasso IV
 - GVP Spectrum

4.6 Atari Hardware Support

All models supported, including the Medusa clone. The following Atari models (or clones) have the “right” processor built-in: Atari Falcon, Atari Falcon with Afterburner 040, Atari TT, and Medusa. However, the standard Falcon does not have an FPU built-in (so you will need to add an FPU). Accelerators are available for all Ataris that will upgrade the CPU to an acceptable level; see 4.1 (the CPU listing) for the CPU/MMU/FPU combinations currently supported by Linux/m68k.

Note that in some of the older Atari TTs there is a bug in the PAL controlling the access to the FPU. This may cause crashes (see 6.2.1 (Known Bugs)).

Supported built-in hardware:

- SCSI
- ACSI
- IDE: Falcon
- All serial ports (including the MIDI port)
- Keyboard (several keymaps are available in Debian/m68k kbd package; some others are available at <<ftp://ftp.uni-erlangen.de/pub/Linux/680x0/bin/system/keymaps/>>)
- Mouse
- Parallel port
- Real-time clock
- Floppy drives: DD, HD, ED

- Atari graphics: ST, TT, Falcon, Medusa (atafb)
- System beep
- Sound: TT, Falcon
- ET4000 and Mach64 in Falcon are partially supported (you can't change the resolution).

Linux's Minix FS is compatible with the Minix V2 FS used with MiNT.

Supported RAM cards:

- FX-card (Falcon)
- Magnum (Falcon)

Supported Ethernet cards:

- RieblCard
- PAM's VME boards
- most Lance-based Ethernet cards should work

Screen extenders: Screenblaster, Onscreen work, others should work too.

Other peripherals: Atari Laser Printer

4.7 VME Hardware Support

4.7.1 Motorola MVME166 and MVME167 boards

(written by Richard Hirst)

There are multiple VME boards out there on which Linux/m68k could run. There is currently one port for the Motorola MVME166 and MVME167 boards. These are complete systems on a card, so there's no need to support any external controller cards:

- 68040 at 25 or 33MHz (at least)
- 4 to 32MB DRAM (maybe more)
- CD2401 four channel serial controller
- NCR53C710 SCSI controller
- Parallel port (currently unsupported)
- Intel 82596CA Ethernet controller (under construction)
- VMEchip2 VME interface chip
- PCCchip2 local controller chip

- VSBchip2 (MVME166 only)
- MK48T08 BBRAM and TOD clock
- 68230 PIT

The released source is based on 2.0.29, but it is now being integrated in to the 2.1 development tree. The system will happily loop building the kernel for hours on end with no problems (no '040 MMU bug here!) It takes about 20 minutes to build a kernel.

More information can be found at *Richard's VMEbus port page* <<http://www.sleepie.demon.co.uk/linuxvme/index.html>>.

4.7.2 Motorola MVME162 board

Richard has also ported Linux/m68k to the MVME162 (after the original port was made by Vaughn Skinner and lost in a disk crash). It "basically works as well as the 166 and 167" ports according to Richard.

The needed files can be downloaded from *Richard's VME page* <<http://www.sleepie.demon.co.uk/linuxvme/index.html>>.

4.7.3 Motorola MVME147 board

A port to the MVME147 board was made by Dave Frascione <chaos@mindspring.com>. It has been integrated into Richard's VME port, and can be found at the same page.

4.7.4 Other VME Boards: VME 17x, BVME4000/6000

The VME 17x series of boards are similar to the 16x series, except they have 68060 processors. According to Richard Hirst, the corresponding 16x port should run fine with the data cache disabled (which will cause a performance drop-off), if you compile the kernel with 68060 support. The caching problem will be resolved in due course.

Richard has also teamed up with BVM Ltd. to port Linux/m68k to that company's BVME4000 and 6000 boards. The port will be integrated into the development tree in the near future; you can obtain it from *Richard's VME pages* <<http://www.sleepie.demon.co.uk/linuxvme/index.html>>. More information is available on *BVM's home page* <<http://www.bvmltd.co.uk/>>.

Richard is currently working on a port to the Tadpole TP34V, a 68030-based VME single-board computer.

4.8 Macintosh Hardware Support

(written by Joe Pranevich; updated by David Kilzer)

Currently, development on the port is going along nicely. The latest 2.0 kernel is 2.0.33p11, which contains enough drivers for a "production" machine on some systems. The latest 2.1 kernel is 2.1.105, which includes Mac IDE support and some Quadra SCSI support.

Currently “supported” machines are listed on <http://www.mac.linux-m68k.org/>. Most 68030 and 68040 Macs with an FPU will boot to the login prompt using a RAM disk. FPU-less Macs will hang at some point during the boot process.

The following Macintoshes have the “right” processor built-in: all “Classic” Macs (except as noted below), all “II” Macs (the original Mac II requires a 68851 PMMU), all “LC” Macs (except as noted below), all Performas, all Centris and Quadras, and all Powerbooks/Duos (except as noted below). “Classic” Mac exceptions include the original Macintosh (128k), 512k, 512ke, XL, Plus, SE, SE FDHD, and the original Mac Classic (all of which use 68000 processors). Powerbook exceptions include the Portable and the Powerbook 100 (both use a 16 MHz 68HC000). The only “LC” Mac exception is the original 68020-based Mac LC, which lacks a PMMU slot, though a hardware hack and third-party processor upgrades are available. See 4.1 (the CPU listing) for a comprehensive list of supported CPU/MMU/FPU combinations.

As for drivers, we have NCR5380 and NCR53c9[46] SCSI, Mac IDE, NS8390 (Daynaport) Ethernet, NuBus, ADB (Mac-II, IIsi and CUDA styles) for keyboard and mouse (also used by some NeXTs), video (most of Apple’s video boards, except RBV–RAM-based video–boards), and probably some other goodies. We also have a working installer and booter (Penguin).

Updates, as always, on <http://www.mac.linux-m68k.org/>.

4.9 Sun 3 Workstation Hardware Support

(written by Joaquin Menchaca)

There is a current Linux/m68k port to the Sun 3 Series. The machines intended to be supported are Sun 3/50, Sun 3/60, Sun 3/75, Sun 3/150, and others of similar design. The Sun 3/80 and Sun 3/40X have radically different hardware and will thus have to be supported by a different port.

Pekka Pietikäinen <pp@netppl.fi> is the point of contact for this port. A patch relative to 2.0.25 is available from him. More information is available at <http://www.netppl.fi/~pp/sun3/>.

4.10 NeXT Workstation Hardware Support

(written by Joaquin Menchaca)

There is an ongoing movement to port Linux/m68k to the NeXT hardware. Currently the booter is being worked on. Progress is difficult because the lack of documentation available for this platform. Apple is dropping 680x0 support in future OS releases and current technical support does not make this effort easier.

Several individuals have expressed interest in the port to the m68k NeXT computers, including:

- Martin Bähr <mbaehr@iaeste.or.at>
- Sarlo <sarlo@tezcat.com>
- Mike Allison <mallison@konnections.com>
- Cal McInvale <calm@tpdinc.com>

The current status of this porting effort is unclear. More information should be available at the Li/NeXT web site, <http://www.black.linux-m68k.org/> and/or Zach Brown's site, <http://www.zabbo.net/linux/next/>.

4.11 Hewlett-Packard (HP)/Apollo Domain Workstation Hardware Support

The first pre-alpha release of Linux/m68k for Apollo Domain workstations is now available at <ftp://ftp.ba.be/pub/apollo/>. It apparently only runs from a ramdisk, but it does support the monochrome framebuffer, 3c505 Ethernet card, and the keyboard. Contact Peter De Schrijver Peter.DeSchrijver@linux.cc.kuleuven.ac.be for more details.

4.12 HP 9000/300 Workstation Hardware Support

(written by Phil Blundell)

Phil Blundell Philip.Blundell@pobox.com is working on this port. Phil has set up a web page at <http://www.tazenda.demon.co.uk/phil/linux-hp/> with more information about this port.

4.13 Unsupported 680x0 Hardware

There is currently no Linux/m68k port for several 680x0 based computers that would be able to run Linux. The reason for this is rather simple: Nobody has written it. The reasons for that are many: The people who already have most/all of the knowledge on the Linux side of the port are usually busy maintaining/improving one of the existing ports. Another quite common reason is that no or only insufficient documentation on the hardware of that platform exists. Sun 3s are an even more special case: Unlike all other machines mentioned here, they don't use Motorola MMUs (except the Sun 3/80, which uses a 68030). See 5 (Future Development) for details on progress (or lack thereof) toward completing ports to these systems.

As far as I know, you can run *NetBSD* <http://www.netbsd.org/> or *OpenBSD* <http://www.openbsd.org/> on some of the unsupported systems.

4.14 Support for other processors

Linux also runs on several other platforms in varying states of usability:

- Alpha (Linux/AXP): <http://www.azstarnet.com/~axplinux/>
- PowerPC processors (LinuxPPC): <http://www.linuxppc.org/> and <http://linuxppc.cs.nmt.edu/>; for PowerUP cards, see 8.9 (the PowerUP information).
- PowerMacs using OSF Mach microkernel (MkLinux): <http://www.mklinux.apple.com/>
- Sparc processors (SparcLinux): <http://www.geog.ubc.ca/sparclinux.html>
- I think I heard some rumors about an Intel version too ; -)

This list is *not* exhaustive. A definitive guide can be found at http://www.ctv.es/USERS/xose/linux/linux_ports.html.

4.15 Definitely Unsupported Hardware: The Hall of Shame

(inspired by recent traffic on the kernel list)

The following is a list of devices for which the manufacturer either will not provide programming information or will not make that information available under an agreement that allows us to release the source code for the driver.

You are encouraged to voice your displeasure at these manufacturers if you own these products and to refrain from purchasing their products until they adopt a more sensible policy.

- Apple Computer, Inc.: Apple has only released programming information as part of the source code for its MkLinux port of Linux to its PowerPC-based Macintoshes. They have not released any programming information for the 680x0 Macintoshes.
- Interworks (ICard PCMCIA Ethernet card)
- Individual Computers (Catweasel floppy controller): Information on how to program the floppy controller portion of the card is only being released under a license that does not allow us to publish the source code for the driver.

5 Future Development

This is a concise listing of current projects underway for Linux/m68k. Let me know if your project isn't listed so I can add it.

I have marked the ones that haven't had any recent activity with ? (these may revert to unclaimed status if I don't eventually get mail from someone saying that they are being done).

5.1 General

- A port of *the Generalized Graphics Interface* <http://www.ggi-project.org/> (a kernel-level I/O abstraction) to Linux/m68k: Geert Uytterhoeven <Geert.Uytterhoeven@cs.kuleuven.ac.be>. (This project seems to have been abandoned in favor of Geert's *abstract console scheme* <http://www.cs.kuleuven.ac.be/~geert/Console/>.)
- Atari TOS emulation: *The OSIS Project* <http://www.nocrew.pp.se/osis/>.
- AmigaOS emulation: *The Amiga Replacement OS project* <http://www.lysator.liu.se/amiga/aos/>.
- Linux/m68k-specific edition of *Linux Installation and Getting Started*: Chris Lawrence <faq@linux-m68k.org>.

- Rewrite of 68882 FPU emulation under GPL or under a less restrictive BSD-style license (like the MIT/X Consortium license): Roman Zippel <zippel@fh-brandenburg.de>, Rob Pelkey <rpelkey@abacus.bates.edu> and others. A recent mail from Roman claimed that the emulation was getting close to ready.
- **(Unclaimed)** MacOS emulator (based on Shapeshifter, perhaps?)

5.2 Amiga

- Drivers for Arcnet cards are in alpha/beta: Frank Neumann <Frank.Neumann@informatik.uni-oldenburg.de>. Frank is looking for someone else who is willing to adopt the code; current sources can be found at <<ftp://ftp.informatik.uni-oldenburg.de/pub/linux/local/driver/>>.
- ? Support for the A2410 and DMI Resolver graphics boards is planned: Topi Kanerva <topi@susanna oulu.fi> and Jes Degn Sørensen <Jes.Sorensen@cern.ch>.
- ? Support for the Commodore A2232 7-port serial card is under development: Eric van Dijken <E.vanDijken@PTT-Telecom.nl>.
- Support for the Phase5 PowerPC boards: Roman Zippel <zippel@fh-brandenburg.de>, Jes Degn Sørensen <Jes.Sorensen@cern.ch> and Jesper Skov <jskov@cygnus.co.uk> (Phase5 contact person). See the APUS FAQ <<http://sunsite.auc.dk/ftp/pub/os/linux/apus/docs/faq.html>> for the current status of this project.
- Access to ISA cards via GoldenGate/BridgeBoard (as in *OpenBSD* <<http://www.openbsd.org/>>): Arno Griffioen <arno@usn.nl> has done some work in this direction on the GoldenGate II. He currently has some non-DMA devices working using the standard Intel drivers; he is currently looking for testers.
- Support for the Blizzard 1230-I and 1230-II SCSI boards: Jesper Skov <jskov@cygnus.co.uk>. The current test release of this driver is available at <<ftp://sunsite.auc.dk/local/os/linux/680x0/testing/blzII/>>; Jesper is also attempting to get information on the 1230-III board.

The following drivers were asked about (or I thought of them) but nobody has reported interest in writing them:

- Concept and code for the use of chip ram and the blitter from user space.
- Support for the Utilities Unlimited Emplant serial and SCSI.
- BSC Alfa Data MultiFaceCard II support. Docs available from Jörg Dorchain <dorchain@mpi-sb.mpg.de>.
- Improved Phase5 Cybervision support (on-the-fly mode changes, 15/16/24 bits-per-pixel frame-buffer support, 3.2-based XF68_Cyber, new 3D version, 3D MPEG add-on support, etc.). Somewhat happening already.
- Prodev Merlin graphics card.
- Commodore A2090 SCSI controller.
- Commodore Bridgeboards (A2286 in particular).

- Commodore/Ameristar A4066 Ethernet card.
- Commodore A2024 monitor (a high resolution grayscale monitor).
- DraCo (Amiga clone/nonlinear video editor) support.
- Access parallel IDE devices on some Amiga IO cards. My understanding is that both the GVP IOExtender and the Multiface III can support parallel IDE drives, at least in theory.
- Access to ISA devices on the BoXeR motherboard.

5.3 Atari

- The port to the Hades (an Atari clone) is underway and is at least somewhat working: Wout Klaren <W.Klaren@inter.NL.net>.

5.4 VME

- Earlier MVME cards (with 68020 and 68030 processors).

5.5 Other Systems

I am not aware of any current subprojects for other ports that are being developed separate from their own kernel trees. Please let me know if any new projects need to be added.

6 The Kernel

6.1 Recompiling Linux

(Much of this section written by Jesper Skov)

Since the release kernels found at the various Linux/m68k sites are compiled for the average user, they contain drivers for just about every piece of hardware supported by Linux. They will also be compiled so they can be run on all the processors in the Motorola 68000 family.

This scheme enables everybody to boot Linux, but it also reduces the performance of Linux: drivers for hardware you don't have or use take up (non-swappable) memory, and in case of processor specific programming, it is required to check which set of instructions to run.

Therefore, if you are a bit adventurous and have time to spare, you should recompile a kernel for your machine configuration. Especially if you are not a kernel hacker (i.e., you don't care about all those 2-3 line patches :) and use the same kernel for several days/weeks, the time will be well-spent. (Of course, if you are a kernel hacker, the time will be well-spent as well. :)

Recompiling your own kernel will make it possible to configure it to your exact machine configuration, thus giving the best performance possible.

6.1.1 Finding The Sources

The Linux/m68k sources are available by FTP from sunsite.auc.dk in “/local/os/linux/680x0/vx.x”, x.x denoting the latest version numbers.

The sources are also mirrored at the sites listed in 11.6 (FTP Sites).

DO NOT try to use standard Linux kernel source trees (from e.g. `ftp.kernel.org`) to compile Linux/m68k. These trees are often out-of-date and may include serious bugs due to changes being made on some architectures not being propagated to Linux/m68k. Stick to Jes’s source trees (or Jesper’s for Linux/APUS) unless you really know what you are doing.

6.1.2 What You Need to Recompile

You will need at least gcc version 2.7.2 (preferably 2.7.2.1 or later) to compile the kernel.

Recompiling the kernel for the MC68060 processor requires binutils 2.7.0.3 or later, fixing a linking problem with the ifsp (integer/floating point support) files.

Linux 2.1.x and later require binutils 2.7.0.3 (or later) with the CHIP fix. This makes it possible to specify which processor (chip) the instructions should be assembled for. Thus it is now possible to write mnemonics instead of opcodes for the bigger processors, easing the reading of the code and removing the problem of wrong mnemonic/opcode translations.

GCC can be found at any of the Linux/m68k FTP sites (see 11.6 (Linux/m68k ftp sites)). The required version of the binutils package can be found at `<ftp://sunsite.auc.dk/local/os/linux/680x0/bin/>`.

6.1.3 How to Compile

There are three important stages: configuring the kernel; generating the dependencies and compilation of the main kernel (vmlinux); and building modules. You can choose to either use modules (non-essential modules can be demand-loaded by the kernel) or not use modules (which means all of your drivers will be built into the kernel). Using modules makes your kernel and drivers slightly bigger, but if you don’t use several devices most of the time (like your printer port, CD-ROM, and various filesystem formats) you will save system memory when they are not in use (as none of the kernel itself can be swapped to disk).

Note that this is a generic set of instructions; Debian comes with a package called “kernel-package” that automates the compilation and installation steps (and integrates them into the Debian package system; a side benefit of installing kernel-package is that installing it makes you install all of the other packages needed to compile a kernel). No doubt Red Hat has a similar facility. In any event, the general principles discussed here apply no matter how you actually compile the kernel.

Configuration

1. `cd` to your kernel source directory (e.g. `cd /usr/src/linux`)
2. Type `make config`. (If you have the ncurses headers installed on your system, you may want to use `make menuconfig` instead, because it is more forgiving of mistakes, though it is slower. You can

also use `make xconfig` if you have Tcl/Tk and want to configure under X, or `make oldconfig` if you have already configured a kernel and just want to be prompted for new options.)

3. Answer the questions you are asked (or answer all of the questions under each menu, if you're using `menuconfig`). Most questions will be Yes or No questions; if you choose to use modules, you can also answer M to many of the questions to build a module.
4. If you do use modules, you cannot modularize your boot device (i.e. your IDE or SCSI controller) or your root partition's filesystem format (usually `ext2`); I recommend against modularizing the ramdisk support (in case you ever need to boot from a ramdisk for some reason). I also recommend that you answer yes to the questions about `kernel.d` or `kmod` support (and be sure to get a copy of `modutils` and install it on your system).
5. Eventually it will stop asking questions and you will get your command prompt back. (`menuconfig` has a "Save and Exit" option that will get you out when you're done answering questions.)

Compiling the kernel This part is easy. Type `make clean; make dep; make` (all on one line, with semicolons between each command). Then go away while your computer heats your house; the compilation speed depends on (a) how much RAM you have, (b) how fast your CPU is and (c) how fast your hard drive and its controller are. Slow computers (like 68020s) can take over a day to compile a kernel; some 68060 owners have reported compilation times measured in minutes. Some computers (with lots of RAM) will benefit from running the last `make` as a parallel `make` (using the `-j` switch); see the manual page for `make` for details. When it is done, you should have a file called `vmlinux` with the kernel in it.

Compiling the modules Compiling the modules is also easy; type `make modules; make modules_install` and then go home and eat popcorn (you can combine this line with the compilation line above, e.g. chain the 5 makes together on one line). You should end up with a lot of files under `/lib/modules/X.Y.Z` (where X.Y.Z is your kernel version number).

Installing your new kernel Generally, all you should need to do is copy the `vmlinux` file to wherever your bootstrap expects to find it (for `atboot` and `amiboot`, this will be on a native filesystem partition; for LILO it will probably be in `/` or `/boot`, but this can be configured differently). Make sure you keep a copy of your previous `vmlinux` file where you can get to it (or set up LILO so you can boot from the previous file if necessary), just in case something goes wrong.

6.1.4 Submitting Kernel Changes

Linux/m68k releases are built and released by Jes Degn Sørensen. "Built" means that you can write a patch against the current version or patchlevel and send it to Jes and he will integrate it into one of the next releases. Make sure you state against which version the patch was made. Please note that Jes has no way to test Atari-specific patches.

It is also considered good etiquette to send your patch to the Linux/m68k mailing list, so (a) Jes can see if other people say it works (a crude form of peer review, if you will) and (b) so everyone else on the list has a new patch to fool around with (a crude form of everyone getting their kernel-hacking fix ;).

Note: If you patch processor-specific code (e.g. 68030 vs. 68040 MMU or FPU) make sure that you document the dependencies.

6.1.5 Bug Reports

Send bug reports to the author of the code or to Jes. Probably a better approach is to post it to the linux-m68k mailing list or to the appropriate newsgroup. If there are bugs that will probably stay in the code for an extended period of time let me know so I may publish them here.

6.2 Known kernel bugs

6.2.1 TT-FPU bug

Problem:

Linux reports *** COPROCESSOR PROTOCOL VIOLATION *** FORMAT=9 or something similar.

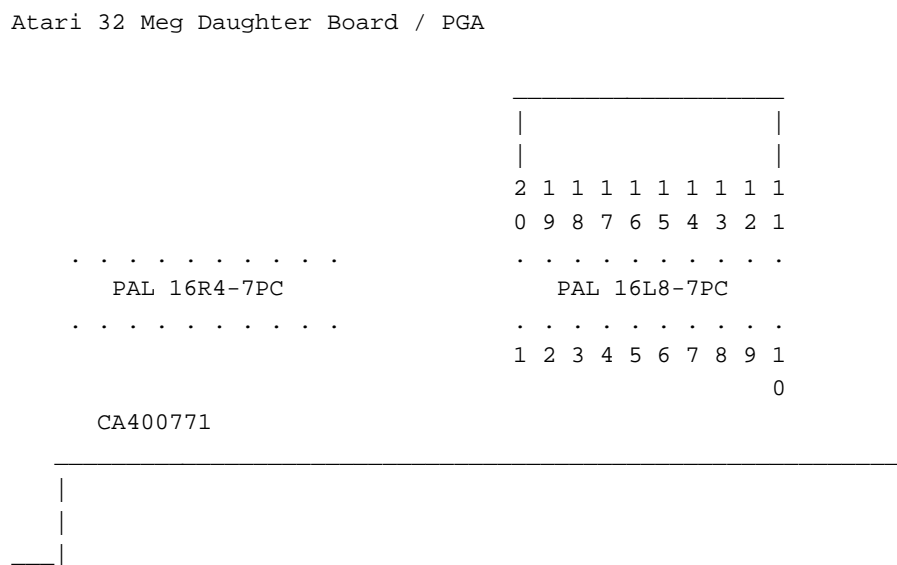
Fix:

Pull the 16L8 PAL's pin 11 free (this is the signal 'XBG') and solder it to +5V. This prevents the PAL from tri-stating AS and DS until XBGACK has gone low.

To make your 32 MHz daughter-card work:

1. Pull pin 11 of the 16L8 PAL out of the board
2. Solder pin 11 of the IC to +5V (pin 20 of 16L8)

```
****This applies to the CA400771 32 MHz daughter board****
*****Other boards should not have this bug*****
```



6.2.2 Amiga with GVP 16-bit RAM

Problem:

When using a GVP card with 16 bit RAM on an Amiga with 68040, Linux/m68k dies. (This also happens with Commodore A2052 boards.)

Fix:

Unfortunately, there is no known solution to that problem. So your best bet is to get some real 32 bit RAM. The 16-bit RAM may still be used as a ramdisk using the z2ram device, see 8.11 (the z2ram device).

When using the 2.0.x (and earlier) kernels, you *must* stop it from being used as normal RAM (-m option of amiboot). Quoting from the README for amiboot:

```
If you have some non-AutoConfig memory you want to use under Linux, or if you
want to disable some parts of your memory (e.g. Zorro II RAM on '040 based
systems), you have to use a memory file and the --memfile option. This file
contains information about the memory chunks you want to use under Linux. The
format for the file is:
```

```
chipramsize
[0xfastchunkaddr fastchunksize]
[0xfastchunkaddr fastchunksize]
...
```

For example, if you don't want Linux to use your 2nd meg of chipram, you would create a file that looks contains only:

```
1048576
```

If you had 1M of chip ram, 2M of 16 bit FAST ram at address 0x200000 and 16M of 32 bit FAST ram at address 0x80000000, and you didn't want Linux to use the slow 16 bit FAST ram, you'd create a file that looks like:

```
1048576
0x80000000 16777216
```

The memory file can also be used to specify in which block of memory the kernel will be put. Normally Amiboot will put the kernel in the first block of Fast RAM it will find. If you use a memory file, it will put the kernel in the first block of fast RAM you specify.

In recent 2.1.x kernels and the 2.2 pre-releases, the 16-bit ram is automatically disabled on Zorro-III systems, so you don't need to make a memory file.

6.2.3 Zorro II DMA Bug on A3640 rev 3.0

Problem:

You get strange system crashes and segmentation faults with a Zorro II SCSI controller and an Amiga 3000 or 4000 with a Commodore A3640 (revisions 3.0 and 3.1).

Fix:

The immediate fix is to make your SCSI controller stop using DMA. If you're using a A2091 or GVP Series II controller, add the kernel parameter `wd33c93=nodma:1` to the boot command you are using. However, this will result in irritatingly slow disk accesses.

A better solution is to replace the A3640 with a newer version of the card (revision 3.2) or a third-party card like the Phase5 Cyberstorm, Warp Engine, GVP/QuikPak 4060D, or Apollo 4060. You may also be able to upgrade your A3640 yourself, if you have some electronics knowledge.

6.2.4 MC68060 performance issues

(written by Jesper Skov)

Problem:

In order to streamline the design of the MC68060, Motorola left out the implementation of a few addressing modes and instructions. The '060 remains user-level compatible with the other family members by emulating these addressing modes and instructions in software.

Whenever the '060 encounters an unimplemented instruction, a special exception is taken that enters the ifpsp (Integer and Floating Point Software Package), which is part of the kernel. Here the instruction is emulated and processing is resumed.

Obviously, this adds an overhead to the use of the system and since gcc 2.7.2 uses the unimplemented instructions for 64 bit multiplication and division there is reason for concern. Judging from my tests (highly inaccurate :) I expect a boost of at least 10% when applications can be recompiled with an 060 aware gcc (that would be release 2.8.0, due out last year ;)

Fix:

Recompile applications with an 060-aware gcc when available. A patch to gcc 2.7.2 was recently posted that may also help.

7 Common problems

A lot of these no longer apply to new users, but may be of interest to people who have been running Linux/m68k for a while.

This is by no means an exhaustive list. If you have any suggestions for entries, please send them to me.

Note that system-specific questions are in separate sections of the FAQ; you should read this section and the one pertaining to your system (if there is one) before assuming your question isn't answered here. Also note that system requirements are covered earlier in the FAQ.

7.1 I can't find the man page for XXX

There is a *wealth* of Linux documentation out there for the original PC Linux which almost always applies to Linux/m68k. Check out the documentation at your favorite Linux FTP site.

Having said that, if you're using a proper distribution (Debian or Red Hat), you should have man pages for all executable files; file a bug report if the pages are missing.

7.2 How can I access my SCSI devices?

To access a SCSI device, you need to know two things: where it is logically located on your SCSI chain and what type of device it is.

Where it is on the chain determines what order it will appear in on the device list. Note that the SCSI ID is what is used to determine location on the chain; this ID will normally be between 0 and 6 (but can be between 0 and 15 if you have an Ultra-Wide SCSI controller).

What type of device it is determines how it is addressed.

Some examples:

- `/dev/sda` is the first SCSI fixed disk (hard drive or removable hard drive) on the chain. `/dev/sdb` is the second SCSI fixed disk. To access partitions on hard drives, you follow the device name with the partition number (e.g. `/dev/sda1` is the first partition on `/dev/sda`); you can access up to 128 drives and up to 15 partitions per drive.
- `/dev/scd0` is the first SCSI CD-ROM on the chain. `/dev/scd1` is the second SCSI CD-ROM. You can have up to 256 SCSI CD-ROM drives.
- `/dev/st0` is the first SCSI tape drive on the chain. `/dev/st1` is the second SCSI tape drive on the chain. You can access up to 32 tape drives.
- `/dev/sg0` is the first miscellaneous (“generic”) SCSI device on the chain (most often this will be a scanner; it can also be a CD writer). `/dev/sg1` is the second generic SCSI device. You can have up to 256 generic devices.

Note that to use an external SCSI device, it must be switched on when you boot the system. Also, it is a bad idea to swap removable fixed disks while the system is switched on (it is OK, however, to swap CD-ROMs and tapes, when they aren’t mounted).

If you have multiple controllers, the device assignments will get horribly confusing (there is a logic to it, but it defied my powers of explanation); I recommend reading the boot messages to determine what device addresses are being assigned to each device.

7.3 How do I access my IDE devices?

Unlike the SCSI driver (which can distinguish between CD-ROMs, tape drives and fixed disks), the address of most devices on the IDE bus (except tape drives) only depends on where it appears to be on the bus. Assuming you have an IDE controller, all you need to know to access it is what “hard drive position” it appears in:

- Master on primary controller: `hda`
- Slave on primary controller: `hdb`
- Master on secondary controller/doubler: `hdc`
- Slave on secondary controller/doubler: `hdd`

You can have more than two controllers: for example, hde and hdf correspond to the master and slave on the tertiary controller. You can currently have up to six IDE controllers on a system (and thus up to 12 drives).

This information should also appear in the boot messages. You can access up to 63 partitions on an IDE fixed disk. Note that the distinction between “primary” and “logical” partitions only applies to disks partitioned using the MS-DOS partitioning scheme.

Parallel IDE devices are currently not supported on any Linux/m68k system; however, the underlying hardware support is available on at least some Amiga IO cards.

7.3.1 How do I use an IDE CD-ROM?

Generally if you have just an IDE hard drive and an IDE CD-ROM, the CD-ROM will be hdb (depending on your master/slave settings).

Simply make a mount point (e.g. `mkdir /cdrom`) and then do `mount -t iso9660 /dev/hd? /cdrom`, replacing the ? with the drive letter from the list above.

7.3.2 How do I use an IDE tape?

IDE tape drives are accessed through `/dev/hdt0` (`/dev/nhdt0` for no-rewind-on-close). Note that only one IDE tape drive is supported.

7.4 My SCSI bus locks up when the kernel probes for devices

(written by Frank Neumann)

Some devices dislike being polled on LUNs (logical units) other than 0. What can happen here is that the SCSI bus just locks up because the device does not answer the inquiry. Quite a couple of drives have already been added to the blacklist of “bad” devices in the kernel, but there are probably more. If you discover this behavior on one of your SCSI devices, you might try adding it to the blacklist (in `drivers/scsi/scsi.c`) yourself or ask someone to do it for you if you are sure about it.

If you think you’re adventurous and want to fix the kernel for a specific SCSI device yourself, here is what you could do. Note that these instructions are for the Amiga, but they apply in general to all systems:

Under AmigaOS, use the `scsiutil` command (available on Aminet) and its `-i` option to send an Inquiry command to that particular device. Write down its vendor identification, product identification and Product revision level. For instance, an Apple CD-300 CD-ROM drive might give (at the bottom) this output:

```
Vendor identification: SONY
Product identification: CD-ROM CDU-8003A
Product revision level: 1.9a
```

Now go into the kernel source tree and (under `drivers/scsi/scsi.c`) add your drive to the blacklist of drives that have problems (just search for “blacklist”). Recompile the kernel and try it out without the `wd33c93=` options you probably used so far.

If it works, you might want to send your change as a unified context diff (use “diff -u”) to the Linux/m68k mailing list.

7.5 I displayed a binary file, and now my console is totally screwed up

(written by Frank Neumann)

Once in a while, it may happen to you that you try to read a binary file. Text viewers like `more` will interpret the input they get as control characters, to for instance change to an alternate character set. This may result in a strange looking prompt, made up of special characters. In such a case, you need to reset the terminal to its initial state. There are several ways to do this, here's what I use: You have to type (blindly):

```
echo ^V^O
```

Read this as: Control-V, Control-O. The sequence "Control-O" does just what we want: It resets the text attributes and character set, and also clears the screen. You have to mask the control character with Control-V, otherwise the shell would directly try to use the "Control-O" for its purposes.

Control-V Esc c is another useful sequence that does a more complete reset of the console (but you usually won't need to use it).

You can also avoid this problem by using `less` or `most` instead of `more`; both of these pagers are available in Debian. Be sure to set up a `PAGER` environment variable (in your `.cshrc` or `.bashrc`) so programs like `man` use your preferred pager instead.

7.6 Can I use both ELF and a.out libraries/binaries in my system?

(written by Frank Neumann and Jesper Skov)

No problem. If you moved to ELF according to Andreas Schwab's hints (`<ftp://tsx-11.mit.edu/pub/linux/680x0/ELF/README>`), you already have a mixed system. All old a.out shared libraries, stubs, static libraries and simple object files (`*.so`, `*.sa`, `*.o`, `*.a`) are now in `/usr/m68k-linuxaout/lib`, except for `libc` and `libm` which remain in `/lib`. The ELF shared libraries are in the usual places (`/lib`, `/usr/lib` and maybe `/usr/X11R6/lib` and `/usr/local/lib`) and don't interfere with the a.out libraries.

When starting a program that is either a.out or ELF, the corresponding link loader (`ld.so` and `ld-linux.so` respectively) will determine what shared libraries are required and load them on the fly. This of course works for both a.out and ELF binaries. You only have to keep in mind that with a mixed system (using some binaries that require ELF libraries along with others that require a.out libraries) both ELF and a.out libraries have to be kept in memory (in particular, but not limited to, `libc` and `libm`). This certainly costs valuable memory. So, the long-term solution will be a pure ELF installation (libraries and binaries).

You can download the ELF versions of the programs via FTP, download sources and recompile, or "Debianize" your system (I really don't recommend the latter unless you really know what you are doing).

If you install any recent distribution, you will have the a.out compatibility libraries but (hopefully) no a.out binaries.

Note: Concerning a.out libraries, a couple of people had problems with the last `libc` that was created (4.7.2). So we recommend staying with 4.6.27 which most people were using.

Note: a.out is effectively dead on Linux/m68k. You can probably live without a.out support (you might want it available as a module, however).

7.7 “less” behaves oddly when I press a key

The older versions of the root-filesys have the wrong device numbers for `/dev/tty`. Delete it and do a `/dev/MAKEDEV std` (you do have a somewhat current `/dev/MAKEDEV`, don't you?)

7.8 What are the current major/minor device numbers for `/dev/xxx`?

The “official” version of MAKEDEV for Linux can be found at <http://metalab.unc.edu/pub/Linux/system/Admin/>. A list of the major/minor device numbers can also be found in the kernel tree as `Documentation/devices.txt`.

Debian/m68k includes a MAKEDEV that is pre-configured for Linux/m68k.

7.9 How can I tell an a.out binary from an ELF one?

Use the `file` command. It will either tell you ‘mc68020 demand paged’ if it is an a.out binary, and give a longer (self-explanatory) description if it is ELF.

You can also use `ldd`. If it says something about a “DLL jump”, the binary is in a.out format; otherwise, it's in ELF.

For statically linked binaries, `ldd` reports `statically linked (ELF)` for ELF binaries. I have no idea what it says for a.out binaries, because I don't have anything statically linked a.out any more (at least, not that I can find).

7.10 GCC complains that it can't find shared libraries while linking

(written by Geert Uytterhoeven)

The current `ld` requires you to create links from `*.so` to `*.so.6` for all libs, so you should have e.g. `libX11.so -> libX11.so.6, libX11.so.6 -> libX11.so.6.0`

The development packages in Debian automatically handle this situation.

7.11 How do I byte-swap an ext2 filesystem?

Please read Michael Schmitz's mini-HOWTO on this topic; it's available at <http://www.linux-m68k.org/ext2swap.html>. The mini-HOWTO also explains how to check if you need to do anything.

7.12 My kernel hangs at the login prompt

It is possible that you have an IDE controller on your system without any IDE devices attached to it. If you have been experiencing problems like this, you should upgrade to kernel 2.0.28. If upgrading doesn't fix the problem, ask for help in the newsgroup.

7.13 I just upgraded to 2.1.21 (or later) and modules don't work

You need modutils-2.1.23 or later. Note that these modutils are incompatible with kernels from 2.1.0 to 2.1.17, so you'll need to keep a copy of modutils-2.1.13 around if you plan on switching back and forth. The sources for modutils can be found at `<ftp://ftp.redhat.com/pub/alphabits/>` or a more reliable mirror, `<ftp://tsx-11.mit.edu/pub/linux/projects/alphabits/>`.

If you're running 2.1.26 or later, no patches should be necessary (beyond the ones you had to apply to get that version to compile).

Modules in 2.1 series kernels also require running libc 5.4.17 or later (which in turn requires ld.so 1.8.5 or later). In any event, you probably should be running glibc by now.

Current versions of modutils are available in Debian that support both 2.0 and recent 2.1/2.2 kernels.

7.14 Where can I get Linux/m68k on CD-ROM?

See 11.8 (Distributions) for an answer to this question.

7.15 How do I patch my kernel?

Once you have a copy of the Linux/m68k kernel, you should rarely need to get a completely new tree. Instead, you can patch the kernel sources to the next released version.

For example, if you have the 2.0.25 kernel tree already, you need to get the file `linux-2.0.28.diff.gz` via FTP (don't get the file with the word "native" in it unless you have the same version kernel tree for Linux/i386). Then use `cd` to get to the directory above your kernel tree (probably `/usr/src`), and make a copy using hard links to save a lot of space:

```
cp -rl linux-2.0.25 linux-2.0.28
```

You may also want to change your symbolic link `linux` `->` `linux-2.0.25` to point to the new tree:

```
ln -sf linux-2.0.28 linux
```

This way, your links in `/usr/include` don't have to be changed every time you upgrade your kernel (i.e. you can link `/usr/include/linux` `->` `/usr/src/linux/include/linux` instead of using the kernel version number hard-coded). Then, `cd` to `linux-2.0.28` and type the following 2 commands:

```
rm -rf include/asm
zcat (path of linux-2.0.28.diff.gz) | patch -p1 -s
```

If all goes well, it will work for a minute or two and then return you to your shell's prompt. Make sure the patch applied correctly by typing `find . -name '*.rej'`. If no filenames are listed, everything worked perfectly.

Now do a `make clean` to delete all the backup `.orig` files left by patch, and then do a normal `make config`, `make dep` and finally `make`.

Once you've successfully made a copy of the new kernel, you can safely delete the previous version's tree using `rm -r`.

Please note that you will often get rejected patches if you patch a kernel tree that you have already patched from the Linux/m68k mailing list (or from any other source). The easiest way to avoid this is to make a "clean" (i.e. distribution) kernel tree and another tree that you apply the patches from the mailing list to.

Miniproject: Someone with a bit of spare time might want to adapt `scripts/patch-kernel` to understand Linux/m68k diffs. Your fellow Linux/m68k users would be eternally grateful.

7.16 How do I patch a generic Linux kernel tree to work with Linux/m68k?

This is basically the same as patching a Linux/m68k kernel tree. At `sunsite.auc.dk`, Jes maintains two sets of diffs for each kernel version: they are named `X.Y.Z.diff.gz` and `X.Y.Z-native.gz`, where X, Y and Z are the components of the kernel version number.

Most users will simply use the standard patches to patch a previous version of Linux/m68k's tree to the current version (this is the procedure outlined above). If you already have a standard Linux source tree, however, it is easier to use the `-native` patches.

To do this you *must* have the exact, pristine kernel tree released by Linus for that version; i.e. to make Linux/m68k 2.1.127 you need Linux 2.1.127. You may need to patch your standard Linux source tree using Linus's patches to get it to a version that corresponds with a Linux/m68k release (not every kernel released by Linus is released by Jes).

Once you have identical version numbers, you should `cd` into your kernel source tree and type the following two commands:

```
rm -rf include/asm
zcat (path of linux-X.Y.Z-native.diff.gz) | patch -p1 -s
```

Any errors will be reported on your screen. There may be errors patching the Makefile for some versions (because the SMP setting used to be in there), but these can usually be ignored safely (check the contents of the reject files). You should also make sure that the ARCH setting in the Makefile has not been overridden.

7.17 How can I get my Jaz drive to work with Linux (or another OS...)?

Recent models of the Jaz drive (with firmware version H.71 and later) are shipped in a so-called "write-verify" mode. This mode causes problems for operating systems (like Linux) that can't run Iomega's proprietary drivers.

I (Chris) have been in touch with Iomega tech support, and they provided me with the low-level commands that allegedly get the Jaz drive into the correct mode. However, the commands they provided don't seem to work (probably (ab)user error, knowing my level of SCSI programming knowledge).

In the meantime, the only thing to do is find a Mac or PC with a SCSI interface and use Iomega's Jaz Tools software to disable the write verify mode. This can also be done with a registered copy of Shapeshifter on the Amiga. On a PC, you will need to use the Windows version of Jaz Tools to disable write-verify (the DOS version of the tools won't do it).

More information about the Jaz drive and Linux (including the Jaz mini-HOWTO) can be found at *Bob Willmot's Jaz-Linux information page* <<http://www.cnct.com/~bwillmot/jaztool/>>.

7.18 When I use `dpkg` or `tar`, I get messages about a “broken pipe”

This usually means that you are running a buggy version of `getty`. The `agetty` and `mingetty` versions of `getty` are known not to have this problem, so replace your current `getty` with one of these. The `getty` version included in Debian/m68k will also work correctly.

[It may not actually be a bug in `getty`. But replacing it seems to fix the problem.]

7.19 What is the current status of FPU emulation?

Older kernels with FPU emulation are available from <<ftp://ftp.nocrew.org/pub/linux/>>. The FPU emulation code used in those kernels was adapted from NetBSD, and has bugs originating from both the emulation code itself and the Linux interface to it. The emulation code does not support all transcendent functions and not all supported functions have full precision implemented. Use of this FPU emulation code is not supported, endorsed, or sanctioned by the kernel developers; please do not send them bug reports, complaints, or even patches to this code.

The reason why the old FPU emulation code is unsupported is that its license is incompatible with the GNU General Public License. The code is restricted under the terms of the Berkeley Source Distribution license, which requires that accompanying code and documentation recognize the contributions of the University of California to the software. While this may seem to be a trivial point, the GPL does not allow any restrictions beyond those in the GPL on how software linked to the GPL can be distributed.

A new effort to write an improved FPU emulator under a GPL-friendly license is in the works; it is planned that this emulator will be licensed so it can be used without any strings attached (probably under something like the MIT X license). If you are interested in helping with the coding (no testing yet), join the discussion on the Linux/m68k mailing list. When the FPU emulator is ready for testing, announcements will be made on the mailing list and in the newsgroups.

If you are interested in getting your hands dirty and doing some actual work on the emulator, the code is at the Linux/m68k CVS repository. The CVS server is `cvs.linux-m68k.org`; login with user name “anon” and a blank password.

7.20 I can't boot from a ramdisk

This is usually because of the `initrd` driver. Make sure you specify on the boot line “`-r <ramdisk-name> root=/dev/ram`”.

The other possibility is that you don't have enough RAM to boot from a RAM disk (you really need at least 6 MBs with recent kernels). In this case, it is possible to write the ramdisk's contents to a high density disk and try booting from this (but, of course, you do need a high density drive to do this). Ask in the newsgroup for help if you don't know how to do this on your own. Alternatively you may be able to get someone else to compile a smaller kernel image specifically for your system (which will save a lot of RAM).

7.21 I can't execute programs in my current directory

As a security precaution, most systems come pre-configured to not include the current directory in your path. The lazy way out is to add "." to your path, but I strongly recommend against doing that (particularly if you are running as root).

The right way to handle this situation is to preface the program name with "./". For example, type "./configure" instead of "configure".

7.22 Video Questions

7.22.1 How do I choose what video mode to use with Linux?

This is done with the "video" boot parameter. For details on what resolutions are supported, you'll want to read `Documentation/m68k/framebuffer.txt` (in 2.1/2.2, `Documentation/fb/framebuffer.txt`) and `Documentation/m68k/kernel-options.txt` in the Linux/m68k kernel source tree (the former document only appeared in 2.0.28).

You can also specify what font you want to use with the `font` option to 'video'.

For example, to boot into 640x480x4 mode on an Amiga with a 31 kHz or multiscan monitor, use `video=vga`.

I use `video=font:PEARL8x8,vga` on my Amiga 4000, for a very nice 80x60 text display.

7.22.2 When I run X, it complains about invalid modes

The easiest way to fix this is to change the "Modes" line in your `/etc/XF86Config` to read:

```
Modes "default"
```

Later on, you can get Geert Uytterhoeven's `fbset` program (from `<ftp://ftp.uni-erlangen.de/pub/Linux/680x0/bin/system/>` or the Debian distribution) and customize your video modes.

If you have an old `XF86Config` laying around, you may want to copy the example provided at `/usr/X11R6/lib/X11/XF86Config.eg` into `/etc/XF86Config` and then edit the new configuration file.

7.22.3 Ok, I've got fbset, but I can't create any video modes

(written by Geert Uytterhoeven)

If you have a PAL-based Amiga with a true Multisync monitor (like the A1960), try these as starters (they should be typed on one line; they are split up to make the FAQ format correctly):

```
ModeLine "vga70" 28.376 640 736 848 912 400 412 414 449 +VSync +CSync
```

```
ModeLine "vga" 28.376 640 736 848 912 480 489 491 521
```

```
ModeLine "832x624" 28.376 832 940 1020 1024 624 628 630 660 Interlace
```



```
+HSync +VSync
```

```
ModeLine "832x600" 28.376 832 964 1044 1096 600 600 602 636 Interlace
+HSync -VSync
```

```
ModeLine "896x672" 28.376 896 1044 1108 1160 672 676 678 708 Interlace
-HSync +VSync -CSync
```

```
ModeLine "960x720" 28.376 960 1132 1172 1224 720 720 722 754 Interlace
-HSync -VSync -CSync
```

```
ModeLine "1024x768" 28.376 1024 1196 1212 1288 768 772 774 804
Interlace -HSync +VSync -CSync
```

For NTSC, you can try replacing 28.376 with 28.636 (but this hasn't been tested). Note that each ModeLine should be entered on one line in your XF86Config file.

7.22.4 How do I create the framebuffer device nodes?

(written by Geert Uytterhoeven)

You can create the frame buffer special device nodes for the Amiga using the following commands:

```
mknod /dev/fb0 c 29 0
mknod /dev/fb0current c 29 0
mknod /dev/fb0autodetect c 29 1
mknod /dev/fb0ntsc c 29 2
mknod /dev/fb0ntsc-lace c 29 3
mknod /dev/fb0pal c 29 4
mknod /dev/fb0pal-lace c 29 5
mknod /dev/fb0multiscan c 29 6
mknod /dev/fb0multiscan-lace c 29 7
mknod /dev/fb0a2024-10 c 29 8
mknod /dev/fb0a2024-15 c 29 9
mknod /dev/fb0euro36 c 29 0
mknod /dev/fb0euro36-lace c 29 11
mknod /dev/fb0euro72 c 29 12
mknod /dev/fb0euro72-lace c 29 13
mknod /dev/fb0super72 c 29 14
mknod /dev/fb0super72-lace c 29 15
mknod /dev/fb0dblntsc c 29 16
mknod /dev/fb0dblntsc-ff c 29 17
mknod /dev/fb0dblntsc-lace c 29 18
mknod /dev/fb0dblpal c 29 19
mknod /dev/fb0dblpal-ff c 29 20
mknod /dev/fb0dblpal-lace c 29 21
mknod /dev/fb0vga c 29 22
mknod /dev/fb0vga70 c 29 23
mknod /dev/fb0user0 c 29 24
mknod /dev/fb0user1 c 29 25
```

```
mknod /dev/fb0user2 c 29 26
mknod /dev/fb0user3 c 29 27
mknod /dev/fb0user4 c 29 28
mknod /dev/fb0user5 c 29 29
mknod /dev/fb0user6 c 29 30
mknod /dev/fb0user7 c 29 31
```

However, these should be pre-created when you install your distribution.

7.22.5 How do I make sure that I do not damage my monitor when running X?

(written by Haidinger Walter)

Before trying to run X, you should edit `/etc/XF86Config`. Look for the `Monitor` section. Set the “`HorizSync`” and “`VertRefresh`” values to the specifications matching your monitor.

That is, e.g. for the M1438S *only*:

```
HorizSync      15-40
VertRefresh    45-90
```

Do a “`man XF86Config`” for a detailed description.

Of course, this provides neither a warranty nor an insurance that you cannot damage your monitor, but it will be much more difficult now...

(Maintainer’s note: I fried my monitor once by blindly copying someone’s `HorizSync` and `VertRefresh` values. Make certain that you are using the correct values for *YOUR* monitor; they should be in the book that came with it. And if your monitor starts acting funny, *switch it off immediately!*: if I had switched mine off, I could have saved myself a \$100 US repair bill.)

7.23 When I try to run the X Window System on 2.0.33, I get an error

There is a minor bug in the released 2.0.33 source tree that breaks framebuffer access. There is a patch at *James Troup’s Linux/m68k pages* <<http://thor.lib.chalmers.se/linuxm68k/>>; it is also available at *the Linux/m68k FTP mirrors* <<ftp://tsx.mit.edu/pub/linux/680x0/kernel-patches/2.0.33/geert-fbmem.diff>>.

This problem was fixed in the 2.0.33pl1 release.

7.24 Kernel 2.1.X doesn’t compile out of the box for me

This is usually because Jes doesn’t have the time to test every single configuration before releasing the kernel sources. Monitor the Linux/m68k kernel mailing list for patches (usually all the major fixes have been posted after about two days).

In addition, James Troup maintains a set of pages with the latest patches and release announcements at <<http://thor.lib.chalmers.se/linuxm68k/>>.

7.25 I get strange crashes with kernel 2.1.X and two IDE drives

(written by Geert Uytterhoeven)

This bug apparently appeared in kernel 2.1.15. Geert says:

Gadi Oxman (the IDE guru) told me this could be due to buggy IDE interfaces, IDE drives or both (or a bug in the driver, of course :-), and he suggested to try "ide0=noautotune" which solved my problem last Thursday. But I can't be 100% sure this really solved it...

More recent work has apparently resolved a lot of the IDE bugs people were seeing; try kernel 2.1.119 or later and see if that will fix the problem.

7.26 Internationalization questions

7.26.1 How do I set a xxx keymap? (xxx = German, French, Swedish, ...)

(written by Christian Steigies)

You will need the "loadkeys" command, it is part of the kbd package.

kbd is maintained by Andries E. Brouwer <aeb@cwi.nl> and you can find it at <ftp://ftp.funet.fi/pub/Linux/PEOPLE/Linus/> and <ftp://ftp.win.tue.nl/pub/linux/util/>. The current version is 0.93.

On the SuSE aktuell (March 1997) an older version is on CD 2 as /kernel/kbd-0.92.tar.gz

Install kbd (gunzip and untar the archive, preferably in /usr/local/src, then "make") and "make install". If you use version 0.92 (or lower?) you need to remove resizeicons from the Makefile). You will then have all the executables in the right place and a whole lot of keymaps in /usr/lib/kbd/keytables/

As of version 0.93 these keytables are more or less useless for m68k users, since they are for PC-style keyboards. Loading one of these on an Amiga or Atari screws up the keyboard layout so that its virtually unusable. Later versions include some Amiga and Atari keymaps you can work from.

You need to make one by yourself or get one from <ftp://ftp.uni-erlangen.de/pub/Linux/680x0/bin/system/keymaps/>, where Roman Hodek has started collecting Amiga and Atari keymaps. Get the keymap you want to install, say de-amiga.map, and put it in /usr/lib/kbd/keytables/ (In newer kbd versions, hopefully m68k keymaps will also be included.)

Typing "loadkeys de-amiga" will then load this keymap. To load this keymap during boot, create a rc.loadkeys in /etc/rc.d like (don't include the lines starting with --):

```
--- rc.loadkeys ---
#! /bin/sh
#
# rc.loadkeys      load German keyboard map
#
# Version:         @(#) /etc/rc.d/rc.loadkeys
#

loadkeys de-amiga
--- rc.loadkeys ---
```

Debian/m68k has a more automated way of handling the keymaps (through the kbd Debian package). The package includes several Amiga and Atari keymaps for various layouts.

7.26.2 How do I create a xxx keymap?

(written by Christian Steigies)

The “showkey” command (part of kbd) will tell you which scancode is generated for every key you press, just write down what you want to be generated by this key ; -)

The easiest way is to get a keymap for your computer and only change the keys you want to be mapped differently. “dumpkeys” shows you the current keyboard mapping, “dumpkeys -l” will show you all the names of the symbols that can be mapped to the keyboard. “man keytables” tells more about creating keytables.

When you have made a keymap, contact Roman Hodek <rnhodek@informatik.uni-erlangen.de> so you can upload it.

7.26.3 How do I set up my shell to use non-ASCII characters?

(written by Christian Steigies and Haidinger Walter)

To type umlauts and more in bash, create an .inputrc in your home dir with:

```
--- .inputrc ---
set meta-flag On
set convert-meta Off
set output-meta On
--- .inputrc ---
```

Within tcsh, you need to use the following procedure:

You need an 8-bit clean tcsh with nls support along with the locale package.

In your tcsh, type “echo \$version”. It should say something like tcsh 6.07.02 (Astron) 1996-10-27 (m68k-unknown-linux) options 8b,nls,dl,= al

Only the options are important. It should show at least “8b” and “nls”. If not, you need to recompile tcsh, but, AFAIK, the tcsh binary on the Linux/m68k mirrors has both options set.

7.26.4 How do I use locales?

(written by Haidinger Walter)

To use locale, you need at least libc-5.4.17. I’d recommend to install the lib/libc-5.4.23.bin.tar.gz package if you haven’t already. 5.4.23 is a bug-fix release to 5.4.17 and does not introduce new features.

Next, you need the locale database. I know two sources:

- SuSE owners can just unpack suse/a1/localedb.tgz (usually on CD1) to /

- The package `lib/glibc-2.0.2-m68k-linux.bin.tar.gz` from the Linux/m68k mirror also contains a `_partial_` locale database. To unpack just the database, type on one line:

```
tar -zxvf glibc-2.0.2-m68k-linux.bin.tar.gz
"usr/share/locale/*" "usr/share/i18n/*"
```

Whether you have the SuSE distribution or not, I strongly recommend that you read the mini-HOWTO 'Locales' at least once. It not only describes how to get, build and install the locale package, but also the system requirements and, most important, the usage of the associated commands and environment variables!

So, just follow the instructions of the 'Locales' mini-HOWTO to setup the locale-package and customize it to your system. Finally, set your environment variables to the appropriate values and put them into your `.tcshrc`.

If you cannot wait that long, have a look at `/usr/share/i18n/locales`. If you're German, try (in `tcsh`):

```
setenv LC_ALL de_DE
setenv LC_LANG de_DE
```

bash users would use:

```
LC_ALL=de_DE
LC_LANG=de_DE
export LC_ALL LC_LANG
```

7.26.5 How do I use my keymap in X?

(thanks to Boris Bojic for mentioning this on linux-apus)

You need to disable the X Keyboard Extension; add the following line to your `XF86Config` file:

```
XkbDisable
```

7.27 I installed glibc and now I get errors about undefined references.

(written by Haidinger Walter)

After installing glibc, run `ldconfig`. Then type `ls -l /usr/lib/libc.so`. Is it a symlink to `libc.so.6`? Well, this is not correct for glibc.

Type:

```
rm /usr/lib/libc.so
echo "GROUP ( libc.so.6 ld.so.1 libc.a )" > /usr/lib/libc.so
```

Note that if you run `ldconfig` now, it may issue a warning about `libc.so` not being a shared library. That is Ok, ignore the warning.

7.28 ps and top do not display the associated tty numbers, just a '?'.

(written by Haidinger Walter)

You are missing the /etc/psdevtab file. Type "touch /etc/psdevtab" (as root) and run ps again.

7.29 "make menuconfig" does not work!

(written by Haidinger Walter)

It says something about an old/bad ncurses library/header but I have the latest lib/ncurses-1.9.9e.bin.tar.gz from the Linux/m68k mirror!

7.29.1 For Watchtower

Get the source of ncurses-4.1.tar.gz. It can be found at <ftp://ftp.fu-berlin.de/pub/unix/gnu/>. Compile it. You should not have any troubles following the descriptions in the INSTALL file. Be sure to install the header files too.

Next, you can type "ln -sf /usr/include /usr/include/ncurses" since some packages look for the ncurses headers in "/usr/include" while others look in "/usr/include/ncurses".

7.29.2 For Debian

Get and install the current version of the ncurses3.4-dev package (or ncurses4-dev, if you're using 2.1). If you're using apt (which you probably should be), use:

```
apt-get -y install ncurses4-dev
```

7.30 How do I use the loop-back device?

(written by Haidinger Walter)

Do a "man 8 mount" and search for a section entitled "THE LOOP DEVICE".

Create the device-nodes if they do not exist yet:

```
mknod /dev/loop0 b 7 0
mknod /dev/loop1 b 7 1
...
mknod /dev/loop9 b 7 9
```

More information and examples can be found in:

- CD-Writing mini-HOWTO (which also explains how to mount cdrom-images)
- Documentation/ramdisk.txt (of the kernel sources)
- Bootdisk-HOWTO

Note: To use loop devices, you must have a kernel that supports them.

7.31 Floppies and modules

(written by Haidinger Walter)

7.31.1 I have compiled floppy support as a module but it doesn't work

Assuming that you have read "Documentation/modules.txt" in the kernel sources and you have already installed the correct version of modutils, you should type "modprobe -c | grep -w major-2"

You should get:

```
alias block-major-2 amiflop
```

If it shows 'floppy' instead of 'amiflop', then the kernel searches for a module named 'floppy', which does not exist for Linux/m68k. You have to add the above line to your /etc/conf.modules to assign the proper name. This is also true for some other modules. See the next question for details.

Note: Atari users will want to replace 'amiflop' with 'ataflop'.

7.31.2 Which aliases do I put to /etc/conf.modules?

The kernel looks for the modules under /lib/modules/. The modules are referenced by name. However, some of the m68k-specific modules have different names from their Intel counterparts. Here is an (incomplete) list:

```
# file /etc/conf.modules
alias eth0 ariadne # depends on your Ethernet card (or off)
alias block-major-2 amiflop
alias char-major-4 amiga_ser
alias char-major-6 lp_intern
alias char-major-14 dmasound
alias net-pf-3 off      # no ax25 module available (yet)
alias net-pf-4 off      # if you don't use the ipx module
alias net-pf-5 off      # if you don't use the appletalk module
```

If you have any alias that are missing here, please mail!

Some of these settings may be done automatically by recent versions of modutils.

7.32 Why is there so little information on the web about the BrandX port?

If BrandX is the Amiga or Atari, it's because for the most part we've found that once you get past installation, running Linux on both systems is pretty much the same. There are a few machine-dependencies that affect day-to-day use (port assignments and keymaps, for example), but 99% of the time it makes no practical difference what system you run Linux on.

7.33 How do I use my mouse in X or with GPM?

The configuration is fairly easy. Tell the application you are using a bus mouse, and make a soft link from `/dev/amigamouse` (for Amigas) or `/dev/atarimouse` (for Ataris) to `/dev/mouse`.

You may also want to enable 3-button emulation if you only have a two-button mouse.

7.34 My English isn't very good. Can I read this FAQ in my native language?

There is a French translation of this FAQ available at <http://www.mygale.org/~atari/Linux68k/Faq/>, translated by Christian Jacolot.

If you're interested in translating the FAQ into another language, please let me know and I'll try to point you in the right direction.

7.35 How do I uncompress the pre-compiled kernels outside of Linux?

You will need a program that can extract gzip-compressed tar archives for your system. Mac users can use recent versions of StuffIt Expander; Windows users should be able to use recent versions of WinZip. There is a program called "Opener" for NeXTs that will also work (which actually calls gzip and tar to work).

There is an Amiga program called "UnTGZ" available from Vapor Software (<http://www.vapor.com/software/>); I have no idea what the license is for this program (and haven't used it).

Alternatively, you can obtain the GNU gzip and tar programs for your current OS. On Aminet (for Amiga users), you should be able to find these in `devel/ade`, `util/gnu` or `util/arc`. Older copies of these tools are in the "tools" directory at Erlangen and its mirrors.

7.36 Where's Netscape for Linux/m68k?

The Mozilla browser apparently works (at least to a certain extent) with the GNU Lesstif library; this is probably your best bet for a Netscape-like browser on Linux/m68k (after all, Mozilla is a stripped-down version of what will be Netscape Communicator 5.0). See <http://www.mozilla.org/>. The current development version of Debian includes a Mozilla package.

Note that there is a LinuxPPC version of Netscape available (it's actually called the "MkLinux" version), which apparently works with Linux/APUS.

7.37 dpkg problems with 2.1/2.2 kernels

To resolve these problems, get the latest releases of `libc6` and `dpkg` from the slink distribution. Patching your kernel is unnecessary.

7.38 Can I install Debian (or Red Hat) over an existing Watchtower installation?

(Thanks to Michael Schmitz for pointing me in the right direction for answering this question)

You *can*; however, I strongly recommend against doing so. There are various reasons: two of the most important are varying file locations and C library conflicts. I speak from experience that you *don't* want to mix files (I'm still cleaning up old `a.out` and `libc5` files from my system).

Here's a procedure that should work well for doing an installation from scratch, while keeping your old user files available. This is not well-tested or anything, but should help you get the idea of what you need to do:

1. Print out your `/etc/passwd` and `/etc/group` files (you may need them later). You should also print out your network configuration information.
2. Back up your Linux partitions; if you are installing on a clean disk (i.e. not the one you're using now), you can forgo this and the next two steps.
3. If you have a home partition with all of your user directories on it, keep it around. Otherwise, make a tar file of your home directories tree and copy that somewhere safe (a non-Linux partition should be fine); make sure you use the `"-p"` flag with tar to retain all of the permissions.
4. Repartition (if you want). But don't clobber home if you already had one.
5. Install the base distribution (Debian, Red Hat, whatever).
6. Add the actual user accounts from your old `/etc/passwd` file to your new system, using your distribution's user adding utility (for Debian, it's `useradd`). Set the passwords to each account (or disable them). If you had any special groups (besides a `users` or `staff` group), you may want to add them as well.
7. Mount your home tree at `/home` (if it was on a partition). If you have a tar file of the home tree, figure out where you're going to put it (`/var/home` might be a good choice, although a separate partition would be preferable), untar the home tree there (again using the `"-p"` flag), and make a symlink (if necessary) from `/home` to the root of your new home tree.
8. If you didn't use the tar file: `cd` to `/home` and change the ownership of the users' files to their new users. For example, for a user named bob (in a group named users), `chmod -R bob.users /home/bob`. If people have interspersed files, you may need to do a `find` operation to get the permissions straight (refer to your old password and group files if necessary).

This outline should at least point you in the right direction; let me know if you have any suggestions for improving these instructions.

7.39 What's the difference between glibc and libc6?

`libc` is the C library; basically, it contains all of the system functions that most (if not all) programs need to run on Linux. It's similar to a combination of `dos.library` and `exec.library` on Amigas, but it also contains a lot of things that are in the C runtime library (like, for example, `ixemul.library` or the `.lib` files included with SAS/C and other compilers).

`libc6` and `glibc` are the same version of `libc`; officially, it's version 2 of the GNU C Library (but it's the sixth major version of the Linux C library). You can read more about `glibc` at the *GNU C Library* <<http://www.gnu.org/software/libc/libc.html>> pages.

The major versions of `libc` for Linux/m68k are:

- *libc4*: Version 4 of the C library is based on the a.out binary format; it was the first version to support dynamic linking (shared libraries). However, a.out dynamic linking had a lot of problems (for example, you had to build the library twice, so you could add a jump table to the library on the second pass, and the library was non-relocatable, so every library had to be allocated a block of space to load into), so it was abandoned (at least on m68k; Intel users may still need it for some esoteric applications). *libc4* is abandoned on m68k. You should not be using *libc4* for anything any more. If you do use it, we will hunt you down and execute you as an example to others. (Not really, but you get the point...)
- *libc5*: Version 5 of the C library was a fairly big improvement over version 4. However, it still had some problems (adding new functions or changing structure sizes introduced subtle bugs) so it is no longer being actively developed. It was the first version of the Linux C Library based on ELF, a different file format that made programs loadable in more flexible ways (it uses hunks, similar to the AmigaOS executable file format). *libc5* is officially deprecated on m68k; use *libc6* for new compilations.
- *libc6*: Version 6 of the Linux C Library is version 2 of the GNU C Library; the confusion is because Linux has had multiple C library versions. This is the newest technology available, and includes features (like “weak symbols”) for allowing new functions and modified structures without breaking existing code using version 6, and avoiding kernel version dependency problems. You should be coding and compiling all code against this version.

7.40 How do I install the X Window System?

This procedure depends on the distribution you are using. For Debian/m68k 2.1 (slink), you will need *at least* the following packages (get the latest versions available):

- xfonts-base
- xfree86-common
- xlib6g
- xserver-common
- xserver-fbdev

For a fully functioning X installation, you’ll probably want a decent window manager (fvwm2, fvwm95, AfterStep, WindowMaker, icewm...), some more X packages (like xcontrib) and some more fonts.

7.41 How can I save the kernel messages when the system crashes?

The Linux/m68k kernel supports a “debug” option on the command line. There are several options for this on the Amiga and on Ataris:

- `debug=ser`: Send debug output to the first serial device (usually the internal one on an Amiga; Modem1 on most Ataris; Modem2 on the Falcon). Your terminal should be set for 9600 bps, 8 data bits, 1 stop bit, no parity.
- `debug=mem`: On Amigas (only), save the kernel output to a reserved block of memory. This output can be read after a reboot by “`dmesg`,” which is available at `<ftp://tsx-11.mit.edu/pub/linux/680x0/tools/amiga/dmesg.gz>`.

- debug=midi: On Ataris (only), send output to the MIDI port (31250 bps, 8N1).
- debug=par: On Ataris (only), send output to the printer port. Probably only useful with a line printer (i.e. dot matrix or daisy wheel).

7.42 Where can I get an MMU or FPU for my computer?

If your computer doesn't have a socket for an MMU or FPU already, you will probably need an accelerator board that includes such a socket, or you will need to replace your current chip with a chip that includes the missing feature (i.e. replace a MC68LC040 with an MC68040).

If you do have a socket (usually for an FPU, although some 68020-based computers will have sockets for both an MMU and FPU), you usually need to get an MMU or FPU that is rated at the same clock speed as your main processor (some boards may allow a different speed if they have multiple clocks).

If you have a 68020 and need an MMU, the 68551 MMU is the only choice. If you have a 68020 or 68030 and need an FPU, there are two choices: the 68881 and 68882; the 68882 has more FPU instructions on-board and is of a newer design, but it will be more expensive.

Most Amiga and Atari mail-order dealers will carry these chips; since these are standard Motorola chips, you don't need to buy them from a dealer for your system. In the U.S., *Paxtron* <<http://www.paxtron.com/>> is generally acknowledged to be the best source for new chips (particularly if you need Amiga custom chips); you may be able to find used ("pulled") chips elsewhere, including at online auctions and the like. At one time, *Amiga International* <<http://www.amiga.de/>> had some surplus 68882 FPU chips for sale as well.

8 Amiga-specific questions

8.1 Where did all my Amiga's chip memory go?

It's still there, but the kernel doesn't offer it to the user. It is used by drivers that use the custom chips (like floppy, framebuffer, and sound) and for saving the kernel log (with the debug option documented earlier). The z2ram driver can use this memory, but this option doesn't always work right (as amifb, SCSI controllers and the sound driver may want to allocate chip memory for various purposes).

8.2 How do I access Amiga files from Linux (and vice versa)?

There is an ext2 filesystem for AmigaOS (available at <<http://titan.cs.bonn.edu/~fasten/>>). It allows you to read/write ext2 partitions. Not all write operations are currently supported. *Important:* Make sure you read the "Known bugs" section.

Other ways to transfer files from Linux to AmigaOS are to use an msdos partition, an Amiga/PC formatted floppy with msdos file system (this requires a Mountlist entry on AmigaOS side), use of a partition with Minix file system and the Minix file handler on AmigaOS side (the driver is somewhat unstable) or by accessing the floppy or any (empty!) partition directly via GNU tar.

You can also read and write Amiga partitions from within Linux (using the affs filesystem). Some older install guides say that affs is read-only, but that restriction was lifted in the 2.0 series of kernels (only Directory Cache disks are read-only in 2.x).

AmigaOS text files are normally formatted using Latin1 text; Linux normally defaults to using Latin1 encoding (also called ISO 8859-1), so you shouldn't have any problems. CR/LF problems should not appear either.

8.3 My SCSI bus locks up when I want to use my DAT drive

(written by Frank Neumann)

This problem seems to be related to certain A3000 Amigas: probably only those with BootROMs. It has been discovered that if you have a DAT drive connected whose SCSI address is smaller than the smallest SCSI address of a hard disk in your Amiga, the bus will lock up very early (under AmigaOS, while the SCSI bus is being scanned: you can notice this by seeing that the SCSI LED is constantly lit and nothing happens). The solution is to use higher address for DAT drives (and maybe also for CD-ROMs) and the lower ones for direct-access media (read that as "hard disks").

8.4 Linux recognizes my Amiga's XXX board, but it doesn't work

Linux/m68k can detect the Amiga's AutoConfig devices. That it is able to detect and correctly classify these devices does not mean that the kernel has an actual driver for that device. The list of supported hardware given earlier should be exhaustive, i.e. anything that is not listed is not supported. Note: turbo boards that appear as AutoConfig devices are almost always supported (except for the SCSI controller, if they house one).

If an AutoConfig device isn't properly detected, contact Geert Uytterhoeven <Geert.Uytterhoeven@cs.kuleuven.ac.be> for details on how to make sure it will be properly detected in future kernels.

Note that kernels after about 2.1.105 require the use of a separate "zorroutils" package to get human-readable information about your Zorro devices. This package should be available at Erlangen already, and is available as a Debian package in the unstable ("potato") distribution.

8.5 Amiboot dies when I start it with VMM running

(written by Martin Apel)

What happens is that amiboot gets loaded into virtual memory and shoots itself out of accessible memory when disabling the MMU. But fortunately there's an easy way to solve this:

Enter amiboot into the task list of VMM with code and data set to "No VM". Then amiboot (version 3.0 or higher) should work correctly.

8.6 How do I compile Amiboot and Amiga LILO?

(written by Geert Uytterhoeven)

To compile these programs, you'll need the Amiga Developer's Environment (the GNU tools for creating AmigaOS programs). Linux/m68k binaries of ADE are available at `<ftp://ftp.uni-erlangen.de/pub/Linux/680x0/bin/devel/ADE.tar.gz>`. You will also want the `ADE.readme` file from the same location.

8.7 How do I access floppies under Linux?

(written by Haidinger Walter)

By mounting. First you need a mount-point, i.e. a directory under which you can access your floppy. You can chose an arbitrary name, I use `/df0` through `/df3` (and `/pc0` to `/pc3` respectively) because I'm used to these names from AmigaOS.

Create the directories. e.g.:

```
mkdir /df0
mkdir /any-name-will-do
mkdir /ados/df0
mkdir /ados/pc0
...
```

Next, check if you have the proper device nodes: Type:

```
ls -l /dev | grep "^b.* 2,  [0-7]"
```

For me, that lists:

```
brw-r--r-- 1 root wheel 2, 0 Mar 31 18:16 df0
brw-r--r-- 1 root wheel 2, 1 Mar 31 18:16 df1
brw-r--r-- 1 root wheel 2, 2 Mar 31 18:16 df2
brw-r--r-- 1 root wheel 2, 3 Mar 31 18:16 df3
brw-r----- 1 root 25 2, 0 Jun 22 1996 fd0-
brw-r----- 1 root 25 2, 4 Feb 26 1994 fd0d360
brw-r----- 1 root 25 2, 4 Jun 22 1996 fd0d360-
brw-r----- 1 root 25 2, 5 Jun 22 1996 fd1d360
brw-r--r-- 1 root wheel 2, 4 Apr 4 11:49 mfd0
brw-r--r-- 1 root wheel 2, 5 Apr 4 11:49 mfd1
brw-r--r-- 1 root wheel 2, 6 Apr 4 11:49 mfd2
brw-r--r-- 1 root wheel 2, 7 Apr 4 11:49 mfd3
brw-r--r-- 1 root wheel 2, 4 Mar 31 18:03 pc0
brw-r--r-- 1 root wheel 2, 5 Mar 31 18:16 pc1
brw-r--r-- 1 root wheel 2, 6 Apr 8 12:03 pc2
brw-r--r-- 1 root wheel 2, 7 Apr 8 12:03 pc3

^                ^  ^                =09  ^^^
block-device      major  minor          node-name=20
```

Do not worry if you have other results. What do you need to know?

- Floppies are block devices.

- Floppies have major device number 2.
- Under Linux/m68k, the minor device numbers are as follows:

```
0 to 3 ... Amiga drives 0 to 3 (i.e. df0 to df3)
4 to 7 ... MS-DOS drives 0 to 3 (i.e. pc0 to pc3)
```

Like under AmigaOS with CrossDOS, drive 0 is a single physical unit. Note that this is different from Linux/i386. You can verify this by reading Documentation/devices.txt of the kernel source. The nodes fd* above are remnants of a Intel configuration; only the major/minor numbers count, not the assigned name!

Now create the device-nodes:

```
mknod /dev/df0 b 2 0
mknod /dev/df1 b 2 1
mknod /dev/df2 b 2 2
mknod /dev/df3 b 2 3

mknod /dev/pc0 b 2 4
mknod /dev/pc1 b 2 5
mknod /dev/pc2 b 2 6
mknod /dev/pc3 b 2 7
```

Set the desired permissions with the chmod command.

Remember, the names (here: df0 and pc0) are arbitrary. However, on Intel Linux systems, the floppy nodes are named /dev/fd*. To access devices under different node-names, just create symlinks. e.g:

```
ln -sf /dev/pc0 /dev/fd0
ln -sf /dev/pc1 /dev/fd1
ln -sf /dev/pc2 /dev/fd2
ln -sf /dev/pc3 /dev/fd3
```

Now, MS-DOS drive 0 can be accessed by /dev/fd0 as well as /dev/pc0. If you want /dev/fd0 to be an Amiga drive, link it to /dev/df0 instead. Of course, this are just examples from my configuration. You can choose other names if you like.

After having mount-point and device-node, you can mount your floppy.

For an AmigaOS disk in drive 0:

```
mount -t affs /dev/df0 /df0
ls /df0
...
umount /df0
```

For a MS-DOS disk in drive 1:

```
mount -t msdos /dev/pc1 /pc1
ls /pc0
...
umount /pc0
```

Warnings:

- Only mount if floppy already in drive and
- you must not remove the floppy before umount'ing it!

You can also put this into your `/etc/fstab` file. Mine looks like this:

```
# device      mountpoint    type    options                                freq passno
# -----
# Amiga Floppies
/dev/df0      /ados/df0     affs    defaults,noauto,noexec                0 2
/dev/df1      /ados/df1     affs    defaults,noauto,noexec                0 2
/dev/df2      /ados/df2     affs    defaults,noauto,noexec                0 2
/dev/df3      /ados/df3     affs    defaults,noauto,noexec                0 2
#
# MS-DOS Floppies
/dev/pc0      /ados/pc0     msdos   defaults,noauto,noexec                0 2
/dev/pc1      /ados/pc1     msdos   defaults,noauto,noexec                0 2
/dev/pc2      /ados/pc2     msdos   defaults,noauto,noexec                0 2
/dev/pc3      /ados/pc3     msdos   defaults,noauto,noexec                0 2
```

I'm not quite sure about the `freq/passno` fields. Do a "man 5 fstab" and a "man 8 mount" for more info.

Other topics about floppies:

8.7.1 mtools

You can use the "mtools" package to access MS-DOS disks without the need of mount/umount. The `mtools-3.6.tar.gz` package compiled without any problems out of the box for me. The nodes `/dev/fd0` and `/dev/fd1` are used to access the MS-DOS drives. If you followed my descriptions above it is not necessary to edit `mtools.conf` (in `/etc` or `/usr/local/etc`)

8.7.2 formatting

Hhm. Good question. There are some binaries in `bin/system/floppy` at `ftp.uni-erlangen.de`. Unfortunately for me, `fdformat` dies with a segmentation fault and `amifdformat`-formatted disks can't be mounted using `affs`. Any suggestions are welcome!

8.8 Can I use an Amiga-formatted partition as my root partition?

The current `affs` driver doesn't support Unix-style devices on Amiga partitions, which makes it impossible to use an Amiga partition as root. A future release may support an emulation of devices and other Unix special files, perhaps based on the `umsdos` filesystem already available (or you can always try to code the support yourself).

You can use `affs` disks as secondary partitions. The AmigaOS 2.0+ symbolic and hard links are used to emulate the similar Unix features (hard links work as expected; soft links outside of the same directory will not work as expected from both sides, because of the differing semantics of AmigaOS and Linux path names).

8.9 Can I run Linux on my Phase5 PowerUP card's PowerPC CPU?

Yes, if you really know what you are doing. Jesper Skov writes:

```
We released the first beta version of a Linux 2.1.79 port to Phase5's
Amiga PowerUP hardware on January 29 (1998).  A current kernel image
and kernel diffs can be found at:
```

```
ftp://sunsite.auc.dk/local/os/linux/apus/
```

```
The kernel is still in a beta state and is not suitable for users. We
just decided it was time that we got some feedback from other hackers
- and to let everyone know that APUS (Amiga PowerUP Systems) support
is a work in progress.
```

```
Incidentally, the port has been at its current state for a couple of
months, but we have had some problems with Phase5 that have now been
resolved. Phase5 is very interested in seeing this port completed and
has been very helpful lately. We appreciate this very much.
```

```
The APUS specific changes should be merged into the vger tree
RSN. Progress reports will be posted on the m68k kernel list.
```

```
I will see to that the m68k FAQ is updated on this subject. Please
refer people to www.linux-m68k.org.
```

```
Thanks!
```

```
JEesper on behalf of:
```

```
Roman Zippel : zippel@fh-brandenburg.de
Jes Sorensen  : Jes.Sorensen@cern.ch
Jesper Skov   : jskov@cygnus.co.uk
```

There is now an APUS-specific FAQ available on the web at <http://sunsite.auc.dk/ftp/projects/apus/docs/faq.html>. Information on the Linux/APUS mailing list is in the APUS FAQ.

CyberstormPPC cards are reported to be working well; BlizzardPPC cards appear to be more problematic (they seem to work for some people but not for others). Non-Phase5 PowerPC cards will eventually be supported (when/if they ever make it to market).

Also note that Linux/m68k will work fine with the 680x0 series CPU which is also installed on your PowerUP card.

8.10 Can I use an IDE doubler with Linux/m68k?

Some IDE doublers (devices that let you attach up to four IDE devices to the A1200 and A4000 internal IDE controllers) may work with the following kernel option: `ide=doubler`. Those that conform to the IDEfix "standard" should work without modifications.

8.11 What video modes does my card support?

Using `fbset`, you can program video modes (see the framebuffer documentation in the kernel tree for details). However, there are some modes built into most drivers that you can specify at boot time: typical mode names are 640x480, 800x600 and 1024x768. Some cards can also autodetect the mode you enable in AmigaOS and use it.

As of kernel 2.1.124, here is a fairly comprehensive list of supported modes. “bpp” stands for “bits per pixel”: it is the color depth. All modes are non-interlaced unless otherwise specified.

- CLGen (Piccolo; Piccolo SD64; Picasso II; Picasso IV; Spectrum):
 - Autodetect: Use mode set in AmigaOS (with `-keep-video` option to `Amiboot`)
 - 640x480: 31.25 kHz, 60 Hz, 25 MHz PixClock (8 bpp)
 - 1024x768: 55.8 kHz, 70 Hz, 80 MHz PixClock (8 bpp)
- Cybervision 64 (original and 3D):
 - 640x480-8
 - 800x600-8
 - 1024x768-8
 - 1152x886-8
 - 1280x1024-8
 - 1600x1200-8
 - 800x600-16
- Retina Z3:
 - 640x480: 8 bpp
 - 800x600: 8 bpp
 - 1024x768i: 8 bpp, interlaced
 - 640x480-16: 16 bpp
 - 640x480-24: 24 bpp

8.12 The Amiga memory device driver (z2ram)

8.12.1 What is the z2ram device?

Note: The available minors depend on the kernel version you are using; this discussion pertains to the driver as included the 2.2 kernel series (the 2.0 kernel series does not support the “memory list” options). As of 2.2.1, z2ram has the same capabilities on both Linux/m68k and Linux/APUS.

Basically, the z2ram device driver allows you to use memory that is not being used by Linux/m68k as swap space or a ramdisk (you could use it for a `/tmp` partition, for example). While the name implies the device only is useful if you have “Zorro II” memory, it actually permits the use of any chip or fast memory on your system that is not already being used by Linux/m68k.

The z2ram driver is enabled using the “Amiga Zorro II ramdisk support” option during kernel configuration. It may already be a module in your default kernel setup.

It is a block device (major 37); you can have a number of different z2ram devices operating at once. Each minor number provides access to different areas of memory:

- Minor 0: Use available chip and Zorro II fast memory
- Minor 1: Use only Zorro II-addressable fast memory
- Minor 2: Use only chip memory
- Minor 3: (unused)
- Minors 4-7: Use memory list entry 1-4 (see below)

Minor 1 is most useful if you have Zorro II memory on your system (perhaps on a SCSI controller card) that is slower than the memory on your motherboard or accelerator card, but is still faster than a hard disk. These memory areas are automatically excluded from your system memory during the boot process. Note that you may have Zorro II memory even if you don’t have a Zorro bus on your computer (for example, if you have an A500 or A1000 expansion box; PCMCIA memory cards may also be seen as Zorro II memory on the A600 and A1200).

Minors 0 and 2 provide access to chip memory. While this can be useful at times (and chip memory is generally faster than Zorro II memory on the A3000 and A4000), parts of the kernel that expect to be able to get extra chip memory on demand may cause problems. The amifb framebuffer device is one example of a device that may need chip memory “on the fly” (for example, if you increase the depth or size of a framebuffer); amifb may react very poorly to running out of memory it expects to find. If you are using another framebuffer instead of amifb, chip memory access through the z2ram device may be less problematic.

Minor 3 isn’t used by anything at the moment.

Minors 4 through 7 were developed for Linux/APUS, but can also be useful under Linux/m68k on Amigas with more than one memory area where the access times to the different types of memory significantly differ (i.e. an Amiga with an accelerator card that has on-board SIMM sockets and other memory on the motherboard or expansion bus). In these situations, you can use the z2ram device in conjunction with the Linux paging code to help ensure that slower memory is only used when the faster memory blocks are completely full. To take advantage of this capability, you should configure your kernel to “Use one physical chunk of memory only” (under the “Advanced options” section after you select the CPU you want to use).

Once you have recompiled your kernel this way, you should make sure the memory block with the best access to your CPU is being used as your main system memory (usually this will be the memory on your accelerator card or on the motherboard), i.e. is listed first when amiboot lists the chunks of memory that have been found. You may need to make a “memory list” for amiboot if the AmigaOS memory priorities are not ordered sensibly. In any event, amiboot will list the chunks of memory it decides are available in the order they should be used; the first chunk listed will be used as your main system memory.

The remaining chunks (actually, the second through fifth chunks... if you have more than five memory chunks, you can hack the driver to permit more) can be used by the z2ram driver using the above-listed minor numbers. So the second chunk on the list will be minor 4, the third chunk will be minor 5, etc.

8.12.2 Ok, now that I know what the device does, how do I use it?

You have two options: you can use each z2ram area as either swap space or a partition (but not both at the same time for a particular instance... although you could put a swap file on a z2ram partition). Basically all you have to do is enable the z2ram driver in your kernel and create the block special files you need in /dev; then you can treat each z2ram area like any other block device (such as a floppy or hard disk drive).

Note that if you have multiple blocks of memory, you can use some blocks as swap and others as partitions; you just can't use the same memory block for both purposes at once.

Specific instructions for both uses follow:

Swap space

1. Make a block special file for the device. This is done with the `mknod` command. For example, to make a device called `"/dev/fastram"` that uses your Zorro II memory, use `mknod -mode=600 /dev/fastram b 37 1`. Here 1 is the minor number used for Zorro II memory (see above), it could also be 0, 2, 4, 5, 6 or 7, as appropriate.
2. Prepare the swap space for use. Following the example above, `mkswap /dev/fastram`.
3. Make the swap space available to the system. For this, use the `"swapon"` command: `swapon -p 1 /dev/fastram`. The `-p 1` parameter tells the kernel to put this swap space in at a higher priority (1) than the default (a negative number, which depends on how any swap spaces are already enabled), so it will be used before your hard disk.

To make this change permanent, you will need to edit your system startup files to prepare and enable the swap space on each boot (by adding the `"mkswap"` and `"swapon"` lines to one of your early startup scripts... on Debian, you should add a small script to `/etc/rcS.d` around priority S04: see the Debian init information for more details). Here's a simple init script that will handle this procedure during each boot:

```
mkswap /dev/fastram
swapon -p 1 /dev/fastram
```

If you are using z2ram as a module, you may need to make sure the module dependencies are available when the script runs so `kmod` (assuming you use it) can automatically load the driver.

Ramdisk The principles involved in the ramdisk are similar to those for a swap partition:

1. Make a "block special file" for the device. This is done with the `mknod` command. For example, to make a device called `"/dev/mbram"` that uses your motherboard memory (which, I assume for the purposes of this example, is the second entry on your memory list), use `mknod -mode=600 /dev/mbram b 37 4`. Here 4 is the minor number used for the second memory list entry (see above), it could also be 0, 1, 2, 5, 6 or 7, as appropriate.
2. Prepare the ramdisk for use: `mke2fs /dev/mbram` will do the job nicely (you could also use another filesystem format if you like, but `ext2` is probably the best for most uses).

3. Make the ramdisk available to the system. For this, just mount it somewhere; e.g. `mount -t ext2 /dev/mbram /mnt`. The `-t ext2` tells mount that you have an ext2fs filesystem, and should be changed if you use some other filesystem like Minix.

To set up a /tmp ramdisk for each boot, you should edit your startup scripts to format the ramdisk (since, for obvious reasons, the ramdisk doesn't stay formatted over reboots) and mount it. You may also need to set the permissions for the tmp directory properly on each boot. Here's a sample script that takes care of things:

```
mke2fs -q /dev/mbram
mount -t ext2 /dev/mbram /tmp
chown root.root /tmp
chmod 1777 /tmp
```

On a Debian system, this should be dumped into an rcS.d file; see the Debian policy manual for an explanation of the Debian init file scheme. The note under the swap space section about z2ram as a module may also apply.

9 Atari-specific questions

9.1 I can't use ttyS3 and ttyS4 simultaneously on my Atari

This is perfectly normal: ttyS3 and ttyS4 correspond to different physical external connectors on the Atari, but these connectors use the same internal hardware (Channel A of the SCC).

9.2 I'm having problems with my Falcon with Afterburner 040. Any tips?

(written by Roman Hodek, Petr Stehlik and Thomas Kruse)

Roman says:

On a Falcon with Afterburner040, the bootstrap can be run only on a fully initialized machine, i.e. the AB040 software drivers must be loaded. More specifically, bootstrap relies on the `_CPU` cookie to be set to 68040, which is done by that driver. But there may be also other dependencies...

Additionally, the bootstrap must have its program flags set to "don't run in TT-RAM" and "don't allocate memory from TT-RAM". This is due to the fact that TT-RAM on the AB040 is mapped by the MMU and addresses in it can become invalid as soon as the MMU is switched off before launching Linux. Bootstrap issues an error message if either its code or data reside in TT-RAM, so you can't make it wrong, but maybe you don't know how to fix it... :-)

And finally: Currently the Linux kernel won't run properly in TT-RAM. Better use the `-s` flag to bootstrap ("load kernel to ST-RAM"). We're working on the problem, but a solution isn't evident yet...

Petr provides the following advice:

The Afterburner040 is a card with 68040 CPU and two FastRAM slots for the Atari Falcon 030. There are several different revisions of that card that also affect Linux. General rules for successful Linux boot are as follows:

1. Use bootstrap (ataboot) version 3.1 or higher (older versions of ataboot didn't recognize FastRAM properly)
2. Use kernel version 2.1.72 or higher (older versions could not reboot the machine, but generally worked, too)
3. The kernel must be compiled for the 68040 CPU only (so no 68020, 68030 nor 68060 support!), otherwise it won't boot. The reason for this is unknown at the moment (December 1997). Also, for certain revisions of the Afterburner040 card the specific 68040 optimizations (RMW) must not be used as well. You can either compile your own kernel or visit <http://ft3.zlin.vutbr.cz/stehlik/soft.htm>, where you can find some pre-compiled kernels for Afterburner040.
4. ataboot must run in ST-RAM and it must also allocate ST-RAM, so please clear both 'Run in TT-RAM' and 'Allocate TT-RAM' flags in the header of ataboot. For this purpose you can use programs like SETFLAGS.PRG or similar.
5. Accelerated Falcons (higher bus speed than 16 MHz) usually suffer from some strange problem with lost MFP interrupts when kernel is loaded in FastRAM (might be recognized by too high BogomIPS value at bootup: standard value for Afterburner040/32 MHz is 21.29), so it's good idea to put the kernel into ST-RAM with the '-s' flag on the ataboot command line.
6. The 'Swap-to-ST-RAM' option of Linux kernel 2.1.72+ might be very useful on Afterburner040 because the ST-RAM is very slow (about ten times slower than FastRAM) and that's a good reason for not using it as normal 'working' memory.

Thomas Kruse says that "not only must the read modify write optimization be turned off, the *whole* 68040 specific optimizations must be turned off - otherwise the kernel won't run reliably on different circumstances and sizes of the kernel."

The discussion on the mailing list seems to indicate that there isn't a specific approach that seems to work well for everyone. Hopefully this situation will be resolved soon.

9.3 I'm having strange problems with my PAK030 board in my ST!?!

(written by Roman Hodek)

For users of the PAK030 board for STs, burst mode for memory is best disabled when running Linux. With burst mode enabled, several users experienced spurious memory corruption problems.

9.4 How do I access my Atari SLM laser printer?

The laser printer is accessed through `/dev/slm0`; if you have multiple printers, you can use e.g. `/dev/slm1`.

9.5 How do I access my ACSI drives?

ACSI drives are accessed through `/dev/adXY`, where X is the drive letter (a-p, corresponding to the first through sixteenth drives) and Y is the partition number.

10 Macintosh-specific questions

10.1 Where the **** is the Mac site?

It is supposed to be at <http://www.mac.linux-m68k.org/>; if it's not there, try <http://maclinux.wwaves.com/>.

If it's at neither of those places, it has disappeared for a while (these things happen). Try <http://shadow.cabi.net/MacLinux/> for some out-of-date information. If you just want to install Debian on a Mac, try the *Debian/m68k for Macs Installation Guide* <http://www.linux-m68k.org/debian-mac.html>. There may be some files of interest at <ftp://baltimore.wwaves.com/>.

Note that the Linux/m68k web site maintainer does *not* maintain the Mac site, and usually knows less than you do about its current status. When he does know something, it will be put in the FAQ.

11 Other sources for information, sources and binaries

11.1 Installation Guides

A fairly up-to-date list of installation guides is kept on the *distributions page* <http://www.linux-m68k.org/dists.html> at the Linux/m68k Home Pages; see also 11.8 (Distributions) in this FAQ.

11.2 Other Documents

Further documents can be found at *the Linux Documentation Project's Home Page* <http://metalab.unc.edu/LDP/>. These documents were originally written for Linux/i386, but many are useful for Linux/m68k users too (e.g. HOWTOs on UUCP, PPP and the general Linux FAQ).

A FAQ on Motorola chips (including the 680x0 microprocessors) is available at <http://www.oise.on.ca/~rboys/m68kfaq.html>.

Last but not least: Look into the Documentation directory of the 2.x kernel trees.

11.3 Newsgroups

comp.os.linux.m68k

The group *comp.os.linux.m68k* is intended to further interest in, and development of, the port of the Linux operating system to the 680x0 architecture. All discussion in the newsgroup should be in English. The group's RFD (Request for Discussion), CFVs (Calls for Votes) and final vote tally, along with the group charter, can be found at <http://www.hensa.ac.uk/ftp/mirrors/uunet/usenet/news.announce.newsgroups/comp/comp.os.linux.m68k>.

comp.os.linux.development.system

This group is on Linux *kernel* development only. From time to time it contains messages dealing with the Linux/m68k kernel.

comp.os.linux.announce

This group announces new Linux-related products. Announcements for new versions of Linux/m68k may be found there.

maus.os.linux68k

This group deals with Linux/m68k only. The languages currently used are German and English. This newsgroup is also available via FidoNet (as LINUX-68K.GER).

comp.unix.amiga

This group is for discussions on Amiga Unix, Minix, NetBSD and Linux/m68k on the Amiga.

de.comp.sys.amiga.unix

Similar to comp.unix.amiga, but in German.

11.4 Mailing Lists

There is one mailing list for Linux/m68k, which is named linux-m68k. As there is now a newsgroup for Linux/m68k, topics on this list should be restricted to development issues if possible.

(written by Benjamin Lorenz)

You can subscribe to linux-m68k@lists.linux-m68k.org by sending a mail to *linux-m68k-request@lists.linux-m68k.org*, with a random subject and a single line in the mail body containing "subscribe linux-m68k". You may want to subscribe to linux-m68k-digest@lists.linux-m68k.org instead: in this case, you will receive one mail per day containing all mails to the list from the last 24 hours. If you prefer to read mail in this way, please unsubscribe from linux-m68k to reduce net load!

You can download archives of the digest mails! They are stored at `<ftp://ftp.phil.uni-sb.de/pub/linux-m68k/maillinglist>`.

Another mailing list archive that supports searching can be found at `<http://aire.ncl.ac.uk/Atari/Mailing-Lists/Linux-m68k-phil-List.index.html>`.

The kernel list is also available from sunsite.auc.dk as a nntp news feed (`<nntp://sunsite.auc.dk/sunsite.linux.m68k>`). It is the fifth or so member on the mailing list so it's *fast*.

Other mailing lists are available for more specialized purposes; I recommend visiting *the Linux/m68k Home Pages* `<http://www.linux-m68k.org/mail.html>` for further details.

11.5 WWW sites

- The Linux/m68k Home Pages: `<http://www.linux-m68k.org/>` (The Netherlands), `<http://www.clark.net/pub/lawrency/linux/>` (USA), `<http://home.olemiss.edu/~cnlawren/linux/>` (USA), `<http://www.lordsutch.com/linux/>` (USA), `<http://amiga.nvg.org/linux/mirrors/lawrency/>` (Norway), or `<http://www.se.linux-m68k.org/>` (Sweden).
- Linux/m68k Registration Site (Geert Uytterhoeven): `<http://www.cs.kuleuven.ac.be/~geert/Linux/m68k/>`.

- Dirk Wetter's Linux/m68k WWW page: <http://bunsen.pci.uni-hannover.de/linux68k.html>.
- For information on Linux in general, try <http://www.linuxhq.com/>, <http://metalab.unc.edu/LDP/> and <http://www.linuxnow.com/>.

11.6 FTP sites

- Linux/m68k 2.0, 2.1 and 2.2 kernels: <ftp://sunsite.auc.dk/local/os/linux/680x0/> or <http://sunsite.auc.dk/ftp/local/os/linux/680x0/>.
- The Linux/m68k server: <ftp://ftp.uni-erlangen.de/pub/Linux/680x0/>.
- Mirrors of this server include (please use the one nearest to you, most of these mirrors are updated daily):
 - <ftp://ftp.belnet.be/packages/Linux-680x0/>
 - <ftp://ftp.fu-berlin.de/pub/atari/linux/>
 - <ftp://ftp.funet.fi/pub/Linux/BETA/680x0/>
 - <ftp://ftp.germany.eu.net/pub/os/Linux/Mirror.SunSITE/>
 - <ftp://ftp.informatik.tu-muenchen.de/pub/comp/os/linux/680x0/>
 - <ftp://ftp.nvg.unit.no/pub/linux/680x0/>
 - <ftp://ftp.phil.uni-sb.de/pub/linux-m68k/erl/>
 - <ftp://ftp.spc.uchicago.edu/pub/linux/680x0/>
 - <ftp://ftp.tu-clausthal.de/pub/systems/Linux/680x0/>
 - <ftp://ftp.lip6.fr/pub/atari/Linux68k/linux-m68k/>
 - <ftp://ftp.twi.tudelft.nl/pub/Linux/680x0/>
 - <ftp://ftp.unina.it/pub/linux/linux68k/>
 - <ftp://src.doc.ic.ac.uk/computing/operating-systems/Linux/tsx-11-mirror/680x0/>
 - <ftp://tsx-11.mit.edu/pub/linux/680x0/>
 - <ftp://ftp.kernel.org/pub/mirrors/tsx-11/680x0/>
 - <ftp://ftp.linuxberg.com/pub/distributions/Linux-m68k/>
 - Please tell me if your favorite mirror is not on this list.
 - The kernel source for Linux/m68k can be found in [680x0/v2.0/](ftp://tsx-11.mit.edu/pub/linux/680x0/v2.0/) and [680x0/v2.1/](ftp://tsx-11.mit.edu/pub/linux/680x0/v2.1/), a lot of binaries in [680x0/bin/](ftp://tsx-11.mit.edu/pub/linux/680x0/bin/). A few more tools reside in [680x0/tools/](ftp://tsx-11.mit.edu/pub/linux/680x0/tools/).
- The two main Linux servers are:
 - <ftp://tsx-11.mit.edu/pub/linux/sources/>
 - <ftp://metalab.unc.edu/pub/Linux/system/>
- Linux on Amiga: <ftp://ftp.informatik.uni-oldenburg.de/pub/amiga/linux/local/>

- Various stuff: <ftp://ftp.phil.uni-sb.de/pub/linux-m68k/>

The following addresses are known to offer FTP via e-mail:

- ftpmail@info2.rus.uni-stuttgart.de <<mailto:ftpmail@info2.rus.uni-stuttgart.de>>
- ftpmail@ftp.inf.tu-dresden.de <<mailto:ftpmail@ftp.inf.tu-dresden.de>>
- ftpmail@informatik.uni-oldenburg.de <<mailto:ftpmail@informatik.uni-oldenburg.de>>

To get more info on FTP-mail send a mail with subject “help” to one of the addresses mentioned above.

11.7 Modem

If you have a modem, you can (could?) get Linux/m68k from the following location in Germany:

```
System name: nasim
Phone: +49 89 5469593, ZyX19200
Login: Anon-uucp: nuucp - no password / ZModem: gast - no password
Contents: full 680x0-tree of tsx-11 in /pub/linux-68k
Get first: index file /pub/linux-68k/ls-lR.nasim.linux-68k.gz
Other features: provides uucp access to 680X0 channel (read only) and
the linux.act.* news-groups
Admin: Frank Bartels <knarf@nasim.cube.net>
```

11.8 Distributions

ALD, the Autoren Linux68K Distribution for the Atari, is available from Whiteline (anyone got any information on its current status?)

You can use Debian/m68k 2.0 [hamm] (and, after March 2, Debian/m68k 2.1 [slink]), and help develop future versions (like potato; Debian releases have names based on the names of characters in Pixar’s film “Toy Story”). Debian is available via FTP at <ftp://ftp.debian.org/debian/>; installation guides are available for *Amigas* <http://www.informatik.uni-oldenburg.de/~amigo/debian_inst.html>, *Ataris* <<http://www.linux-m68k.org/debian-atari.html>> and *Macs* <<http://www.linux-m68k.org/debian-mac.html>>. Debian is also available on CD-ROMs from several vendors <<http://www.debian.org/distrib/vendors>> including *Linux Systems Labs* <<http://www.lsl.com/>> and *Chris Lawrence* <<http://www.clark.net/pub/lawrenc/cc/debian-m68k.html>>.

You can also use and help improve Jes Sørensen’s unofficial Red Hat port, available at <ftp://sunsite.auc.dk/projects/680x0/redhat/>. Ron Flory has written an installation guide for this port; you can get it at <http://www.feist.com/~rjflory/linux/rh/rh-inst.html>. Red Hat Software is distributing this unofficial port (along with Red Hat ports for PowerPC, UltraSPARC and MIPS) as part of its *Linux Rough Cuts* <<http://www.redhat.com/products/product-details.phtml?id=rc>> package; you can also obtain it from *Holger Lubitz* <<http://linuxservice.de/>>.

Lastly, Eagle Linux is a distribution available for the Amiga, based on the Debian 2.0 project. You can read more about it at Eagle’s web site, <<http://www.eagle-cp.com/>>.

11.9 IRC (Internet Relay Chat)

An IRC channel for Linux/m68k exists, called #Linux68k. Start irc and select one of the servers of LinuxNET, such as `irc.blackdown.org`. A list of servers can be found at `<http://www.blackdown.org/LinuxNET/LinuxNET.html>`. Just do a

```
/join #Linux68k
```

and look out for others. If nobody is there, there is a simple reason for this: you are the first to join this channel (actually, the first one to join a channel creates this channel). There already is a well-established #Linux.

Note: The attempt to establish a channel #Linux68k via the “regular” IRC servers more or less failed because of many, many net splits. LinuxNET is much more reliable.

11.10 Magazines

Linux Journal `<http://www.ssc.com/lj/>` (ISSN 1075-3583) is the monthly magazine of the Linux community. It is aimed at everyone from the casual user to die-hard kernel hacker. In the US, most large booksellers carry the magazine.

Linux Gazette `<http://www.ssc.com/lg/>` is a monthly on-line publication with tricks and tips from ordinary users, along with longer how-to articles. You can read it on the World Wide Web at the location above; issues are also available as Debian packages.

11.11 Books

Try the new *books page* `<http://www.linux-m68k.org/books.html>` at the Linux/m68k Home Pages for an expanded selection of books (from the selection that appeared here before).

12 Famous last words

12.1 Credits

I want to thank everyone who contributed to this FAQ. There are many people who did so by answering questions in the newsgroups or on the mailing list, or by asking the questions. Some sections are marked “written by”; this means that the text was originally written by that person but has been edited by Jörg or myself.

Extra thanks to the following people for their suggestions and submissions:

- Martin Apel `<apel@tecmath.de>`
- Richard Hirst `<srh@gpt.co.uk>`
- Roman Hodek `<rnhodek@informatik.uni-erlangen.de>`
- David Kilzer `<ddkilzer@earthlink.net>`

- Thomas Kruse <tkruse@home.globe.de>
- Benjamin (Benni) Lorenz <benni@phil.uni-sb.de>
- Joaquin Menchaca <menchaca@tibco.com>
- Frank Neumann <Frank.Neumann@informatik.uni-oldenburg.de>
- Jim Partan <jimp@waves1.who.edu>
- Joe Pranevich <joepran@telerama.lm.com>
- Raoul van Putten <rlputten@cip.informatik.uni-erlangen.de>
- Jesper Skov <jkskov@cygnus.co.uk>
- Christian Steigies <steigies@physik.uni-kiel.de>
- Petr Stehlik <pstehlik@zln.cz>
- Geert Uytterhoeven <Geert.Uytterhoeven@cs.kuleuven.ac.be>
- Haidinger Walter <e9225662@student.tuwien.ac.at>
- Last, but certainly not least, Jörg Mayer <jmayer@telemation.de>, who started the ball rolling and wrote about 75% of this.

Thanks also to Christian Jacolot <jacolot@ubolib.univ-brest.fr> for translating this FAQ into French and keeping it updated.

Some credit is also due to J. Michael Straczynski <jmsatb5@aol.com>, to whom I owe a few of the section titles (but *please* don't mail him Linux questions...).

The phrase "go home and eat popcorn" (and various derivatives thereof) is a registered trademark of G. Elton Graves, Ph.D. <G.E.Graves@rose-hulman.edu>; all rights reserved. Don't bother him with Linux questions either.

Suggestions can be made to Chris Lawrence <faq@linux-m68k.org>. I also read `comp.os.linux-m68k`, `comp.unix.amiga` and the `linux-m68k` list, looking for suggestions.

12.2 Copyright and License

This FAQ is Copyright © 1995-96 Jörg Mayer and Copyright © 1997-99 Chris Lawrence.

This FAQ may be freely redistributed under the terms of the GNU General Public License, version 2 (or at your discretion, any later version of that license).

You may also freely redistribute formatted versions of the verbatim FAQ (i.e. the FAQ as it was released by the original copyright holders) with or without providing or offering the SGML source.

Amiga, Atari, Commodore, Motorola, MS-DOS, Sun, Unix and maybe a few more words I used in this text are trademarks. So what?