Linux Frequently Asked Questions with Answers

David Merrill

david@lupercalia.net

Robert Kiesling

Revision History

Revision 1.20	2001−12−04	Revised by: rk

This is the list of Frequently Asked Questions for Linux, the free operating system kernel that runs on many modern computer systems. The kernel source code documentation says that Linux ``aims for POSIX compliance.'' Linux uses mostly free, GNU system utilities and application software, although commercial programs are available also. Originally written for 386/486/586 Intel/ISA bus machines, Linux versions exist for nearly every hardware platform in existence that is capable of running it. (Please refer to the question, ``*What Is Linux?*'' below.) This FAQ is meant to be read in conjunction with the Linux Documentation Project's HOWTO series. (``*Where Are the Linux FTP Archives?*'' and, ``*Where Is the Documentation?*'') The *INFO−SHEET* and *META−FAQ* also list sources of Linux information. Please read them, and, ``*If this Document Still Hasn't Answered Your Question....*'' before posting to a Usenet news group. You can also get Postscript, PDF, HTML, and SGML versions of this document. (``*Formats in Which This FAQ Is Available.*'') Linux Frequently Asked Questions with Answers is distributed under the terms of the GNU Free Documentation License. Refer to ``*Disclaimer and Copyright.*.''

Table of Contents

1. Introduction and General Information....1
1.1. What Is Linux?....1
1.2. How to Get Started....1
1.3. What Software does Linux Support?....2
1.4. How to Find a Particular Application....2
1.5. What Hardware Is Supported?....3
1.6. Ports to Other Processors....3
1.7. Disk Space Requirements: Minimal, Server, and Workstation....5
1.8. Minimum and Maximum Memory Requirements....5
1.9. Does Linux Support Universal System Bus Devices?....6
1.10. What Is Linux's Open−Source License?....6
1.11. Is Linux *nix?....7

2. Network Sources and Resources....8
2.1. Where Is the Latest Kernel Version on the Internet?....8
2.2. Where Is the Documentation?....8
2.3. Where Is the Linux Stuff on the World Wide Web?....10
2.4. What News Groups Are There for Linux?....11
2.5. What Other FAQ's and Documentation Are There for Linux?....13
2.6. Where Are the Linux FTP Archives?....14
2.7. How To Get Linux without FTP Access....15
2.8. How To Get Information without Usenet Access....16
2.9. What Mailing Lists Are There?....16
2.10. Where Are Linux Legal Issues Discussed?....16
2.11. Sources of Information for Unmaintained Free Software Projects....17
2.12. Are the News Groups Archived Anywhere?....17
2.13. Where To Find Information About Security Related Issues....17
2.14. Where To Find Linux System Specifications....17

3. Compatibility with Other Operating Systems....18
3.1. Can Linux Use the Same Hard Drive as MS−DOS? OS/2? 386BSD? Win95?....18
3.2. How To Access Files on a MS−DOS Partition or Floppy....18
3.3. Does Linux Support Compressed Ext2 File Systems?....19
3.4. Can Linux Use Stacked/DBLSPC/Etc. DOS Drives?....19
3.5. Can Linux Access OS/2 HPFS Partitions?....20
3.6. Can Linux Access Amiga File Systems?....20
3.7. Can Linux Access BSD, SysV, Etc. UFS?....20
3.8. Can Linux Access SMB File Systems?....20
3.9. Can Linux Access Macintosh File Systems?....21
3.10. Can Linux Run Microsoft Windows Programs?....21
3.11. Where Is Information about NFS Compatibility?....21
3.12. Can Linux Use True Type Fonts?....22
3.13. Can Linux Boot from MS−DOS?....22
3.14. How Can Linux Boot from OS/2's Boot Manager?....23

4. File Systems, Disks, and Drives....24
4.1. How To Get Linux to Work with a Disk....24
4.2. How To Undelete Files....24

Table of Contents

4.3. How To Make Backups....25
4.4. How To Resize a Partition (Non−Destructively)....25
4.5. Is There a Defragmenter for Ext2fs?....25
4.6. How To Create a File System on a Floppy....25
4.7. Does Linux Support Virtualized File Systems Like RAID?....26
4.8. Does Linux Support File System Encryption?....26
4.9. Linux Prints Nasty Messages about Inodes, Blocks, and the Like....26
4.10. The Swap Area Isn't Working....27
4.11. How To Add Temporary Swap Space....27
4.12. How To Remove LILO So the System Boots DOS Again?....28
4.13. Why Does fdformat Require Superuser Privileges?....29
4.14. The System Checks the Ext2fs Partitions Each Reboot....29
4.15. Root File System Is Read−Only....29
4.16. What Is /proc/kcore?....29
4.17. The AHA1542C Doesn't Work with Linux....29
4.18. Where Is the Journalling File System on the Net?....30

5. Porting, Compiling and Obtaining Programs....31
5.1. How To Compile Programs....31
5.2. How To Install GNU Software....31
5.3. Where To Get Java....31
5.4. How To Port XXX to Linux....32
5.5. What Is ld.so and How To Get It?....32
5.6. How To Upgrade the Libraries without Trashing the System....32
5.7. How To Use Code or a Compiler Compiled for a 486 on a 386....33
5.8. What Does ``gcc −O6'' Do?....34
5.9. Where Are linux/*.h and asm/*.h?....34
5.10. What To Do about Errors Trying to Compile the Kernel....34
5.11. How To Make a Shared Library....35
5.12. Programs Are Very Large....35
5.13. Does Linux Support Threads or Lightweight Processes?....35
5.14. Where To Find lint for Linux....36
5.15. Where To Find Kermit for Linux....36
5.16. How To Use Linux with a Cable Modem....36
5.17. Is There an ICQ Program That Runs under Linux?....36

6. Solutions to Common Miscellaneous Problems....37
6.1. FTP Transfers Seem to Hang....37
6.2. Free Dumps Core....37
6.3. Netscape Crashes Frequently....37
6.4. FTP or Telnet Server Won't Allow Logins....38
6.5. How To Keep Track of Bookmarks in Netscape?....38
6.6. The Computer Has the Wrong Time....39
6.7. Setuid Scripts Don't Seem to Work....39
6.8. Free Memory as Reported by free Keeps Shrinking....39
6.9. When Adding More Memory, the System Slows to a Crawl....39
6.10. Some Programs (E.g. xdm) Won't Allow Logins....40
6.11. Some Programs Allow Logins with No Password....40

Table of Contents

6.12. The Machine Runs Very Slowly with GCC / X / 40
6.13. System Only Allows Root Logins....41
6.14. The Screen Is All Full of Weird Characters Instead of Letters....41
6.15. I Screwed Up the System and Can't Log In to Fix It....42
6.16. I Forgot the root Password....42
6.17. There's a Huge Security Hole in rm!....43
6.18. lpr and/or lpd Don't Work....43
6.19. Timestamps on Files on MS−DOS Partitions Are Set Incorrectly....43
6.20. How To Get LILO to Boot the Kernel Image....44
6.21. How To Make Sure the System Boots after Re−Installing the Operating System....44
6.22. The PCMCIA Card Doesn't Work after Upgrading the Kernel....44
6.23. How To Remove (or Change) the Colors in the ls Display....46
6.24. Why Won't a Program Work in the Current Directory?....46

7. How To Do This or Find Out That....47
7.1. How To Find Out If a Notebook Runs Linux....47
7.2. Installing Linux Using FTP....47
7.3. Resuming an Interrupted Download....48
7.4. Boot−Time Configuration....49
7.5. Formatting Man Pages without man or groff....49
7.6. How To Scroll Backwards in Text Mode....50
7.7. How To Get Email to Work....50
7.8. Sendmail Pauses for Up to a Minute at Each Command....50
7.9. How To Enable and Select Virtual Consoles....51
7.10. How To Set the Time Zone....52
7.11. Dial−up PPP Configuration....52
7.12. What Version of Linux and What Machine Name Is This?....55
7.13. What Is a ``core'' File?....55
7.14. How To Enable or Disable Core Dumps....55
7.15. How To Upgrade/Recompile a Kernel....56
7.16. Can Linux Use More than 3 Serial Ports by Sharing Interrupts?....57
7.17. Configuring Emacs's Default Settings....57
7.18. How To Make a Rescue Floppy....57
7.19. How To Remap a Keyboard to UK, French, Etc.?....58
7.20. How To Get NUM LOCK to Default to On....58
7.21. How To Set (Or Reset) Initial Terminal Colors....58
7.22. How To Have More Than 128Mb of Swap....59
7.23. How To Prevent Errors when Linking Programs with Math Functions....59

8. Miscellaneous Information and Questions Answered....60
8.1. How To Program XYZ Under Linux....60
8.2. What's All This about ELF? glibc?....60
8.3. How To Determine What Packages Are Installed on a System....61
8.4. What Is a .gz File? And a .tgz? And .bz2? And... ?....61
8.5. What Does VFS Stand For?....62
8.6. What is a BogoMip?....62
8.7. What Online/Free Periodicals Exist for Linux?....63
8.8. How Many People Use Linux?....63

Table of Contents

8.9. How Many People Use Linux? (Redux.)....64
8.10. What Is the Best (Distribution|SCSI Card|Editor|CD−ROM Drive|....)....64
8.11. How Does One Pronounce Linux?....64

9. Frequently Encountered Error Messages....65
9.1. Modprobe Can't Locate Module, XXX, and Similar Messages....65
9.2. Unknown Terminal Type ``linux" and Similar....65
9.3. INET: Warning: old style ioctl... called!....66
9.4. ld: unrecognized option '−m486'....66
9.5. GCC Says, ``Internal compiler error."....66
9.6. Make Says, ``Error 139."....66
9.7. Shell−Init: Permission Denied when I Log In....67
9.8. No Utmp Entry. You Must Exec ... when Logging In....67
9.9. Warning−−bdflush Not Running....67
9.10. Warning: obsolete routing request made....67
9.11. EXT2−fs: warning: mounting unchecked file system....67
9.12. EXT2−fs warning: maximal count reached....68
9.13. EXT2−fs warning: checktime reached....68
9.14. df Says, ``Cannot read table of mounted file systems."....68
9.15. fdisk Says, ``Partition X has different physical/logical..."....69
9.16. fdisk: Partition 1 does not start on cylinder boundary....69
9.17. fdisk Says Partition n Has an Odd Number of Sectors....69
9.18. Mtools Utilities Say They Cannot Initialize Drive X....69
9.19. At the Start of Booting: Memory tight....70
9.20. The System Log Says, ``end_request: I/O error, ..."....70
9.21. ``You don't exist. Go away."....70
9.22. ``Operation not permitted."....70
9.23. programname: error in loading shared libraries: lib xxx..so. x: cannot open shared object file: No such file or directory....71
9.24. ``init: Id "x" respawning too fast: disabled for 5 minutes."....71
9.25. FTP server says: ``421 service not available, remote server has closed connection."....72

10. The X Window System....73
10.1. Does Linux Support X?....73
10.2. How To Get the X Window System to Work....73
10.3. Where To Find a Ready−Made XF86Config file....74
10.4. What Desktop Environments Run on Linux?....74
10.5. xterm Logins Show Up Strangely in who, finger....75
10.6. How to Start a X Client on Another Display....75

11. How to Get Further Assistance....76
11.1. If this Document Still Hasn't Answered Your Question....76
11.2. What to Put in a Request for Help....77
11.3. How To Email Someone about Your Problem....78

12. Acknowledgments and Administrivia....79
12.1. Where To Send Comments....79
12.2. Formats in Which This FAQ Is Available....79

Table of Contents

12.3. Authorship and Acknowledgments....79
12.4. Disclaimer and Copyright....80

1. Introduction and General Information

1.1. What Is Linux?

Linux is the kernel of operating systems that look like and perform as well or better than the famous operating system from AT&T Bell Labs. Linus Torvalds and a loosely knit team of volunteer hackers from across the Internet wrote (and still are writing) Linux from scratch. It has all of the features of a modern, fully fledged operating system: true multitasking, threads, virtual memory, shared libraries, demand loading, shared, copy−on−write executables, proper memory management, loadable device driver modules, video frame buffering, and TCP/IP networking.

Most people, however, refer to the operating system kernel, system software, and application software, collectively, as ``Linux,'' and that convention is used in this FAQ as well.

Linux was written originally for 386/486/586−based PC's, using the hardware facilities of the 80386 processor family to implement its features. There are now many ports to other hardware platforms. (``*Ports to Other Processors.*'')

There are also Linux distributions specifically for mobile and handheld platforms. An API specification and developers kit for the Crusoe Smart Microprocessor developed by Transmeta Corporation are at `http://www.transmeta.com/`. Information on the Linux distribution for the Compaq iPAQ is at `http://www.handhelds.org/`

Refer also to the Linux *INFO−SHEET* for more details as well as the answers to ``*Where Is the Documentation?*'', ``*What Hardware Is Supported?*'', and ``*Ports to Other Processors.*'', below. A list updated weekly is at: `http://lwn.net/` Archive of many of the distributions are on line at: `ftp://ftp.tux.org/` and `http://planetmirror.com/pub/linux`.

The Linux kernel is distributed under the GNU General Public License. (``*What Is Linux's Open−Source License?*'')

There is a historical archive of all versions of the Linux kernel at `http://ps.cus.umist.ac.uk/~rhw/kernel.versions.html`.

1.2. How to Get Started.

There are a handful of major Linux distributions. For information about them, and how they are installed, see Matthew Welsh's *Installation and Getting Started*, or *IGS* for short. It's located at the Linux Documentation Project Home Page, `http://www.linuxdoc.org/`, and on the Linux FAQ home page, `http://www.mainmatter.com/`

The information in *IGS* is somewhat dated now. More up−to−date information about first−time Linux installation is located in the LDP's *Installation HOWTO*, also located at the LDP Home Page.

Postings on the Usenet News groups, including the FAQ, are archived on `http://groups.google.com/`. Search for ``comp.os.linux.*,'' ``alt.uu.comp.os.linux.*`, or whatever is appropriate, to retrieve articles from the Linux News groups. (``*What News Groups Are There for Linux?*'')

Commercial distributions are available from book and electronics stores. Some hardware vendors now ship systems with Linux pre−installed.

There is a very thorough installation guide on line at
`http://heather.cs.ucdavis.edu/~matloff/linux.html`.

Some distributions can still be installed via anonymous FTP from various Linux archive sites, but in many cases, the size of the distribution makes this impractical. (``*Where Are the Linux FTP Archives?*'') There are also a large number of releases which are distributed less globally that suit special local and national needs. Many of them are archived at `ftp://ftp.tux.org/`

1.3. What Software does Linux Support?

All of the standard open source utilities, like GCC, (X)Emacs, the X Window System, all the standard Unix utilities, TCP/IP (including SLIP and PPP), and all of the hundreds of programs that people have compiled or ported to it.

There is a DOS emulator, called DOSEMU. The latest stable release is 0.98.3. The FTP archives are at `ftp://ftp.dosemu.org/dosemu` The Web site is `http://www.dosemu.org`.

The emulator can run DOS itself and some (but not all) DOS applications. Be sure to look at the `README` file to determine which version you should get. Also, see the *DOSEMU−HOWTO* (slightly dated at this point it doesn't cover the most recent version of the program), at
`ftp://metalab.unc.edu/pub/Linux/docs/HOWTO`.

Work has been progressing on an emulator for Microsoft Windows binaries. (``*Can Linux Run Microsoft Windows Programs?*'')

iBCS2 (Intel Binary Compatibility Standard) emulator code for SVR4 ELF and SVR3.2 COFF binaries can be included in the kernel as a compile−time option. There is information at
`ftp://tsx-11.mit.edu/pub/linux/BETA/ibcs2/README`.

For more information see the *INFO−SHEET*, which is one of the HOWTO's (``*Where Is the Documentation?*'' and ``*How To Port XXX to Linux.*'')

Some companies have commercial software available. They often announce their availability on *comp.os.linux.announce* try searching the archives. (``*Are the News Groups Archived Anywhere?*'')

1.4. How to Find a Particular Application.

Look first in the Linux Software Map. It's at:
`ftp://ibiblio.org/pub/Linux/docs/linux-software-map/`, and on the other FTP sites. A search engine is available on the World Wide Web at `http://www.boutell.com/lsm/`.

Also check out the *Freshmeat* Web site: `http://www.freshmeat.net`, which is where many new announcements of free software first appear. Freshmeat is basically a site index that continuously updates the notices of new or upgraded software for Linux, and maintains indexes of the announcements and links to

their URL's.

The FTP sites (``*Where Are the Linux FTP Archives?*'') often have `ls-lR` or `INDEX` directory listings which you can search using **grep** or a text editor. The directory listings files can be very large, however, making them unwieldy for quick searches.

Also look at the Linux Projects Map:
`ftp://ftp.ix.de/pub/ix/Linux/docs/Projects-Map.gz`.

There's a search engine for Linux FTP archives at: `http://lfw.linuxhq.com/`.

Searching for ``Linux'' on the World Wide Web provides copious references. (``*Where Is the Linux Stuff on the World Wide Web?*'')

If you don't find anything, you could download the sources to the program yourself and compile them. See (See: ``*How To Port XXX to Linux.*'') If it's a large package that may require some porting, post a message to *comp.os.linux.development.apps*.

If you compile a large-ish program, please upload it to one or more of the FTP sites, and post a message to *comp.os.linux.announce* (submit your posting to `linux-announce@news.ornl.gov`).

If you're looking for an application program, the chances are that someone has already written a free version. The *comp.sources.wanted* FAQ has instructions for finding the source code.

1.5. What Hardware Is Supported?

A minimal Linux installation requires a machine for which a port exists, at least 2Mb of RAM, and a single floppy drive. But to do anything even remotely useful, more RAM and disk space are needed. Refer to: ``*Ports to Other Processors.*'', ``*Disk Space Requirements: Minimal, Server, and Workstation.*'', and ``*Minimum and Maximum Memory Requirements.*''

Intel CPU, PC-compatible machines require at least an 80386 processor to run the standard Linux kernel.

Linux, including the X Window System GUI, runs on most current laptops. Refer to the answer for: ``*How To Find Out If a Notebook Runs Linux.*'' There are numerous sources of information about specific PC's, video cards, disk controllers, and other hardware. Refer to the *INFO-SHEET*, *Laptop HOWTO*, and the *Hardware HOWTO*. (``*Where Is the Documentation?*'')

1.6. Ports to Other Processors.

The Web site, Overview of Linux Ports:
`http://www.itp.uni-hannover.de/~kreutzm/de/lin_plattforms.html` provides a listing of known ports.

Another site with a list of ports is:
`http://lodda.igo.uni-hannover.de/ports/linux_ports.html`

In addition, the following information is available about specific ports:

On Intel platforms, VESA Local Bus and PCI bus are supported.

MCA (IBM's proprietary bus) and ESDI hard drives are mostly supported. There is further information on the MCA bus and what cards Linux supports on the Micro Channel Linux Web page, `http://www.dgmicro.com/mca`. Refer also to the answer for: ``*Where Is the Linux Stuff on the World Wide Web?*''

There is a port of Linux to the 8086, known as the Embeddable Linux Kernel Subset (ELKS). This is a 16−bit subset of the Linux kernel which will mainly be used for embedded systems, at: `http://www.linux.org.uk/Linux8086.html`. Standard Linux does not run 8086 or 80286 processors, because it requires task−switching and memory management facilities found on 80386 and later processors.

Linux supports multiprocessing with Intel MP architecture. See the file `Documentation/smp.tex` in the Linux kernel source code distribution.

A project has been underway for a while to port Linux to suitable 68000−series based systems like Amigas and Ataris. The Linux/m68K FAQ is located at `http://www.clark.net/pub/lawrencc/linux/faq/faq.html`. The URL of the Linux/m68k home page is `http://www.linux−m68k.org/faq/faq.html`.

There is a m68k port for the Amiga by Jes Sorensen, which is located at `ftp://sunsite.auc.dk/pub/os/linux/680x0/redhat/`. The installation FAQ for the package, by Ron Flory, is at `http://www.feist.com/~rjflory/linux/rh/`.

There is also a linux−680x0 mailing list. (``*What Mailing Lists Are There?*'')

There is (or was) a FTP site for the Linux−m68k project on `ftp.phil.uni−sb.de/pub/atari/linux−68k`, but this address may no longer be current.

Debian GNU/Linux is being ported to Alpha, Sparc, PowerPC, and ARM platforms. There are mailing lists for all of them. See `http://www.debian.org/MailingLists/subscribe`

One of the Linux−PPC project pages has moved recently. Its location is `http://www.debian.org/MailingLists/subscribe`. `http://www.linuxppc.org`, and the archive site is `ftp://ftp.linuxppc.org/linuxppc`.

There is a Linux−PPC support page at `http://www.cs.nmt.edu/~linuxppc/`. There you will find the kernel that is distributed with Linux.

There are two sites for the Linux iMac port: `http://w3.one.net/~johnb/imaclinux`, and `http://www.imaclinux.net:8080/content/index.html`.

A port to the 64−bit DEC Alpha/AXP is at `http://www.azstarnet.com/~axplinux/`. There is a mailing list at `vger.redhat.com`: (``*What Mailing Lists Are There?*'')

Ralf Baechle is working on a port to the MIPS, initially for the R4600 on Deskstation Tyne machines. The *Linux−MIPS* FTP sites are `ftp://ftp.fnet.fr/linux−mips` and `ftp://ftp.linux.sgi.com/pub/mips−linux`. Interested people may mail their questions and

offers of assistance to linux@waldorf-gmbh.de.

There is (or was) also a MIPS channel on the Linux Activists mail server and a linux−mips mailing list. (``*What Mailing Lists Are There?*'')

There are currently two ports of Linux to the ARM family of processors. One of these is for the ARM3, fitted to the Acorn A5000, and it includes I/O drivers for the 82710/11 as appropriate. The other is to the ARM610 of the Acorn RISC PC. The RISC PC port is currently in its early to middle stages, owing to the need to rewrite much of the memory handling. The A5000 port is in restricted beta testing. A release is likely soon.

For more, up−to−date information, read the newsgroup *comp.sys.acorn.misc*. There is a FAQ at http://www.arm.uk.linux.org.

The Linux SPARC project is a hotbed of activity. There is a FAQ and plenty of other information available from the UltraLinux page, http://www.ultralinux.org/.

The Home Page of the UltraSPARC port ("UltraPenguin") is located at http://sunsite.mff.cuni.cz/linux/ultrapenguin−1.0/, although the URL may not be current.

There is also a port to SGI/Indy machines ("Hardhat"). The URL is http://www.linux.sgi.com/.

1.7. Disk Space Requirements: Minimal, Server, and Workstation.

About 10Mb for a very minimal installation, suitable for trying Linux, and not much else.

You can fit a typical server installation, including the X Window Systemt GUI, into 80Mb. Installing Debian GNU/Linux takes 500Mb−1GB, including kernel source code, some space for user files, and spool areas.

Installing a commercial distribution that has a desktop GUI environment, commercial word processor, and front−office productivity suite, will claim 1−1.5 GB of disk space, approximately.

1.8. Minimum and Maximum Memory Requirements.

At least 4MB, and then you will need to use special installation procedures until the disk swap space is installed. Linux will run comfortably in 4MB of RAM, although running GUI apps is impractically slow because they need to swap out to disk.

Some applications, like StarOffice, require 32 MB of physical memory, and compiling C++ code can easily consume over 100 MB of combined physical and virtual memory.

There is a distribution, ``Small Linux,'' that will run on machines with 2MB of RAM. Refer to the answer to: ``*Where Are the Linux FTP Archives?*''

A number of people have asked how to address more than 64 MB of memory, which is the default upper limit in most standard kernels. Either type, at the BOOT lilo: prompt:

```
mem=XXM
```

Or place the following in your `/etc/lilo.conf` file:

```
append="mem=XXM"
```

The parameter ``*XX*M" is the amount of memory, specified as megabytes; for example, ``128M."

If an ``append=" directive with other configuration options already exists in `/etc/lilo.conf`, then add the `mem=` directive to the end of the existing argument, and separated from the previous arguments by a space; e.g.:

```
# Example only; do not use.
append="parport=0x3bc,none serial=0x3f8,4 mem=XXM"
```

Be sure to run the ``**lilo**" command to install the new configuration.

If Linux still doesn't recognize the extra memory, the kernel may need additional configuration. Refer to the `/usr/src/linux/Documentation/memory.txt` file in the kernel source as a start.

For further information about LILO, refer to the manual pages for **lilo** and `lilo.conf`, the documentation in `/usr/doc/lilo`, and the answer for: ``*Boot−Time Configuration.*", below.

1.9. Does Linux Support Universal System Bus Devices?

Linux supports a few dozen USB devices at present, and work is underway to develop additional device drivers. There is a Web page devoted to the subject, at `http://www.linux-usb.org/`. There is also LDP documentation, at: (``*Where Is the Linux Stuff on the World Wide Web?*")

1.10. What Is Linux's Open−Source License?

The Linux trademark belongs to Linus Torvalds. He has placed the Linux kernel under the GNU General Public License, which basically means that you may freely copy, change, and distribute it, but you may not impose any restrictions on further distribution, and you must make the source code available.

There is a FAQ for the GPL at: `http://www.gnu.org/copyleft/gnu-faq.html`.

This is not the same as Public Domain. See the Copyright FAQ, `ftp://rtfm.mit.edu/pub/usenet/news.answers/law/copyright`, for details.

Full details are in the file `COPYING` in the Linux kernel sources (probably in `/usr/src/linux` on your system).

The licenses of the utilities and programs which come with the installations vary. Much of the code is from the GNU Project at the Free Software Foundation, and is also under the GPL.

Note that discussion about the merits or otherwise of the GPL should be posted to the news group gnu.misc.discuss, and not to the *comp.os.linux* hierarchy.

For legal questions, refer to the answer: (``*Where Are Linux Legal Issues Discussed?*'')

1.11. Is Linux *nix?

Not officially, until it passes the Open Group's certification tests, and supports the necessary API's. Even very few of the commercial operating systems have passed the Open Group tests. For more information, see `http://www.unix-systems.org/what_is_unix.html`.

[Bob Friesenhahn]

2. Network Sources and Resources

2.1. Where Is the Latest Kernel Version on the Internet?

Make that versions. The 2.0 series kernels are still available for older machines. The latest production kernel series is 2.2.x. The updates to this kernel are bug fixes. The new 2.4 kernel sources are also on−line.

The Web page at `http://www.kernel.org/` lists the current versions of the development and production kernels.

If you want to download the source code, FTP to `ftp.xx.kernel.org`, where ``xx'' is the two−letter Internet domain abbreviation of your country; e.g., ``us'' for United States, ``ca'' for Canada, or ``de'' for Germany. Kernel versions 2.2.x are archived in the directory `pub/linux/kernel/v2.2`, as are patches for the prerelease versions. The kernel source code is archived as a `.tar.gz` file, and as a `.tar.bz2` file.

Follow the instructions in any of the standard references to compile the kernel, as you would with any other custom kernel. The `Documentation` subdirectory contains information by the authors of various subsystems and drivers, and much of that information is not documented elsewhere.

If you want to participate in kernel development, make sure that you sign on to the linux−kernel mailing list to find out what people are working on. Refer to the answer: ``*What Mailing Lists Are There?*''

There is a story about the features of the 2.4 series kernels at `http://features.linuxtoday.com/stories/8191.html`.

2.2. Where Is the Documentation?

Look in the following places, and the sites that mirror them.

- `http://www.linuxdoc.org/`
- `ftp://ftp.funet.fi/pub/OS/Linux/doc/HOWTO/`
- `ftp://tsx−11.mit.edu/pub/linux/docs/HOWTO/`
- `ftp://metalab.unc.edu/pub/Linux/docs/HOWTO/`

For a list of Linux FTP sites, refer to the answer for: ``*Where Are the Linux FTP Archives?*''

If you don't have access to FTP, try the FTP−by−mail servers: `ftpmail@decwrl.dec.com`, `ftpmail@doc.ic.ac.uk`, or: `ftp−mailer@informatik.tu−muenchen.de`.

A complete list of HOWTO's is available in the file *HOWTO−INDEX* at `http://www.linuxdoc.org/HOWTO/HOWTO−INDEX/howtos.html`. The mini−HOWTO's are indexed at `http://www.linuxdoc.org/HOWTO/HOWTO−INDEX/mini.html`.

A search engine at the Linux FAQ Home Page, `http://www.mainmatter.com/`, allows you to search LDP HOWTO's, the Linux FAQ, man pages, and *Network Administrator's Guide*.

In addition, translations are available from `ftp://metalab.unc.edu/pub/Linux/docs/HOWTO/translations/` and mirrors worldwide.

The HOWTO's and other documentation have been translated into the following languages:

Chinese (Big5) (zh)	Croatian (hr)	French (fr)
German (de	Hellenic (el)	Indonesian (id)
Italian (it)	Japanese (ja)	Korean (ko)
Polish (pl)	Slovenian (sl)	Spanish (es)
Swedish (sv)	Turkish (tr)	

Additional documents are always in preparation. Please get in touch with the coordinators if you are interested in writing one. Contact and submission information is at `http://www.linuxdoc.org/mailinfo.html`.

There is also a LDP HOWTO page at `http://howto.tucows.org/`.

The Guide Series produced by the Linux Documentation Project is available from `http://www.linuxdoc.org/`. Please read them if you are new to Unix and Linux.

The *Linux Mobile Guide* is an expanded version of the *Linux−Laptop−HOWTO*. The URL is: `http://home.snafu.de/wehe/howtos.html`.

And, of course, a number of people have written documentation independently of the LDP:

- *Linux Administrators Security Guide,* by Kurt Seifried. `http://www.freek.com/lasg/`.
- *Newbie's Linux Manual.* `http://www.linuxdoc.org/nlm/`.
- *One−Page Linux Manual.* `http://www.powerup.com.au/~squadron/`.
- *Rute Users Tutorial and Exposition.* `http://rute.sourceforge.net/`.
- *Short beginners' manual for Linux.* Also available in Dutch. `http://www.stuwww.kub.nl/people/b.vannunen/linux−man.php3`.
- *Virtual Frame buffer HOWTO,* by Alex Buell. `http://www.tahallah.demon.co.uk/programming/prog.html`.
- *X11 & TrueType Fonts,* by Peter Kleiweg. `http://www.let.rug.nl/~kleiweg/`.

Documentation for kernel developers is on−line: `http://kernelbook.sourceforge.net/`.

To find out about Linux memory management, including performance tuning, see Rik van Riel's Web page at `http://humbolt.geo.uu.nl/Linux−MM/`.

The Linux Consultants HOWTO has a directory of Linux consultants at `http://www.linuxports.com/`.

Gary's Encyclopedia lists over 4,000 Linux related links. Its URL is `http://members.aa.net/~swear/pedia/index.html`.

There is also a FAQ specifically for the Red Hat Linux distribution, at `http://www.best.com/~aturner/RedHat−FAQ/faq_index.html`.

And the Home Page of this FAQ is `http://www.mainmatter.com/`.

2.3. Where Is the Linux Stuff on the World Wide Web?

In addition to the Linux Documentation Project Home Page: `http://www.linuxdoc.org/`, there are many pages that provide beginning and advanced information about Linux.

These two pages provide a good starting point for general Linux information: Linux International's Home Page, at `http://www.li.org/`, and the Linux Online's Linux Home Page at `http://www.linux.org/`.

Both of these pages provide links to other sites, information about general information, distributions, new software, documentation, and news.

Documentation for kernel developers is on–line: `http://kernelbook.sourceforge.net/`.

The tutorial, *Unix is a Four Letter Word...,* is located at `http://www.linuxbox.com/~taylor/4ltrwrd/`. It is a general introduction to Unix operating systems and is not Linux specific.

Additionally, here is a certainly incomplete list of Web pages devoted to Linux:

- AboutLinux.com: `http://www.aboutlinux.com/`.
- Adventures in Linux Programming: `http://members.tripod.com/rpragana/`.
- Dave Central Linux Software Archive: `http://linux.davecentral.com/`.
- debianHELP `http://www.debianhelp.org/`.
- Erlug Webzine (Italian): `http://www.erlug.linux.it/`.
- Free Unix Giveaway List: `http://visar.csustan.edu/giveaway.html`. Lists offers of free Linux CDs. Also available via E–mail: `axel@visar.csustan.edu`, with the Subject: send giveaway_list.
- Information on Linux in corporate environments: `http://www.smartstocks.com/linux.html`.
- Jeanette Russo's Linux Newbie Information: `http://www.stormloader.com/jrusso2/index.html`.
- JustLinux.com: `http://www.justlinux.com/`.
- Linux Cartoons: `http://www.cse.unsw.edu.au/~conradp/linux/cartoons/`.
- LinuxArtist.org: `http://www.linuxartist.org/`.
- Linuxinfor.com – Online Linux Resources: `http://www.linuxinfor.com/`.
- linuXChiX.org: `http://www.linuxchix.org/`.
- LinuxDevices.com: The Embedded Linux Portal: `http://www.linuxdevices.com`.
- Linux Educational Needs Posting Page: `http://www.slip.net/~brk/linuxedpp.htm`.
- Linux in Business: Case Studies: `http://www.bynari.com/collateral/case_studies.html`.
- Linux Hardware Database Laptop Superguide: `http://lhd.zdnet.com/db/superguide.php3?catid=18`.
- Linux Inside: `http://linuxinside.org/`.
- Linux Links: `http://www.linuxlinks.com/`.
- Linux Memory Management Home Page: `http://humbolt.geo.uu.nl/Linux-MM/`.
- Linux Newbie Project: `http://kusma.hypermart.net/`.
- Linux on the Thinkpad 760ED: `http://www.e-oasis.com/linux-tp.html`.
- LinuxOrbit: `http://www.linuxorbit.com/`
- Linux Parallel Port Home Page: `http://www.torque.net/linux-pp.html`.

- Linux MIDI & Sound Applications: `http://sound.condorow.net/`.
- Linux Start: `http://www.linuxstart.com/`.
- Linux Tips and Tricks Page: `http://www.patoche.org/LTT/`.
- Linux Today PR: `http://www.linuxpr.com/`.
- Mandrakeuser.Org: `http://mandrakeuser.org/`
- My Linux Contributions by Richard Gooch: `http://www.atnf.csiro.au/~rgooch/linux/`.
- Micro Channel Linux Web Page: `http://www.dgmicro.com/mca/`.
- Parallel port scanners and SANE: `http://www2.prestel.co.uk/hex/scanners.html`.
- Pascal Central: `http://www.pascal-central.com/`
- PegaSoft Portal: `http://www.vaxxine.com/pegasoft/portal/`
- PocketLinux. `http://www.pocketlinux.com/`.
- Red Hat and ISDN4Linux: `http://www.webideal.de/`.
- SearchLinux: `http://www.searchlinux.com/`.
- The Free Linux CD Project: `http://www.freelinuxcd.org/`.
- The Site for People Learning Perl: `http://learn.perl.org/`.
- USB Linux Home Page: `http://peloncho.fis.ucm.es/~inaky/uusbd-www/`.
- VLUG: The Virtual Linux Users Group: `http://www.vlug.com/`.

Searching for ``Linux'' on Web Search Engines, like Yahoo! (`http://www.yahoo.com/`), Altavista (`http://www.altavista.com/`), or Google (`http://www.google.com/`) will provide copious references to Linux Web sites. Further information about about Web search engines is in the *Web and Internet Search Engine Faq*: `http://www.infobasic.com/pagefaq.html`.

Refer also to the answer for: ``*What Other FAQ's and Documentation Are There for Linux?*''

2.4. What News Groups Are There for Linux?

Comp.os.linux.announce is the moderated announcements group. You should read this if you intend to use Linux. It contains information about software updates, new ports, user group meetings, and commercial products. It is the *only* newsgroup that may carry commercial postings. Submissions for that group should be e–mailed to `linux-announce@news.ornl.gov`.

Comp.os.linux.announce is archived at: `http://www.iki.fi/mjr/linux/cola.html`, and `ftp://src.doc.ic.ac.uk/usenet/comp.os.linux.announce/`.

Also worth reading are the following other groups in the *comp.os.linux.** and *alt.uu.comp.os.linux.** hierarchies you may find many common problems too recent for the documentation but are answered in the newsgroups.

- *alt.uu.comp.os.linux*
- *alt.uu.comp.os.linux.questions*
- *alt.os.linux*
- *alt.os.linux.mandrake*
- *comp.os.linux.admin*
- *comp.os.linux.advocacy*
- *comp.os.linux.alpha*
- *comp.os.linux.answers*
- *comp.os.linux.development*

- *comp.os.linux.development.apps*
- *comp.os.linux.development.system*
- *comp.os.linux.embedded*
- *comp.os.linux.hardware*
- *comp.os.linux.help*
- *comp.os.linux.m68k*
- *comp.os.linux.misc*
- *comp.os.linux.network*
- *comp.os.linux.networking*
- *comp.os.linux.portable*
- *comp.os.linux.powerpc*
- *comp.os.linux.questions*
- *comp.os.linux.redhat*
- *comp.os.linux.security*
- *comp.os.linux.setup*
- *comp.os.linux.test*
- *comp.os.linux.x*
- *comp.os.linux.x.video*

Remember that Linux is POSIX compatible, and most all of the material in the *comp.unix.** and *comp.windows.x.** groups will be relevant. Apart from hardware considerations, and some obscure or very technical low−level issues, you'll find that these groups are good places to start.

Information about e−mail clients (MUA's), mail transfer agents (MTA's), and other related software are in the `comp.mail.*` groups, especially:

- *comp.mail.misc*
- *comp.mail.pine*
- *comp.mail.sendmail*

Questions and information about News reading software are in: *news.software.readers*.

Please read ``*If this Document Still Hasn't Answered Your Question....*" before posting. Cross posting between different *comp.os.linux.** groups is rarely a good idea.

There may well be Linux groups local to your institution or area check there first.

See also ``*How To Get Information without Usenet Access.*"

Other regional and local newsgroups also exist you may find the traffic more manageable there. The French Linux newsgroup is *fr.comp.os.linux*. In Germany there is *de.comp.os.linux.**. In Australia, try *aus.computers.linux*. In Croatia there is *hr.comp.linux*. In Italy, there is *it.comp.linux*.

A search of `http://groups.google.com/` can provide an up−to−date list of News groups.

[Axel Boldt, Robert Kiesling]

2.5. What Other FAQ's and Documentation Are There for Linux?

There are a number of special interest FAQ's on different subjects related to system administration and use, and also on miscellaneous topics like Flying Saucer Attacks (the music) and support for recovering sysadmins.

The official Usenet FAQ archives are: `ftp://rtfm.mit.edu/pub/usenet/`.

The *Internet FAQ Consortium* provides a searchable archive at: `http://www.faqs.org/`. The site also maintains a current archive of Internet Request For Comment (RFC), Best Current Practices (BCP), and For Your Information (FYI) documents.

Here are some FAQ's and documents that might be especially useful, and their network addresses:

- A FAQ for new users: `http://homes.arealcity.com/swietanowski/LinuxFAQ/`.
- AfterStep FAQ: `http://www.linuxinfor.com/en/astepfaq/AfterStep-FAQ.html`.
- BASH Frequently Asked Questions: `ftp://ftp.cwru.edu/pug/bash/FAQ/`.
- `de.comp.os.unix.linux.infos - FAQ`: `http://www.dcoul.de/`.
- Frequently Asked Questions about Open Source: `http://www.opensource.org/faq.html`.
- Ftape–FAQ: `http://www.linuxinfor.com/en/ftapefaq/Ftape-FAQ.html`.
- GNU Emacs: `http://www.lerner.co.il/emacs/faq-body.shtml`.
- GNU Linux in Science and Engineering: `http://www.comsoc.org/vancouver/scieng.html`.
- GNU Troff (*groff*) Info: `http://www.cs.pdx.edu/~trent/gnu/groff/`.
- Gnus 5.x: `http://www.ccs.neu.edu/software/contrib/gnus/`.
- KDE FAQ: `http://www.kde.org/faq.html`.
- GNU General Public License FAQ: `http://www.gnu.org/copyleft/gpl-faq.html`.
- Linux PPP FAQ: `http://www.linuxinfor.com/en/pppfaq/PPP-FAQ.html`.
- Linux–Raid FAQ: `http://www.linuxinfor.com/en/raidfaq/index.html`.
- List of Periodic Information Postings: `ftp://rtfm.mit.edu/pub/usenet/news.answers/periodic-postings/`.
- News.newusers.announce FAQ `http://web.presby.edu/~nnqadmin/nan/`.
- Online Linux Resources: `http://www.linuxinfor.com/en/docfaq.htm`.
- O'Reilly & Associates Openbook Project: `http://www.oreilly.com/openbook/`.
- Sendmail: `http://www.sendmail.org/faq/`.
- *Sendmail: Installation and Operation Guide:* Formatted and *me* source versions are in the `doc/` subdirectory of Sendmail source code distributions. `http://www.sendmail.org/`.
- Technical FAQ for Linux Users: `http://www-106.ibm.com/developerworks/library/l-faq/?n-l-7261`.
- Web Internet Search Engine: `http://www.infobasic.com/pagefaq.html`
- Wu–ftpd: `http://www.wu-ftpd.org/man/` (really a collection of man pages), with HOWTO's at: `http://www.wu-ftpd.org/HOWTO/`
- XTERM Frequently Asked Questions. `http://dickey.his.com/xterm/xterm.faq.html`.

2.6. Where Are the Linux FTP Archives?

There are three main archive sites for Linux:

- `ftp://ftp.funet.fi/pub/OS/Linux/` (Finland).
- `ftp://metalab.unc.edu/pub/Linux/`. Recently renamed to `http://ibiblio.org/pub/linux/` with a nice WWW interface. (US).
- `ftp://tsx-11.mit.edu/pub/linux/` (US).

The best place to get the Linux kernel is `ftp://ftp.kernel.org/pub/linux/kernel/`. Linus Torvalds uploads the most recent kernel versions to this site.

Of the U.S. distributions, Debian GNU/Linux is available at `ftp://ftp.debian.org/pub/debian/`. Red Hat Linux's home site is `ftp://ftp.redhat.com/`, and Linux Slackware's is `ftp://ftp.freesoftware.com/`.

The Small Linux distribution, which can run in 2 MB of RAM, is located at `http://smalllinux.netpedia.net/`.

The contents of these sites is mirrored (copied, usually approximately daily) by a number of other sites. Please use a site close to you it will be faster for you and easier on the network.

- `ftp://ftp.sun.ac.za/pub/linux/sunsite/` (South Africa)
- `ftp://ftp.is.co.za/linux/sunsite/` (South Africa).
- `ftp://ftp.cs.cuhk.hk/pub/Linux/` (Hong Kong).
- `ftp://sunsite.ust.hk/pub/Linux/` (Hong Kong).
- `ftp://ftp.spin.ad.jp/pub/linux/` (Japan).
- `ftp://ftp.nuri.net/pub/Linux/` (Korea).
- `ftp://ftp.jaring.my/pub/Linux/` (Malaysia).
- `ftp://ftp.nus.sg/pub/unix/Linux/` (Singapore).
- `ftp://ftp.nectec.or.th/pub/mirrors/linux/` (Thailand).
- `ftp://planetmirror.com/pub/linux` (Australia). (Also take a look at `http://planetmirror.com/archives.php`.)
- `ftp://ftp.monash.edu.au/pub/linux/` (Australia).
- `ftp://ftp.univie.ac.at/systems/linux/sunsite/` (Austria).
- `ftp://ftp.fi.muni.cz/pub/UNIX/linux/` (Czech Republic).
- `ftp://ftp.funet.fi/pub/Linux/sunsite/` (Finland).
- `ftp://ftp.univ-angers.fr/pub/Linux/` (France).
- `ftp://ftp.iut-bm.univ-fcomte.fr/` (France).
- `ftp://ftp.ibp.fr/pub/linux/sunsite/` (France)
- `ftp://ftp.loria.fr/pub/linux/sunsite/` (France).
- `ftp://ftp.dfv.rwth-aachen.de/pub/linux/sunsite/` (Germany).
- `ftp://ftp.germany.eu.net/pub/os/Linux/Mirror.SunSITE/`(Germany).
- `ftp://ftp.tu-dresden.de/pub/Linux/sunsite/` (Germany).
- `ftp://ftp.uni-erlangen.de/pub/Linux/MIRROR.sunsite/` (Germany).
- `ftp://ftp.gwdg.de/pub/linux/mirrors/sunsite/` (Germany).
- `ftp.rz.uni-karlsruhe.de/pub/linux/mirror.sunsite/` (Germany).
- `ftp://ftp.ba-mannheim.de/pub/linux/mirror.sunsite/` (Germany).
- `ftp://ftp.uni-paderborn.de/pub/Mirrors/sunsite.unc.edu/`(Germany).
- `ftp://ftp.uni-rostock.de/Linux/sunsite/` (Germany).

- ftp.rus.uni-stuttgart.de/pub/unix/systems/linux/MIRROR.sunsite/(Germany).
- ftp://ftp.uni-tuebingen.de/pub/linux/Mirror.sunsite/ (Germany).
- ftp://ftp.kfki.hu/pub/linux/(Hungary).
- ftp://linux.italnet.it/pub/Linux/(Italy).
- ftp://ftp.unina.it/pub/linux/sunsite/ (Italy).
- ftp://giotto.unipd.it/pub/unix/Linux/ (Italy).
- ftp://cnuce-arch.cnr.it/pub/Linux/ (Italy).
- ftp://ftp.flashnet.it/mirror2/metalab.unc.edu/ (Italy).
- ftp://ftp.nijenrode.nl/pub/linux/ (Netherlands).
- ftp://ftp.LeidenUniv.nl/pub/linux/sunsite/ (Netherlands).
- ftp://ftp.nvg.unit.no/pub/linux/sunsite/ (Norway).
- ftp://sunsite.icm.edu.pl/pub/Linux/metalab.unc.edu/ (Poland).
- ftp://ftp.rediris.es/software/os/linux/sunsite/ (Spain).
- ftp://sunsite.rediris.es/software/linux/ (Spain).
- ftp://ftp.cs.us.es/pub/Linux/sunsite-mirror/ (Spain).
- ftp://ftp.etse.urv.es/pub/mirror/linux/ (Spain).
- ftp://tp.etsimo.uniovi.es/pub/linux/ (Spain).
- ftp://ftp.luna.gui.es/pub/linux.new/ (Spain).
- ftp://ftp.metu.edu.tr/pub/linux/sunsite/ (Turkey).
- ftp://unix.hensa.ac.uk/mirrors/sunsite/pub/Linux/ (UK).
- ftp.maths.warwick.ac.uk/mirrors/linux/sunsite.unc-mirror/(UK).
- ftp://ftp.idiscover.co.uk/pub/Linux/sunsite.unc-mirror/(UK).
- ftp://sunsite.doc.ic.ac.uk/packages/linux/sunsite.unc-mirror/(UK).
- (UK)
- ftp://ftp.io.org/pub/mirrors/linux/sunsite/ (Canada).
- ftp://ftp.cc.gatech.edu/pub/linux/ (US).
- ftp://ftp.freesoftware.com/pub/linux/sunsite/ (US).
- ftp://ftp.siriuscc.com/pub/Linux/Sunsite/ (US).
- ftp://ftp.engr.uark.edu/pub/linux/sunsite/ (US).
- ftp://ftp.infomagic.com/pub/mirrors/linux/sunsite/ (US).
- ftp://linux.if.usp.br/pub/mirror/metalab.unc.edu/pub/Linux/(Brazil).
- ftp://farofa.ime.usp.br/pub/linux/ (Brazil).

Please send updates and corrections to this list to the Linux FAQ maintainer, rkiesling@mainmatter.com. Not all of these mirror all of the other ``source'' sites, and some have material not available on the ``source'' sites.

2.7. How To Get Linux without FTP Access.

The easiest thing is probably to find a friend with FTP access. If there is a Linux user's group near you, they may be able to help.

If you have a reasonably good email connection, you could try the FTP−by−mail servers at ftpmail@ftp.sunet.se, or ftpmail@ftp.uni-stuttgart.de.

Linux is also available via traditional mail on CD−ROM. The file ftp://metalab.unc.edu/pub/Linux/docs/HOWTO/Installation-HOWTO, and the file ftp://metalab.unc.edu/pub/Linux/docs/HOWTO/Distribution-HOWTO contain information on these distributions.

2.8. How To Get Information without Usenet Access.

A digest of comp.os.linux.announce is available by mailing the word "subscribe" (without the quotes) as the body of a message to `linux-announce-REQUEST@news-digests.mit.edu`. Subscribing to this list is a good idea, as it carries important information and documentation about Linux.

Please remember to use the *-request addresses for your subscribe and unsubscribe messages; mail to the other address is posted to the news group.

2.9. What Mailing Lists Are There?

The Linux developers now mainly use the Majordomo server at `majordomo@vger.redhat.com`. Send a message with the word "lists" (without the quotes) in the body to get a list of lists there. Add a line with the word, "help," to get the standard Majordomo help file that lists instructions for subscribing and unsubscribing to the lists.

Currently, the kernel list is archived at:
`http://www.uwsg.indiana.edu/hypermail/linux/kernel/`, and
`http://www.lib.uaa.alaska.edu/linux-kernel/archive/`

Please do not post off-topic material to the mailing lists. Most of them are used by Linux developers to talk about technical issues and future developments. They are not intended for new users' questions, advertisements, or public postings that are not directly related to the mailing list's subject matter. *Comp.os.linux.announce* is the place for all public announcements. This is a common Internet policy. If you don't observe this guideline, there's a good chance that you'll be flamed.

There is a linux-newbie list where, "no question is too stupid." Unfortunately, it seems that few experienced users read that list, and it has very low volume.

There are numerous Linux related mailing lists at `http://www.onelist.com/`. Go to the categories page and choose "Linux." There are also mailing list subscription links at:
`http://oslab.snu.ac.kr/~djshin/linux/mail-list/`.

The *Mailing Lists Available in Usenet* page is: `http://paml.net/`. The list information is also on: `ftp://rtfm.mit.edu/`, and is posted to the groups: `news.announce.newgroups`, `news.lists`, and `news.groups`, among others.

2.10. Where Are Linux Legal Issues Discussed?

On the linux-legal mailing list, of course. You can subscribe to it, as with many of the other Linux related lists, by sending a message with the word "help" in the body of the message to `majordomo@vger.redhat.com`.

2.11. Sources of Information for Unmaintained Free Software Projects.

There are Web pages at: `http://unmaintained.sourceforge.net`, and: `http://www.orphansource.org/`.

Please try to contact the original author(s) via e−mail, or the person who listed the software as unmaintained, before even thinking to place a license on the package.

2.12. Are the News Groups Archived Anywhere?

The Usenet Linux news groups are archived at `http://groups.google.com/`.

`ftp://metalab.unc.edu/pub/Linux/docs/linux-announce.archive` contains archives of *comp.os.linux.announce*. These are mirrored from `ftp://src.doc.ic.ac.uk/usenet/`, which also archives *comp.os.linux*, *comp.os.linux.development.apps*, and *comp.os.linux.development.system*.

2.13. Where To Find Information About Security Related Issues.

There's a page of Linux related security information at: `http://www.linuxsecurity.com/`.

Another site is: `http://www.rootshell.com/`, which has information about Internet security and privacy issues.

For information about the *Weekly Linux Security Digest* email newsletter and numerous security related databases, look at `http://securityportal.com/`.

2.14. Where To Find Linux System Specifications.

As a start, look at the Linux Standards Base, `http://www.linuxbase.org/`. The site contains information about test software, file system organization, and shared library naming conventions.

---

3. Compatibility with Other Operating Systems

3.1. Can Linux Use the Same Hard Drive as MS–DOS? OS/2? 386BSD? Win95?

Yes. Linux uses the standard MS–DOS partitioning scheme, so it can share your disk with other operating systems.

Linux has loadable kernel modules for (presumably) all versions of Microsoft FAT and VFAT file systems, including Windows 2000 and WindowsMe. In a correctly configured system, they should load automatically when the partitions are mounted.

Note, however, that many other operating systems may not be exactly compatible. DOS's `FDISK.EXE` and `FORMAT.EXE`, for example, can overwrite data in a Linux partition, because they sometimes incorrectly use partition data from the partition's boot sector rather than the partition table.

In order to prevent programs from doing this, it is a good idea to zero out under Linux the start of a partition you created, before you use MS–DOS or whatever to format it. Type:

```
$ dd if=/dev/zero of=/dev/hdXY bs=512 count=1
```

where *hdXY* is the relevant partition; e.g., `/dev/hda1` for the first partition of the first (IDE) disk.

Linux can read and write the files on your DOS and OS/2 FAT partitions and floppies using either the DOS file system type built into the kernel or mtools. There is kernel support for the VFAT file system used by Windows 9x and Windows NT.

There is reportedly a GPL'd OS/2 device driver that will read and write Linux ext2 partitions.

For information about FAT32 partition support, see `http://bmrc.berkeley.edu/people/chaffee/fat32.html`.

See, (``*What Software does Linux Support?*'') for details and status of the emulators for DOS, MS Windows, and System V programs.

See also, "Can Linux access Amiga file systems? ", "Can Linux access Macintosh file systems? ", "Can Linux access BSD, SysV, etc., UFS? ", and "Can Linux access SMB file systems? "

There are said to be NTFS drivers under development, which should support compression as a standard feature.

3.2. How To Access Files on a MS–DOS Partition or Floppy.

Use the DOS file system, type, for example:

```
$ mkdir /dos
$ mount -t msdos -o conv=text,umask=022,uid=100,gid=100 /dev/hda3 /dos
```

If it's a floppy, don't forget to umount it before ejecting it!

You can use the `conv=text/binary/auto`, `umask=nnn`, `uid=nnn`, and `gid=nnn` options to control the automatic line−ending conversion, permissions and ownerships of the files in the DOS file system as they appear under Linux. If you mount your DOS file system by putting it in your `/etc/fstab`, you can record the options (comma−separated) there, instead of defaults.

Alternatively, you can use mtools, available in both binary and source form on the FTP sites. (``*Where Are the Linux FTP Archives?*'')

A kernel patch (known as the fd−patches) is available which allows floppies with nonstandard numbers of tracks and/or sectors to be used; this patch is included in the 1.1 alpha testing kernel series.

3.3. Does Linux Support Compressed Ext2 File Systems?

The `ext2compr` project provides a kernel patch Information about them is located at `http://e2ompr.memalpha.cx/e2compr/`.

There is also a Web site for the e2compr patches. The code is still experimental and consists of patches for the 2.0 and 2.1 kernels. For more information about the project, including the latest patches, and the address of the mailing list, look up the URL at `http://debs.fuller.edu/e2compr/`.

[Roderich Schupp, Peter Moulder]

zlibc is a program that allows existing applications to read compressed (GNU gzip'ed) files as if they were not compressed. Look at `ftp://metalab.unc.edu/pub/Linux/libs/`. The author is Alain Knaff.

There is also a compressing block device driver, "DouBle," by Jean−Marc Verbavatz, which can provide on−the−fly disk compression in the kernel. The source−only distribution is located at `ftp://metalab.unc.edu/pub/Linux/patches/diskdrives/`. This driver compresses inodes and directory information as well as files, so any corruption of the file system is likely to be serious.

There is also a package called tcx (Transparently Compressed Executables), which allows you to keep infrequently used executables compressed and only uncompress them temporarily when in use. It is located at `ftp://metalab.unc.edu/pub/Linux/utils/compress/`.

3.4. Can Linux Use Stacked/DBLSPC/Etc. DOS Drives?

Until recently, not very easily. You can access DOS 6.X volumes from the DOS emulator ("What software does Linux support? "), but it's harder than accessing a normal DOS volume via the DOS kernel option, a module, or mtools.

There is a recently added package, dmsdos, that reads and writes compressed file systems like DoubleSpace/DriveSpace in MS−DOS 6.x and Win95, as well as Stacker versions 3 and 4. It is a loadable kernel module. Look at `ftp://metalab.unc.edu/pub/Linux/system/filesystems/dosfs/`.

3.5. Can Linux Access OS/2 HPFS Partitions?

Yes, but Linux access to HPFS partitions is read−only. HPFS file system access is available as an option when compiling the kernel or as a module. See the `Documentation/filesystems/hpfs.txt` file in the kernel source distribution. (``*How To Upgrade/Recompile a Kernel.*'') Then you can mount HPFS partition, using, for example:

```
$ mkdir /hpfs
$ mount -t hpfs /dev/hda5 /hpfs
```

3.6. Can Linux Access Amiga File Systems?

The Linux kernel has support for the Amiga Fast File System (AFFS) version 1.3 and later, both as a compile−time option and as a module. The file `Documentation/filesystems/affs.txt` in the Linux kernel source distribution has more information.

See (``*How To Upgrade/Recompile a Kernel.*'')

Linux supports AFFS hard−drive partitions only. Floppy access is not supported due to incompatibilities between Amiga floppy controllers and PC and workstation controllers. The AFFS driver can also mount disk partitions used by the Un*x Amiga Emulator, by Bernd Schmidt.

3.7. Can Linux Access BSD, SysV, Etc. UFS?

Recent kernels can mount (read only) the UFS file system used by System V; Coherent; Xenix; BSD; and derivatives like SunOS, FreeBSD, NetBSD, and NeXTStep. UFS support is available as a kernel compile−time option and a module.

See, (``*How To Upgrade/Recompile a Kernel.*'')

3.8. Can Linux Access SMB File Systems?

Linux supports read/write access of Windows for Workgroups and Windows NT SMB volumes. See the file Documentation/filesystems/smbfs.txt of the Linux kernel source distribution, and (``*How To Upgrade/Recompile a Kernel.*'')

There is also a suite of programs called Samba which provide support for WfW networked file systems (provided they're for TCP/IP). Information is available in the `README` file at `metalab.unc.edu/pub/Linux/system/network/samba/`.

The SMB Web site is `http://www.samba.org/`, and there is also a Web site at `samba.anu.edu.au/samba/`.

3.9. Can Linux Access Macintosh File Systems?

There is a set of user−level programs that read and write the older Macintosh Hierarchical File System (HFS). It is available at `metalab.unc.edu/pub/Linux/utils/disk-management/`.

Access to the newer, HFS+ file systems is still under development.

3.10. Can Linux Run Microsoft Windows Programs?

WINE, a MS Windows emulator for Linux, is still not ready for general distribution. If you want to contribute to its development, look for the status reports in the *comp.emulators.ms−windows.wine* newsgroup.

There is also a FAQ, compiled by P. David Gardner, at `ftp://metalab.unc.edu/pub/Linux/docs/faqs/Wine-FAQ/`.

In the meantime, if you need to run MS Windows programs, the best bet seriously is to reboot. LILO, the Linux boot loader, can boot one of several operating systems from a menu. See the LILO documentation for details.

Also, `LOADLIN.EXE` (a DOS program to load a Linux, or other OS, kernel is one way to make Linux co−exist with DOS. `LOADLIN.EXE` is particularly handy when you want to install Linux on a 3rd or 4th drive on a system (or when you're adding a SCSI drive to a system with an existing IDE).

In these cases, it is common for LILO's boot loader to be unable to find or load the kernel on the "other" drive. So you just create a `C:\LINUX` directory (or whatever), put `LOADLIN.EXE` in it with a copy of your kernel, and use that.

`LOADLIN.EXE` is a VCPI compliant program. Win95 will want to, "shutdown into DOS mode," to run it (as it would with certain other DOS protected−mode programs).

Earlier versions of `LOADLIN.EXE` sometimes required a package called `REALBIOS.COM`, which required a boot procedure on an (almost) blank floppy to map the interrupt vectors (prior to the loading of any software drivers). (Current versions don't seem to ship with it, and don't seem to need it).

[Jim Dennis]

3.11. Where Is Information about NFS Compatibility?

This information is partly taken from Nicolai Langfeldt's excellent NFS HOWTO, and is current as of 10/1/1999.

Most version 2.2.x kernels need a set of patches to install the **knfsd** subsystem, maintained by H.J. Lu, to communicate efficiently (if at all) with Sparc, IBM RS, and Alpha machines, and probably others. This package is actually a collection of patches to the kernel sources. Better support for non−Intel architectures is included in the 2.4 kernels.

There is also a user−space server. Although it lacks remote file locking, it is easier to install. It may be equally efficient.

In the `Documentation/Changes` of recent kernel distributions, there is a list of URL's for both the **knfsd** server and the user−space server.

There is a CVS server available for the kernel−space NFS subsystem, as well as a NFS WWW page at `http://www.linuxnfs.sourceforge.org/`, although the URL requires a password for access. The relevant URL's are listed in the `README.nfs` file at `ftp://ftp.us.kernel.org/`, and other kernel archive sites, along with login information. Patches are at `ftp://ftp.varesearch.com/pub/kernel/latest/patches/`.

The source archives of the user−space server and utilities currently reside on `ftp://linux.mathematik.tu-darmstadt.de:/pub/linux/people/okir/`.

In the case of older Solaris releases, the lack of `statd` or `lockd` on a client or server machine may cause incompatibility. On some versions of Solaris, `statd` can be used to exploit features of the automounter. Sun released a patch to correct this, but `statd` still needs to be started by root on such systems. On recent Solaris systems, refer to the information in `/etc/dfs/dfstab` and the `share(1M)` manual page to enable volume sharing. In addition, the `rpcinfo` program can tell you if `statd` or `lockd` are available on the local or remote machines.

The `linux-kernel` mailing list has on−and−off discussions of the status of the NFS subsystem, which appears to be changing rapidly.

[Nicolai Langfeldt, Robert Kiesling, Anders Hammarquist]

3.12. Can Linux Use True Type Fonts?

Yes. There are a number of True Type font servers for the X Window System. One of them is **xfsft**. Its home page is `http://www.dcs.ed.ac.uk/home/jec/programs/xfsft/`. There are also instructions for configuration.

People have reported success with other True Type font servers. There are links from the **xfsft** Home Page to them as well.

You can also compile True Type Font support into your X server directly. Again, refer to the **xfsft** Home Page for details.

3.13. Can Linux Boot from MS−DOS?

If LILO doesn't work, and if the machine has MS−DOS or Microsoft Windows, you may be left with a computer that won't boot. This can also happen on an upgrade to your Linux distribution. Re−installing LILO is the last thing that the installation does. So it is vitally important when installing or upgrading Linux on a dual boot machine, to have a MS−DOS or Windows rescue disk nearby so you can `FDISK /MBR`. Then you can go about using `LOADLIN.EXE` instead of LILO.

This `config.sys` file is one possible way to invoke `LOADLIN.EXE` and boot MS−DOS or Linux.

```
[menu]
menuitem=DOS, Dos Boot
menuitem=LINUX, Linux Boot

[LINUX]
shell=c:\redhat\loadlin.exe c:\redhat\autoboot\vmlinuz vga=5 root=/dev

[DOS]
STACKS = 0,0
rem all the other DOS drivers get loaded here.
```

This creates a menu where you can directly jump to `LOADLIN.EXE` before all of the MS−DOS drivers get loaded.

The paths and options are peculiar to one machine and should be intuitively obvious to the most casual observer. See the `LOADLIN.EXE` docs for options. They are the same as LILO, and options are just passed to the kernel, anyhow.

[Jim Harvey]

3.14. How Can Linux Boot from OS/2's Boot Manager?

1. Create a partition using OS/2's `FDISK.EXE` (Not Linux's **fdisk**).
2. Format the partition under OS/2, either with FAT or HPFS. This is so that OS/2 knows about the partition being formatted. (This step is not necessary with OS/2 `warp' 3.0.)
3. Add the partition to the Boot Manager.
4. Boot Linux, and create a file system on the partition using **mkfs −t ext2** or **mke2fs**. At this point you may, if you like, use Linux's **fdisk** to change the code of the new partition to type 83 (Linux Native) this may help some automated installation scripts find the right partition to use.
5. Install Linux on the partition.
6. Install LILO on the Linux partition NOT on the master boot record of the hard drive. This installs LILO as a second−stage boot loader on the Linux partition itself, to start up the kernel specified in the LILO configuration file. To do this, you should put

   ```
   boot = /dev/hda2
   ```

   (where `/dev/hda2` is the partition you want to boot from) in your `/etc/lilo/config` or `/etc/lilo.config` file.

7. Make sure that it is the Boot Manager partition that is marked active, so that you can use Boot Manager to choose what to boot.

There is a set of HOWTO's on the subject of multi−boot systems at the LDP Home Page, `http://www.linuxdoc.org/`.

---

4. File Systems, Disks, and Drives

4.1. How To Get Linux to Work with a Disk.

If your disk is an IDE or EIDE drive, you should read the file `/usr/src/linux/drivers/block/README.ide` (part of the Linux kernel source code). This `README` contains many helpful hints about IDE drives. Many modern IDE controllers do translation between `physical' cylinders/heads/sectors, and `logical' ones.

SCSI disks are accessed by linear block numbers. The BIOS invents some `logical' cylinder/head/sector fiction to support DOS.

Older IBM PC−compatible BIOS's will usually not be able to access partitions which extend beyond 1024 logical cylinders, and will make booting a Linux kernel from such partitions using LILO problematic at best.

You can still use such partitions for Linux or other operating systems that access the controller directly.

It's recommend that you create at least one Linux partition entirely under the 1024 logical cylinder limit, and boot from that. The other partitions will then be okay.

Also there seems to be a bit of trouble with the newer Ultra−DMA drives. I haven't gotten the straight scoop on them but they are becoming a very common problem at the SVLUG installfests. When you can get 8 to 12 Gig drives for $200 to $300 it's no wonder.

[Jim Dennis]

4.2. How To Undelete Files.

In general, this is very hard to do on unices because of their multitasking nature. Undelete functionality for the *ext2fs* file system is being worked on, but don't hold your breath.

There are a number of packages available which instead provide new commands for deleting and copying which move deleted files into a `wastebasket' directory. The files can be recovered until cleaned out automatically by background processing.

The Midnight Commander file manager provides an undelete facility that uses Ext2 file system library functions and an undelete directory for each file system. Commercial distribution packages of MC may or may not have this feature enabled, so be sure to look in the source code distribution for instructions on how to enable the undelete feature.

Alternatively, you can search the raw disk device which holds the file system in question. This is hard work, and you will need to be logged in as root to do this. But it can be done. Run **grep** on the raw device; e.g.:

```
grep −b 'bookmarks' /dev/hda
```

If the data has not been overwritten, you should be able to recover it with a text editor.

[Dave Cinege, Daniel Novotny]

4.3. How To Make Backups.

You can back up a directory hierarchy or complete file system to any media using GNU **tar** or **cpio**, the standard *nix tools for this purpose. **tar** seems to be the more commonly used program currently, and includes command line options to make compressed, incremental, and multi−volume backups. Complete information is contained in the documentation, which is in GNU Texinfo format.

The free program, Amanda, receives a lot of mentions on Usenet. Its home page is `http://www.amanda.org/`.

Several commercial backup utilities also exist. They are often included in commercial distributions.

4.4. How To Resize a Partition (Non−Destructively).

Use the **FIPS.EXE** program, included with most Linux distributions,under MS−DOS.

GNU **parted**, a partition editor, is stable enough for non−guru, mere−mortal use with relative confidence. Source code for the latest version is at: `ftp://ftp.gnu.org/pub/gnu/parted/`. There's also a boot disk image for resizing root partitions and for running **parted** on non−Linux machines. The disk image may be easier for beginners. Building from source could require some extra configuration.

Parted also has tutorial−style, plain−text documentation for Linux and FAT (MS−DOS) file systems.

Also, some commercial distributions come with their own partitioning software, like Partition Magic.

4.5. Is There a Defragmenter for Ext2fs?

Yes. There is defrag, a Linux file system defragmenter for *ext2*, *Minix*, and old−style *ext* file systems. It is available at `ftp://metalab.unc.edu/pub/Linux/system/filesystems/defrag-0.70.tar.gz`.

Users of the *ext2* file system can probably do without defrag, because *ext2* contains extra code to keep fragmentation reduced even in very full file systems.

4.6. How To Create a File System on a Floppy.

To format a 3.5−inch, high density floppy:

```
$ fdformat /dev/fd0H1440
$ mkfs -t ext2 -m 0 /dev/fd0H1440 1440
```

For a 5.25 inch floppy, use `fd0h1200` and 1200 as appropriate. For the `B:` drive use `fd1` instead of `fd0`.

The **−m 0** option tells mkfs.ext2 not to reserve any space on the disk for the superuser usually the last 10% is reserved for root.

The first command performs a low−level format. The second creates an empty file system. You can mount the floppy like a hard disk partition and simply **cp** and **mv** files, etc.

Device naming conventions generally are the same as for other unices. They can be found in Matt Welsh's *Installation and Getting Started* guide. Refer to (``*Where Is the Documentation?*'') A more detailed and technical description is *Linux Allocated Devices* by H. Peter Anvin, `hpa@zytor.com`, which is included in LaTeX and ASCII form in the kernel source distribution (probably in `/usr/src/kernel/Documentation/`), as `devices.tex` and `devices.txt`.

4.7. Does Linux Support Virtualized File Systems Like RAID?

The most recent Linux kernels support software RAID, and they will work with RAID disk controllers.

An automounter for NFS partitions is part of most Linux distributions.

In addition, several virtual file system projects exist. One of them, the *Linux Logical Volume Manager*, is located at `http://linux.msede.com/lvm/`.

4.8. Does Linux Support File System Encryption?

Yes. One file system, ppdd, is archived at `http://pweb.de.uu.net/flexsys.mtk/`.

4.9. Linux Prints Nasty Messages about Inodes, Blocks, and the Like.

You may have a corrupted file system, probably caused by not shutting Linux down properly before turning off the power or resetting. You need to use a recent shutdown program to do this for example, the one included in the util−linux package, available on sunsite and tsx−11.

If you're lucky, the program **fsck** (or **e2fsck** or **xfsck** as appropriate if you don't have the automatic **fsck** front−end) will be able to repair your file system. If you're unlucky, the file system is trashed, and you'll have to re−initialize it with mkfs (or **mke2fs**, **mkxfs**, etc.), and restore from a backup.

NB: don't try to check a file system that's mounted read/write this includes the root partition, if you don't see

```
VFS: mounted root ... read-only
```

at boot time.

4.10. The Swap Area Isn't Working.

When you boot (or enable swapping manually) you should see

```
Adding Swap: NNNNk swap-space
```

If you don't see any messages at all you are probably missing

```
swapon -av
```

(the command to enable swapping) in your /etc/rc.local or /etc/rc.d/* (the system startup scripts), or have forgotten to make the right entry in `/etc/fstab`:

```
/dev/hda2       none       swap       sw
```

for example.

If you see:

```
Unable to find swap-space signature
```

you have forgotten to run **mkswap**. See the manual page for details; it works much like **mkfs**.

Running, **free** in addition to showing free memory, should display:

```
             total       used       free
Swap:        10188       2960       7228
```

If typing ``cat /proc/swaps'' reveals only file or partition names, but no swap space information, then the swap file or partition needs re–initialization.

Use **fdisk** (as root) to determine which partition on a hard drive has been designated as the swap partition. The partition still needs to be initialized with **mkswap** before enabling it with **swapon**.

[Andy Jefferson, Steve Withers]

4.11. How To Add Temporary Swap Space.

In addition to a swap partition, Linux can also use a swap file. Some programs, like g++, can use huge amounts of virtual memory, requiring the temporary creation of extra space. To install an extra 64 MB of swap space, for example, use the following shell commands:

```
# dd if=/dev/zero of=/swap bs=1024 count=65535
# mkswap /swap
# swapon /swap
```

The **count=** argument to **dd** determines how big the swap file will be. In this example the swap file's name is `/swap`, but the name and location are, generally, arbitrary, depending only on the file system's available

space and your having write permissions in the directory.

When you don't need the swap space any more, remove it with the following statements:

```
# swapoff /swap
# rm /swap
```

Take a look also at the *Installation HOWTO* and *Installation & Getting Started* for detailed instructions.

If that still doesn't provide enough swap space, refer to (``*How To Have More Than 128Mb of Swap.*'')

4.12. How To Remove LILO So the System Boots DOS Again?

The **lilo** program (not the complete LILO package), uses the command line option **–u** to uninstall the LILO boot loader. You have to supply the device name of the device you installed LILO on, for example:

```
lilo -u /dev/hda
```

This rewrites the original, pre–LILO master boot record back to the first hard drive, from the boot record saved in `/boot/boot.0300`. If you installed LILO to a partition as a secondary boot loader, for example, `/dev/hda1`, **lilo** re–installs the original boot sector from the save file `/boot/boot.0301`. Refer to the **lilo** manual page for details. Thanks to Villy Kruse for reminding me to update this answer.

If you have an earlier version of LILO, you will have to use the DOS (MS–DOS 5.0 or later, or OS/2) `FDISK /MBR` (which is not documented). This will restore a standard MS–DOS Master Boot Record. If you have DR–DOS 6.0, go into `FDISK.EXE` in the normal way and then select the `Re-write Master Boot Record` option.

If you create a boot floppy during the Windows installation process, make sure that it contains the programs FDISK.EXE, FORMAT.COM, and SYS.COM, and use that to re–install MS–DOS on the hard disk.

If you don't have MS–DOS or DR–DOS, you need to have the boot sector that LILO saved when you first installed it. You did keep that file, didn't you? It's probably called `boot.0301` or some such. Type:

```
dd if=boot.0301 of=/dev/hda bs=445 count=1
```

(or `/dev/sda` if you're using a SCSI disk). This may also wipe out your partition table, so beware! If you're desperate, you could use

```
dd if=/dev/zero of=/dev/hda bs=512 count=1
```

This will erase your partition table and boot sector completely: you can then reformat the disk using your favorite software. But this will render the contents of your disk inaccessible you'll lose it all unless you're an expert.

Note that the DOS MBR boots whichever (single!) partition is flagged as ``active.'' You may need to use **fdisk** to set and clear the active flags on partitions appropriately.

4.13. Why Does fdformat Require Superuser Privileges?

The system call to format a floppy can only be done as root, regardless of the permissions of `/dev/fd0*`. If you want any user to be able to format a floppy, try getting the **fdformat2** program. This works around the problems by being setuid to root.

4.14. The System Checks the Ext2fs Partitions Each Reboot.

Refer to (``*EXT2−fs: warning: mounting unchecked file system.*)

4.15. Root File System Is Read−Only.

Remount it. If `/etc/fstab` is correct, you can simply type:

```
mount −n −o remount /
```

If `/etc/fstab` is wrong, you must give the device name and possibly the type, too: e.g.

```
mount −n −o remount −t ext2 /dev/hda2 /
```

To understand how you got into this state, see, (``*EXT2−fs: warning: mounting unchecked file system.*'')

4.16. What Is `/proc/kcore`?

None of the files in `/proc` are really there they're all, ``pretend,'' files made up by the kernel, to give you information about the system and don't take up any hard disk space.

`/proc/kcore` is like an ``alias'' for the memory in your computer. Its size is the same as the amount of RAM you have, and if you read it as a file, the kernel does memory reads.

4.17. The AHA1542C Doesn't Work with Linux.

The option to allow disks with more than 1024 cylinders, which the AHA1542C card can recognize, is only required as a workaround for a PC−compatible BIOS misfeature and should be turned `off' under Linux. For older Linux kernels you need to turn off most of the `advanced BIOS' options all but the one about scanning the bus for bootable devices.

4.18. Where Is the Journalling File System on the Net?

The journalling file system, named Reiserfs has just been released from testing. It is said to make Linux even faster than Linux with the Ext2 file system installed. Complete information is available at `http://devlinux.org/namesys/`.

---

5. Porting, Compiling and Obtaining Programs

5.1. How To Compile Programs.

Most Linux software is written in C and compiled with the GNU C compiler. GCC is a part of every Linux distribution. The latest compiler version, documentation, and patches are on `ftp://ftp.gnu.org/pub/gnu/`.

Programs that are written in C++ must be compiled with the GNU G++ compiler, which is also included in Linux distributions and available from the same place as GCC.

To build version 2.0.x kernels, you will need GCC version 2.7.2.x, approximately. Trying to build an early Linux kernel with a different compiler, like GCC 2.8.x, EGCS, or PGCC, may cause problems because of GCC related code dependencies. Kernel versions 2.3 and 2.4 should compile correctly with more recent compilers.

Information on the EGCS compiler is at `http://www.gnu.org/software/gcc/gcc.html`.

Note that at this time, the kernel developers are not answering bug requests for earlier kernels, but instead are concentrating on developing 2.4.x version kernels and maintaining 2.2.x version kernels.

[J.H.M. Dassen, Axel Boldt]

5.2. How To Install GNU Software.

On a correctly configured system, installing a GNU software package requires four steps:

- With the `source.tar.gz` archive in the `/usr/src/` directory, or wherever you maintain your source files, untar and decompress the package with the command:

```
tar zxvf package-name.tar.gz
```

- Run the **./configure** script in the untarred source archive's top–level directory with whatever command line arguments you need. The options that configure recognizes are usually contained in a file called `INSTALL` or `README`.
- Run **make**. This will build the source code into an executable program (or programs) and may take a few minutes or a few hours, depending on the speed of the computer and the size of the package.
- Run **make install**. This will install the compiled binaries, configuration files, and any libraries in the appropriate directories.

5.3. Where To Get Java.

The Sun Microsystems Java runtime environments and developer's kits are at `http://java.sun.com/java/`.

You can also get the source code, which is licensed by Sun Microsystems.

The home page of IBM's Jikes Java compiler is
`http://www10.software.ibm.com/developerworks/opensource/jikes`.

There is a version of Sun's HotJava browser for Linux at:
`http://www.java.sun.com/products/hotjava/`.

Kaffee, a free Java clone, is available from: `http://www.kaffe.org/`.

There is a resource page of free and commercial Java packages at:
`http://www.blackdown.org/java-linux.html`.

Netscape Communicator comes with its own version of the Java Runtime Environment, so if you need Java only to view Web graphics, you may already have the runtime version of the Java Virtual Machine and libraries that you need installed on your system. You can download Communicator from
`http://www.netscape.com/`.

5.4. How To Port *XXX* to Linux.

In general, *nix programs need very little porting. Simply follow the installation instructions. If you don't know and don't know how to find out the answers to some of the questions asked during the installation procedure, you can guess, but this tends to produce buggy programs. In this case, you're probably better off asking someone else to do the port. If you have a BSD–ish program, you should try using **–I/usr/include/bsd** and **–lbsd** on the appropriate parts of the compilation lines.

5.5. What Is ld.so and How To Get It?

ld.so is the dynamic library loader. Each binary using shared libraries used to have about 3K of start–up code to find and load the shared libraries. Now that code has been put in a special shared library, `/lib/ld.so`, where all binaries can look for it, so that it wastes less disk space, and can be upgraded more easily. `ld.so` can be obtained from `http://tsx-11.mit.edu/pub/linux/packages/GCC/` and mirror sites. The latest version at the time of writing is `ld.so.1.9.5.tar.gz`. `/lib/ld-linux.so.1` is the same thing for ELF ("What's all this about ELF? ") and comes in the same package as the `a.out` loader.

5.6. How To Upgrade the Libraries without Trashing the System.

Note: You should always have a rescue disk set ready when you perform this procedure, in the likely event that something goes wrong!

This procedure is especially difficult if you're upgrading very old libraries like libc4. But you should be able to keep libc4 on the same system with libc5 libraries for the programs that still need them. The same holds true for upgrading from libc5 to the newer–yet glibc2 libraries.

The problem with upgrading dynamic libraries is that, the moment you remove the old libraries, the utilities that you need to upgrade to the new version of the libraries don't work. There are ways around around this. One is to temporarily place a spare copy of the run time libraries, which are in `/lib/`, in `/usr/lib/`, or `/usr/local/lib/`, or another directory that is listed in the `/etc/ld.so.conf` file.

For example, when upgrading libc5 libraries, the files in `/lib/` might look something like:

```
libc.so.5
libc.so.5.4.33
libm.so.5
libm.so.5.0.9
```

These are the C libraries and the math libraries. Copy them to another directory that is listed in `/etc/ld.so.conf`, like `/usr/lib/`:

```
$ cp -df /lib/libc.so.5* /usr/lib/
$ cp -df /lib/libm.so.5* /usr/lib/
$ ldconfig
```

Be sure to run **ldconfig** to upgrade the library configuration.

The files `libc.so.5` and `libm.so.5` are symbolic links to the actual library files. When you upgrade, the new links will not be created if the old links are still there, unless you use the **–f** flag with **cp**. The **–d** flag to **cp** will copy the symbolic link itself, and not the file it points to.

If you need to overwrite the link to the library directly, use the **–f** flag with **ln**.

For example, to copy new libraries over the old ones, try this. Make a symbolic link to the new libraries first, then copy both the libraries and the links to `/lib/`, with the following commands.

```
$ ln -sf ./libm.so.5.0.48 libm.so.5
$ ln -sf ./libc.so.5.0.48 libc.so.5
$ cp -df libm.so.5* /lib
$ cp -df libc.so.5* /lib
```

Again, remember to run **ldconfig** after you copy the libraries.

If you are satisfied that everything is working correctly, you can remove the temporary copies of the old libraries from `/usr/lib/` or wherever you copied them.

5.7. How To Use Code or a Compiler Compiled for a 486 on a 386.

Yes, unless it's the kernel.

The **–m486** option to GCC, which is used to compile binaries for x486 machines, merely changes certain optimizations. This makes for slightly larger binaries that run somewhat faster on a 486. They still work fine on a 386, though, with a small performance hit.

However, from version 1.3.35 the kernel uses 486 or Pentium−specific instructions if configured for a 486 or Pentium, thus making it unusable on a 386.

GCC can be configured for a 386 or 486; the only difference is that configuring it for a 386 makes **−m386** the default and configuring for a 486 makes **−m486** the default. In either case, these can be overridden on a per−compilation basis or by editing `/usr/lib/gcc−lib/i*−linux/` *n.n.n*`/specs`.

There is an alpha version of GCC that knows how to do optimization well for the 586, but it is quite unreliable, especially at high optimization settings. The Pentium GCC can be found on `ftp://tsx−11.mit.edu/pub/linux/ALPHA/pentium−gcc/`. The ordinary 486 GCC supposedly produces better code for the Pentium using the **−m386**, or at least slightly smaller.

5.8. What Does ``gcc −O6'' Do?

Currently, the same as **−O2** (GCC 2.5) or **−O3** (GCC 2.6, 2.7). Any number greater than that does the same thing. The `Makefiles` of newer kernels use **−O2**, and you should probably do the same.

5.9. Where Are `linux/*.h` and `asm/*.h`?

The files `/usr/include/linux/` and `/usr/include/asm/` are often soft links to the directories where the kernel headers are. They are usually under `/usr/src/kernel*/`.

If you don't have the kernel sources, download them. Refer to the answer: (``*How To Upgrade/Recompile a Kernel.*'')

Then, use **rm** to remove any garbage, and **ln** to create the links:

```
$ rm −rf /usr/include/linux /usr/include/asm
$ ln −sf /usr/src/linux/include/linux /usr/include/linux
$ ln −sf /usr/src/linux/include/asm−<architecture> /usr/include/asm
```

The assembly language files reside in architecture−specific directories, so you need to link `/usr/src/include/asm` to `/usr/src/linux/include/asm−i386` on PC compatible systems, to `/usr/src/linux/include/asm−sparc` on Sun Sparc systems, to `/usr/src/linux/include/asm−ppc` on PPC systems, and so on.

You'll also find that you may need to do `make config' as in a newly−unpacked kernel source tree, to create `linux/autoconf.h`.

5.10. What To Do about Errors Trying to Compile the Kernel.

See the previous question regarding the header files.

Remember that when you apply a patch to the kernel, you must use the ``−p0'' or ``−p1'' option: otherwise, the patch may be misapplied. See the patch manual page for details.

``ld: unrecognized option `-qmagic'" means that you should get a newer linker, from ftp://tsx-11.mit.edu/pub/linux/packages/GCC/, in the file binutils-2.8.1.0.1.bin.tar.gz.

5.11. How To Make a Shared Library.

For ELF,

```
$ gcc -fPIC -c *.c
$ gcc -shared -Wl,-soname,libfoo.so.1 -o libfoo.so.1.0 *.o
```

For a.out, get tools-*n.nn*.tar.gz from tsx-11.mit.edu/pub/linux/packages/GCC/src/. It comes with documentation that will tell you what to do. Note that a.out shared libraries are a very tricky business. Consider upgrading your libraries to ELF shared libraries. See the *ELF HOWTO*, at ftp://metalab.unc.edu/pub/Linux/docs/HOWTO/

5.12. Programs Are Very Large.

With an ELF compiler (``*What's All This about ELF? glibc?*"), the most common cause of large executables is the lack of an appropriate .so library link for one of the libraries you're using. There should be a link like libc.so for every library like libc.so.5.2.18.

With an a.out compiler the most common cause of large executables is the **–g** linker (compiler) flag. This produces (as well as debugging information in the output file) a program which is statically linked one which includes a copy of the C library instead of a dynamically linked copy.

Other things worth investigating are **–O** and **–O2**, which enable optimization (check the GCC documentation), and –s (or the strip command) which strip the symbol information from the resulting binary (making debugging totally impossible).

You may wish to use –N on very small executables (less than 8K with the –N), but you shouldn't do this unless you understand its performance implications, and definitely never with daemons.

5.13. Does Linux Support Threads or Lightweight Processes?

As well as the Unix multiprocessing model involving heavyweight processes, which is of course part of the standard Linux kernel, there are several implementations of lightweight processes or threads. Recent kernels implement a thread model, kthreads. In addition, there are the following packages available for Linux.

- GNU glibc2 for Linux has optional support for threads. The archive is available from the same place as glibc2, ftp://ftp.gnu.org/pub/gnu/
- In sipb.mit.edu:/pub/pthread/ or ftp.ibp.fr:/pub/unix/threads/pthreads. Documentation isn't in the package, but is available on the World Wide Web at

`http://www.mit.edu:8001/people/proven/home_page.html`. Newer Linux libc's contain the pthreads source. The GNU Ada compiler on `ftp://metalab.unc.edu/pub/Linux/devel/lang/ada/gnat-3.01-linux+elf.tar.gz` cont binaries made from that source code.

- In `ftp://ftp.cs.washington.edu:/pub/qt-001.tar.Z` is QuickThreads. More information can be found in the technical report, available on the same site is `/tr/1993/05/UW-CSE-93-05-06.PS.Z`.
- In gummo.doc.ic.ac.uk/rex/ is lwp, a very minimal implementation.
- In `ftp://ftp.cs.fsu.edu:/pub/PART/`, an Ada implementation. This is useful mainly because it has a lot of Postscript papers that you'll find useful in learning more about threads. This is not directly usable under Linux.

Please contact the authors of the packages in question for details.

5.14. Where To Find `lint` for Linux.

Roughly equivalent functionality is built into GCC. Use the `-Wall` option to turn on most of the useful extra warnings. See the GCC manual for more details (type **F1–i** in Emacs and select the entry for GCC).

There is a freely available program called `lclint` that does much the same thing as traditional lint. The announcement and source code are available at on `ftp://larch.lcs.mit.edu/pub/Larch/lclint/`; on the World Wide Web, look at `http://lclint.cs.virginia.edu/`.

5.15. Where To Find Kermit for Linux.

Kermit is distributed under a non–GPL copyright that makes its terms of distribution somewhat different. The sources and some binaries are available on `ftp://kermit.columbia.edu`.

The WWW Home Page of the Columbia University Kermit project is `http://www.columbia.edu/kermit/`.

5.16. How To Use Linux with a Cable Modem.

The www.CablemodemInfo.com and xDSL Web page at `http://www.cablemodeminfo.com/` has a section devoted to Linux.

5.17. Is There an ICQ Program That Runs under Linux?

Several ICQ clients are available on metalab.unc.edu. (Refer to: ``*Where Are the Linux FTP Archives?*'') ICQ itself does not have a Linux client, but there is a Java client at `http://www.mirabilis.com/download/`.

---

6. Solutions to Common Miscellaneous Problems

6.1. FTP Transfers Seem to Hang.

FTP transfers that die suddenly are due, apparently, to some form of overrunning buffer. It occurs both with Linux and Microsoft servers. On Linux systems, The problem seems to occur most commonly with the distribution's server software.

If you receive `ftp: connection refused` errors, then the problem is likely due to a lack of authentication. Refer to ``*FTP or Telnet Server Won't Allow Logins.*.''

One remedy is to be replacing the distribution FTP server with the Linux port of the OpenBSD FTP server. The home page is: `http://www.eleves.ens.fr:8080/home/madore/programs/`

To install the BSD server, follow the installation instructions, and refer to the manual pages for `inetd` and `inetd.conf`. (If you have the newer `xinetd`, see below.) Be sure to tell `inetd` to run the BSD daemon alone, not as a subprocess of, for example, `tcpd`. Comment out the line that begins ``ftp'' in the `/etc/inetd.conf` file and replace it with a line similar to (if you install the new `ftpd` in `/usr/local/sbin/`):

```
# Original entry, commented out.
#ftp    stream  tcp     nowait  root    /usr/sbin/tcpd /usr/sbin/in.ftpd

# Replacement entry:
ftp     stream  tcp     nowait  root    /usr/local/sbin/ftpd -l
```

The replacement daemon will become effective after rebooting or sending (as root) a `SIGHUP` to `inetd`, e.g.:

```
# kill -HUP inetd
```

To configure `xinetd`, create an entry in `/etc/xinetd.d` per the instructions in the `xinetd.conf` manual page. Make sure, again, that the command–line arguments for `ftpd` are correct, and that you have installed the `/etc/ftpusers` and `/etc/pam.d/ftp` files. Then restart `xinetd` with the command: `/etc/rc.d/init.d/xinetd restart`. The command should report ``OK,'' and the restart will be noted in the system message log.

6.2. Free Dumps Core.

In Linux 1.3.57 and later, the format of `/proc/meminfo` was changed in a way that the implementation of free doesn't understand. Get the latest version, from metalab.unc.edu, in `/pub/Linux/system/Status/ps/procps-0.99.tgz`.

6.3. Netscape Crashes Frequently.

Netscape shouldn't crash, if it and the network are properly configured. Some things to check:

- Make sure that the MOZILLA_HOME environment variable is correctly set. If you installed Netscape under /usr/local/netscape/, for example, that should be the value of MOZILLA_HOME. Set it from the command line (e.g, ``export MOZILLA_HOME="/usr/local/netscape"'' under **bash** or add it to one your personal or system initialization files. Refer to the manual page for your shell for details.
- If you have a brand–new version of Netscape, try a previous version, in case the run–time libraries are slightly incompatible. For example, if Netscape version 4.75 is installed (type ``netscape --version'' at the shell prompt), try installing version 4.7. All versions are archived at ftp://ftp.netscape.com/.
- Netscape uses its own Motif and Java Runtime Environment libraries. If a separate version of either is installed on your system, ensure that they aren't interfering with Netscape's libraries; e.g., by un–installing them.
- Make sure that Netscape can connect to its default name servers. The program will appear to freeze and time out after several minutes if it can't. This indicates a problem with the system's Internet connection; likely, the system can't connect to other sites, either.

6.4. FTP or Telnet Server Won't Allow Logins.

This applies to server daemons that respond to clients, but don't allow logins. On new systems that have Pluggable Authentication Modules installed, look for a file named, ``ftp,'' or ``telnet,'' in the directory /etc/pam/ or /etc/pam.d/. If the corresponding authentication file doesn't exist, the instructions for configuring FTP and Telnet authentication and other PAM configuration, should be in /usr/doc/pam-<version>. Refer also to the answer for ``*FTP server says: ``421 service not available, remote server has closed connection.''*.''

If it's an FTP server on an older system, make sure that the account exists in /etc/passwd, especially ``anonymous.''

This type of problem may also be caused a failure to resolve the host addresses properly, especially if using Reverse Address Resolution Protocol (RARP). The simple answer to this is to list all relevant host names and IP addresses in the /etc/hosts files on each machine. (Refer to the example /etc/hosts and /etc/resolv.conf files in: ``*Sendmail Pauses for Up to a Minute at Each Command.*.'') If the network has an internal DNS, make sure that each host can resolve network addresses using it.

If the host machine doesn't respond to FTP or Telnet clients at all, then the server daemon is not installed correctly, or at all. Refer to the manual pages: inetd and inetd.conf on older systems, or xinetd and xinetd.conf, as well as ftpd, and telnetd.

6.5. How To Keep Track of Bookmarks in Netscape?

This probably applies to most other browsers, too. In the Preferences/Navigator menu, set your home page to Netscape's bookmarks.html file, which is located in the .netscape (with a leading period) subdirectory. For example, if your login name is ``smith,'' set the home page to:

```
file://home/smith/.netscape/bookmarks.html
```

Setting up your personal home page like this will present you with a nicely formatted (albeit possibly long)

page of bookmarks when Netscape starts. And the file is automatically updated whenever you add, delete, or visit a bookmarked site.

6.6. The Computer Has the Wrong Time.

There are two clocks in your computer. The hardware (CMOS) clock runs even when the computer is turned off, and is used when the system starts up and by DOS (if you use DOS). The ordinary system time, shown and set by **date**, is maintained by the kernel while Linux is running.

You can display the CMOS clock time, or set either clock from the other, with `/sbin/clock` (now called **hwclock** in many distributions). Refer to: `man 8 clock` or `man 8 hwclock`.

There are various other programs that can correct either or both clocks for system drift or transfer time across the network. Some of them may already be installed on your system. Try looking for **adjtimex** (corrects for drift), Network Time Protocol clients like **netdate**, **getdate**, and **xntp**, or NTP client−server suite like **chrony**. Refer to: ``*How to Find a Particular Application.*.''

6.7. Setuid Scripts Don't Seem to Work.

That's right. This feature has been disabled in the Linux kernel on purpose, because setuid scripts are almost always a security hole. Sudo and SuidPerl can provide more security than setuid scripts or binaries, especially if execute permissions are limited to a certain user ID or group ID.

If you want to know why setuid scripts are a security hole, read the FAQ for *comp.unix.questions*.

6.8. Free Memory as Reported by free Keeps Shrinking.

The ``free'' figure printed by **free** doesn't include memory used as a disk buffer cache shown in the ``buffers'' column. If you want to know how much memory is really free add the ``buffers'' amount to ``free.'' Newer versions of **free** print an extra line with this info.

The disk buffer cache tends to grow soon after starting Linux up. As you load more programs and use more files, the contents get cached. It will stabilize after a while.

6.9. When Adding More Memory, the System Slows to a Crawl.

This is a common symptom of a failure to cache the additional memory. The exact problem depends on your motherboard.

Sometimes you have to enable caching of certain regions in your BIOS setup. Look in the CMOS setup and see if there is an option to cache the new memory area which is currently switched off. This is apparently most common on a '486.

Sometimes the RAM has to be in certain sockets to be cached.

Sometimes you have to set jumpers to enable caching.

Some motherboards don't cache all of the RAM if you have more RAM per amount of cache than the hardware expects. Usually a full 256K cache will solve this problem.

If in doubt, check the manual. If you still can't fix it because the documentation is inadequate, you might like to post a message to *comp.os.linux.hardware* giving all of the details make, model number, date code, etc., so other Linux users can avoid it.

6.10. Some Programs (E.g. xdm) Won't Allow Logins.

You are probably using non−shadow password programs and are using shadow passwords.

If so, you have to get or compile a shadow password version of the programs in question. The shadow password suite can be found at `ftp://tsx-11.mit.edu:/pub/linux/sources/usr.bin/shadow/`. This is the source code. The binaries are probably in `linux/binaries/usr.bin/`.

6.11. Some Programs Allow Logins with No Password.

You probably have the same problem as in (``*Some Programs (E.g. xdm) Won't Allow Logins.*''), with an added wrinkle.

If you are using shadow passwords, you should put a letter `x' or an asterisk in the password field of `/etc/passwd` for each account, so that if a program doesn't know about the shadow passwords it won't think it's a passwordless account and let anyone in.

6.12. The Machine Runs Very Slowly with GCC / X / ...

You may have too little real memory. If you have less RAM than all the programs you're running at once, Linux will swap to your hard disk instead and thrash horribly. The solution in this case is to not run so many things at once or buy more memory. You can also reclaim some memory by compiling and using a kernel with less options configured. See (``*How To Upgrade/Recompile a Kernel.*'')

You can tell how much memory and swap you're using with the free command, or by typing:

```
$ cat /proc/meminfo
```

If your kernel is configured with a RAM disk, this is probably wasted space and will cause things to go slowly. Use LILO or rdev to tell the kernel not to allocate a RAM disk (see the LILO documentation or type ``man rdev'').

6.13. System Only Allows Root Logins.

You probably have some permission problems, or you have a file `/etc/nologin`.

In the latter case, put "`rm -f /etc/nologin`" in your `/etc/rc.local` or `/etc/rc.d/*` scripts.

Otherwise, check the permissions on your shell, and any file names that appear in error messages, and also the directories that contain these files, up to and including the root directory.

6.14. The Screen Is All Full of Weird Characters Instead of Letters.

You probably sent some binary data to your screen by mistake. Type `echo '\033c'` to fix it. Many Linux distributions have a command, **reset**, that does this.

If that doesn't help, try a direct screen escape command.

```
$ echo 'Ctrl-V Ctrl-O'
```

This resets the default font of a Linux console. Remember to hold down the Control key and type the letter, instead of, for example, **Ctrl**, then **V**. The sequence

```
$ echo 'Ctrl-V Esc C'
```

causes a full screen reset. If there's data left on the shell command line after typing a binary file, press **Ctrl-C** a few times to restore the shell command line.

Another possible command is an alias, "`sane`," that can work with generic terminals:

```
$ alias sane='echo -e "\\033c";tput is2; \
> stty sane line 1 rows $LINES columns $COLUMNS'
```

The alias is enclosed with open quotes (backticks), not single quotes. The line break is included here for clarity, and is not required.

Make sure that `$LINES` and `$COLUMNS` are defined in the environment with a command similar to this in `~/.cshrc` or `~/.bashrc`,

```
$ LINES=25; export $LINES; $COLUMNS=80; export $COLUMNS
```

using the correct numbers of `$LINES` and `$COLUMNS` for the terminal.

Finally, the output of "**stty -g**" can be used to create a shell script that will reset the terminal:

1. Save the output of "**stty -g**" to a file. In this example, the file is named "termset.":

```
$ stty -g >termset
```

The output of ``**stty –g**" (the contents of ``termset") will look something like:

```
500:5:bd:8a3b:3:1c:7f:15:4:0:1:0:11:13:1a:0:12:f:17:16:0:0:73
```

2. Edit ``termset" to become a shell script; adding an interpreter and ``**stty**" command:

```
#!/bin/bash
stty 500:5:bd:8a3b:3:1c:7f:15:4:0:1:0:11:13:1a:0:12:f:17:16:0:0:73
```

3. Add executable permissions to ``**termset**" and use as a shell script:

```
$ chmod +x termset
$ ./termset
```

[Floyd L. Davidson, Bernhard Gabler]

6.15. I Screwed Up the System and Can't Log In to Fix It.

Reboot from an emergency floppy or floppy pair. For example, the Slackware boot and root disk pair in the install subdirectory of the Slackware distribution.

There are also two, do–it–yourself rescue disk creation packages in `ftp://metalab.unc.edu/pub/Linux/system/recovery/`. These are better because they have your own kernel on them, so you don't run the risk of missing devices and file systems.

Get to a shell prompt and mount your hard disk with something like

```
$ mount -t ext2 /dev/hda1 /mnt
```

Then your file system is available under the directory `/mnt` and you can fix the problem. Remember to unmount your hard disk before rebooting (**cd** somewhere else first, or it will say it's busy).

6.16. I Forgot the `root` Password.

Note: Incorrectly editing any of the files in the `/etc/` directory can severely screw up a system. Please keep a spare copy of any files in case you make a mistake.

If your Linux distribution permits, try booting into single–user mode by typing ``single" at the `BOOT lilo:` prompt. With more recent distributions, you can boot into single–user mode when prompted by typing ``linux 1,"``linux single," or ``init=/bin/bash."

If the above doesn't work for you, boot from the installation or rescue floppy, and switch to another virtual console with **Alt–F1 –– Alt–F8**, and then mount the root file system on `/mnt`. Then proceed with the steps below to determine if your system has standard or shadow passwords, and how to remove the password.

Using your favorite text editor, edit the root entry of the `/etc/passwd` file to remove the password, which is located between the first and second colons. *Do this only if the password field does not contain an ``x," in which case see below.*

```
root:Yhgew13xs:0:0: ...
```

Change that to:

```
root::0:0: ...
```

If the password field contains an ``x,'' then you must remove the password from the `/etc/shadow` file, which is in a similar format. Refer to the manual pages: ``man passwd,'' and ``man 5 shadow.''

[Paul Colquhuon, Robert Kiesling, Tom Plunket]

6.17. There's a Huge Security Hole in rm!

No there isn't. You are obviously new to unices and need to read a good book to find out how things work. Clue: the ability to delete files depends on permission to write in that directory.

6.18. lpr and/or lpd Don't Work.

First make sure that your `/dev/lp*` port is correctly configured. Its IRQ (if any) and port address need to match the settings on the printer card. You should be able to dump a file directly to the printer:

```
$ cat the_file >/dev/lp1
```

If **lpr** gives you a message like `myname@host: host not found`" it may mean that the TCP/IP loopback interface, `lo`, isn't working properly. Loopback support is compiled into most distribution kernels. Check that the interface is configured with the ifconfig command. By Internet convention, the network number is 127.0.0.0, and the local host address is 127.0.0.1. If everything is configured correctly, you should be able to telnet to your own machine and get a login prompt.

Make sure that `/etc/hosts.lpd` contains the machine's host name.

If your machine has a network–aware **lpd**, like the one that comes with LPRng, make sure that `/etc/lpd.perms` is configured correctly. Also look at the *Printing HOWTO*. "Where can I get the HOWTO's and other documentation? ".

6.19. Timestamps on Files on MS–DOS Partitions Are Set Incorrectly

There is a bug in the program **clock** (often found in `/sbin`). It miscounts a time zone offset, confusing seconds with minutes or something like that. Get a recent version.

6.20. How To Get LILO to Boot the Kernel Image.

From kernel versions 1.1.80 on, the compressed kernel image, which is what LILO needs to find, is in `arch/i386/boot/zImage`, or `arch/i386/boot/bzImage` when it is built, and is normally stored in the `/boot/` directory. The `/etc/lilo.conf` file should refer to the `vmlinuz` symbolic link, not the actual kernel image.

This was changed to make it easier to build kernel versions for several different processors from one source tree.

6.21. How To Make Sure the System Boots after Re–Installing the Operating System.

This should work whether you're re–installing Linux or some other, commercial, operating system:

- Insert a blank, formatted floppy in drive A:
- Save a copy of the boot hard drive's Master Boot Record to the floppy, by executing the command:

```
#dd if=/dev/hda of=/dev/fd0 count=1
```

  `dd` is a standard program on Linux systems. A MS–Windows compatible version is available from `ftp://ftp.gnu.org/`, as well as many MS software archives.

- Test that the floppy boots the system by rebooting with the floppy in the A: drive.
- Then you should be able to install the other operating system (on a different hard drive and/or partition, if you don't want to uninstall Linux).
- After installation, boot Linux again from the floppy, and re–install the MBR with the command: `/sbin/lilo`.

[Jacques Guy]

6.22. The PCMCIA Card Doesn't Work after Upgrading the Kernel.

The PCMCIA Card Services modules, which are located in `/lib/modules/`*version*`/pcmcia`, where *version* is the version number of the kernel, use configuration information that is specific to that kernel image only. The PCMCIA modules on your system will not work with a different kernel image. You need to upgrade the PCMCIA card modules when you upgrade the kernel.

When upgrading from older kernels, make sure that you have the most recent version of the run–time libraries, the modutils package, and so on. Refer to the file `Documentation/Changes` in the kernel source tree for details.

Important: If you use the PCMCIA Card Services, do not enable the `Network device support/Pocket and portable adapters` option of the kernel configuration menu, as this conflicts with the modules in Card Services.

Knowing the PCMCIA module dependencies of the old kernel is useful. You need to keep track of them. For example, if your PCMCIA card depends on the serial port character device being installed as a module for the old kernel, then you need to ensure that the serial module is available for the new kernel and PCMCIA modules as well.

The procedure described here is somewhat kludgey, but it is much easier than re−calculating module dependencies from scratch, and making sure the upgrade modules get loaded so that both the non−PCMCIA and PCMCIA are happy. Recent kernel releases contain a myriad of module options, too many to keep track of easily. These steps use the existing module dependencies as much as possible, instead of requiring you to calculate new ones.

However, this procedure does not take into account instances where module dependencies are incompatible from one kernel version to another. In these cases, you'll need to load the modules yourself with insmod, or adjust the module dependencies in the `/etc/conf.modules` file. The `Documentation/modules.txt` file in the kernel source tree contains a good description of how to use the kernel loadable modules and the module utilities like **insmod**, **modprobe**, and **depmod**. `Modules.txt` also contains a recommended procedure for determining which features to include in a resident kernel, and which to build as modules.

Essentially, you need to follow these steps when you install a new kernel.

- Before building the new kernel, make a record with the **lsmod** command of the module dependencies that your system currently uses. For example, part of the **lsmod** output might look like this:

```
Module          Pages    Used by
memory_cs           2           0
ds                  2    [memory_cs]     3
i82365              4           2
pcmcia_core         8    [memory_cs ds i82365]   3
sg                  1           0
bsd_comp            1           0
ppp                 5    [bsd_comp]      0
slhc                2    [ppp]   0
serial              8           0
psaux               1           0
lp                  2           0
```

  This tells you for example that the `memory_cs` module needs the ds and `pcmcia_core` modules loaded first. What it doesn't say is that, in order to avoid recalculating the module dependencies, you may also need to have the `serial`, `lp`, `psaux`, and other standard modules available to prevent errors when installing the pcmcia routines at boot time with **insmod**. A glance at the `/etc/modules` file will tell you what modules the system currently loads, and in what order. Save a copy of this file for future reference, until you have successfully installed the new kernel's modules. Also save the **lsmod** output to a file, for example, with the command: **lsmod >lsmod.old−kernel.output**.

- Build the new kernel, and install the boot image, either `zImage` or `bzImage`, to a floppy diskette. To do this, change to the `arch/i386/boot` directory (substitute the correct architecture directory if you don't have an Intel machine), and, with a floppy in the diskette drive, execute the command:

```
$ dd if=bzImage of=/dev/fd0 bs=512
```

if you built the kernel with the **make bzImage** command, and if your floppy drive is `/dev/fd0`. This results in a bootable kernel image being written to the floppy, and allows you to try out the new kernel without replacing the existing one that LILO boots on the hard drive.

- Boot the new kernel from the floppy to make sure that it works.
- With the system running the new kernel, compile and install a current version of the PCMCIA Card Services package, available from metalab.unc.edu as well as other Linux archives. Before installing the Card Services utilities, change the names of `/sbin/cardmgr` and `/sbin/cardctl` to `/sbin/cardmgr.old` and `/sbin/cardctl.old`. The old versions of these utilities are not compatible with the replacement utilities that Card Services installs. In case something goes awry with the installation, the old utilities won't be overwritten, and you can revert to the older versions if necessary. When configuring Card Services with the ``make config'' command, make sure that the build scripts know where to locate the kernel configuration, either by using information from the running kernel, or telling the build process where the source tree of the new kernel is. The ``make config'' step should complete without errors. Installing the modules from the Card Services package places them in the directory `/lib/modules/`*version*`/pcmcia`, where *version* is the version number of the new kernel.
- Reboot the system, and note which, if any, of the PCMCIA devices work. Also make sure that the non−PCMCIA hardware devices are working. It's likely that some or all of them won't work. Use **lsmod** to determine which modules the kernel loaded at boot time, and compare it with the module listing that the old kernel loaded, which you saved from the first step of the procedure. (If you didn't save a listing of the **lsmod** output, go back and reboot the old kernel, and make the listing now.)
- When all modules are properly loaded, you can replace the old kernel image on the hard drive. This will most likely be the file pointed to by the `/vmlinuz` symlink. Remember to update the boot sector by running the **lilo** command after installing the new kernel image on the hard drive.

Also look at the questions, How do I upgrade/recompile my kernel? and Modprobe can't locate module, "XXX," and similar messages.

6.23. How To Remove (or Change) the Colors in the ls Display.

The shell command, ``unalias ls,'' should completely unset the configuration that some distributions provide as standard. To change the colors, refer to the ls man page (``man ls'').

6.24. Why Won't a Program Work in the Current Directory?

Because the current directory (i.e., ``.'') is not in the search path, for security reasons, as well as to insure that the correct program versions are used. If an intruder is able to write a file to a world−writable directory, like `/tmp`, presumably he or she would be able to execute it if the directory were in the search path. The solution to this is to include the directory in the command; e.g., ``./myprog,'' instead of ``myprog.'' Or add the current directory to your `PATH` environment variable; e.g., ``export PATH=".:"$PATH'' using bash, although this is discouraged for the reasons mentioned above.

---

7. How To Do This or Find Out That...

7.1. How To Find Out If a Notebook Runs Linux.

There's no fixed answer to this question, because notebook hardware is constantly updated, and getting the X display, sound, PCMCIA, modem, and so forth, working, can take a good deal of effort.

Most notebooks currently on the market, for example, use ``Winmodems,'' which often do not work with Linux because of their proprietary hardware interfaces. Even notebooks which are certified as ``Linux compatible,'' may not be completely compatible.

Information about installing Winmodems in general is contained in the *Winmodems−and−Linux HOWTO*. (Refer to ``*Where Is the Documentation?*'')

You can find the most current information, or ask other users about their notebook experiences, on the linux−laptop mailing list, which is hosted by the `vger.redhat.com` server. (Refer to ``*What Mailing Lists Are There?*'')

A mailing list for Linux on IBM Thinkpads has its home page at `http://www.topica.com/lists/linux-thinkpad/`.

Another Thinkpad mailing list is hosted by `http://www.bm-soft.com/`. Send email with the word ``help'' in the body of the message to `majordomo@www.bm-soft.com`.

There is a Web page about Linux on IBM Thinkpads at `http://peipa.essex.ac.uk/tp-linux/`.

The Linux Laptop home page is at `http://www.cs.utexas.edu/users/kharker/linux-laptop/`.

For information about interfacing peripherals like Zip and CD−ROM drives through parallel ports, refer to the Linux Parallel Port Home Page, at `http://www.torque.net/linux-pp.html`.

If you need the latest version of the PCMCIA Card Services package, it is (or was) located at `ftp://cb-iris.stanford.edu/pub/pcmcia/`, but that host no longer seems to be available. Recent distributions are on `ftp://metalab.unc.edu/pub/Linux/kernel/pcmcia/`. You will also need to have the kernel source code installed as well. Be sure to read the *PCMCIA−HOWTO*, which is included in the distribution.

7.2. Installing Linux Using FTP.

Most distributions are too large and complex to make FTP installation practical. Installing a basic Linux system that doesn't have a GUI or major applications is possible with FTP, however. The main non−commercial distribution in use is Debian GNU/Linux, and this answer describes an installation of a basic Debian system, to which you can add other Linux applications and commercial software as necessary.

This answer describes installation on IBM−compatible machines with an Intel x86 or Pentium processor. You will need a machine with at least a 80386 processor, 8 Mb of memory, and about 100 Mb of disk space. More memory and a larger disk is necessary however, for practical everyday use.

For other hardware, substitute ``−arm,'' ``−ppc,'' ``−m68k,'' or other abbreviation in directory names for ``−i386.''

For detailed and hardware−specific information refer to:
`http://www.debian.org/releases/stable/`.

- Connect using anonymous FTP to `ftp.debian.org` and **cd** to the `pub/debian/dists/stable/main/disks−i386/current/` subdirectory.
- Retrieve the binary image files for the rescue disk, and the drivers disk. Depending on the floppy drive installed on your machine, retrieve either the diskette images with "1200" in the names if you have a 1.2 Mb, 5.25−in. floppy, or the disks with "1440" in the name if the computer has a 3.25−in., 1.44 Mb floppy. Then retrieve the base system diskettes. Note that there are 7 base system images in the 1.44−Mb set (which have a ``14'' in their names) , and 9 in the 1.2−Mb set of images (which have a ``12'' in their names). You will use these to create the basic installation diskettes. If you have a Linux machine, you can use **dd** to write the images to the diskettes. If you are creating the installation diskettes on a MS−DOS machine, also download the `RAWRITE.EXE` MS−DOS utility, which will copy the raw binary images to floppy disks. Also download the `install.en.txt` document, which contains the detailed installation instructions.
- Create the installation disk set on floppies using either **dd** under Linux (e.g.: ``dd `if=resc1440.bin of=/dev/fd0`''), or the `RAWRITE.EXE` utility under MS−DOS. Be sure to label each installation diskette.
- Insert the rescue diskette into the floppy drive and reboot the computer. If all goes well, the Linux kernel will boot, and you will be able start the installation program by pressing **Enter** at the `boot:` prompt.
- Follow the on−screen instructions for partitioning the hard disk, installing device drivers, the basic system software, and the Linux kernel. If the machine is connected to a local network, enter the network information when the system asks for it.
- To install additional software over the Internet, be sure that you have installed the `ppp` module during the installation process, and run (as root) the `/usr/sbin/pppconfig` utility. You will need to provide your user name with your ISP, your password, the ISP's dial−up phone number, the address(es) of the ISP's Domain Name Service, and the serial port that your modem is connected to, `/dev/ttyS0`−`/dev/ttyS3`. Be sure also to specify the `defaultroute` option to the PPP system, so the computer knows to use the PPP connection for remote Internet addresses.
- You may have to perform additional configuration on the PPP scripts in the `/etc/ppp` subdirectory, and in particular, the ISP−specific script in the `/etc/ppp/peers` subdirectory. There are basic instructions in each script. For detailed information, refer to the Debian/GNU Linux installation instructions that you downloaded, the pppd manual page (type `man pppd`), and the *PPP HOWTO* from the Linux Documentation project, `http://www.linuxdoc.org/`.
- Once you have a PPP connection established with your ISP (it will be displayed in the output of ifconfig), use the dselect program to specify which additional software you want to install. Use the **apt** `[A]ccess` option to retrieve packages via anonymous FTP, and make sure to use the `[U]pdate` option to retrieve a current list of packages from the FTP archive.

7.3. Resuming an Interrupted Download.

You can use the ``reget'' command of the standard **ftp** client program after reconnecting to pick up where you left off.

Clients like **ncftp** support resumed FTP downloads, and **wget** supports resumed FTP and HTTP downloads.

7.4. Boot–Time Configuration.

You can configure Linux at the `lilo:` prompt either by typing the kernel arguments at the `BOOT lilo:` prompt, or by adding an ``append='' directive to the `/etc/lilo.conf` file; for example:

```
# At the LILO prompt (example only):
BOOT lilo: parport=0x3bc,7 parport=0x3bc,none serial=0x3f8,4 serial=0x2f8,3
```

```
# Example statement for /etc/lilo.conf:
append="parport=0x3bc,none serial=0x3f8,4 serial=0x2f8,3"
```

If you modify the `/etc/lilo.conf` file, be sure to run the **lilo** command to install the new configuration.

Configuration notes for specific hardware devices are in the documentation of the kernel source distribution, `/usr/src/linux/Documentation` in most distributions.

Refer to the **lilo** and `/etc/lilo.conf` manual pages, as well as the LDP *BootPrompt–HowTo* (``*Where Is the Documentation?*''), and the documentation in `/usr/doc/lilo`.

7.5. Formatting Man Pages without man or groff.

The **man2html** program translates **groff** text to HTML, which you can view with a Web browser. The **man2html** program, and many like it, are availble on the Web. Look for them with your favorite search engine.

The unformatted manual pages are stored in subdirectories of `/usr/man`, `/usr/local/man`, and elsewhere.

If you want to view text, use **nroff** and **less**. Both of these programs have MSDOS versions with an implementation of the *man* macro package available as well. An example would be:

```
$ nroff -man /usr/man/man1/ls.1 | less
```

If you know where to find a good implementation of the *man* macros without installing *groff*, please let the FAQ maintainer know.

If the manual page filename ends in ``.gz,'' then you'll need to uncompress it before formatting it, using **gzip -d** or **gunzip**. A one–line example would be:

```
$ gzip -dc /usr/man/man1/ls.1.gz | nroff -man | less
```

7.6. How To Scroll Backwards in Text Mode.

With the default US keymap, you can use **Shift** with the **PgUp** and **PgDn** keys. (The gray ones, not the ones on the numeric keypad.) With other keymaps, look in `/usr/lib/keytables`. You can remap the **ScrollUp** and **ScrollDown** keys to be whatever you like.

The **screen** program, `http://vector.co.jp/vpack/browse/person/an010455.html` provides a searchable scrollback buffer and the ability to take ``snapshots'' of text−mode screens.

Recent kernels that have the VGA Console driver can use dramatically more memory for scrollback, provided that the video card can *actually handle* 64 kb of video memory. Add the line:

```
#define VGA_CAN_DO_64B
```

to the start of the file `drivers/video/vgacon.c`. This feature may become a standard setting in future kernels. If the video frame buffer is also enabled in the kernel, this setting may not affect buffering.

In older kernels, the amount of scrollback is fixed, because it is implemented using the video memory to store the scrollback text. You may be able to get more scrollback in each virtual console by reducing the total number of VC's. See `linux/tty.h`.

[Chris Karakas]

7.7. How To Get Email to Work.

For sending mail via SMTP (Simple Mail Transfer Protocol) and receiving mail from an ISP's POP (Post Office Protocol) server, you can use a desktop client like Netscape Communicator or KDE kmail. You will need to enter the names of the SMTP and POP servers in the preferences of the respective application, as well as your E−mail address (username@isp's−domain−name), and your dial−up password. The same applies to Usenet News. Enter the name of the NNTP (Network News Transfer Protocol) server in your News client's preferences section. You may also have to provide the IP addresses of the ISP's primary and secondary name servers.

If you have a traditional MTA (Mail Transport Agent) like Sendmail, Smail, qmail, or Exim, you'll need to follow the instructions in each package. Basically, configuration entails determining which host machine, either on your local LAN or via dial−up Internet, is the ``Smart Host,'' if you're using SMTP. If you're using the older UUCP protocol, then you'll need to consult the directions for configuring UUCP, and also make sure that your ISP's system is configured to relay mail to you.

Information about Internet hosting, and News and E−mail in general, is available on the Usenet News group *news.announce.newusers*, and those FAQ's are also archived at `ftp://rtfm.mit.edu/pub/usenet/`.

7.8. Sendmail Pauses for Up to a Minute at Each Command.

Make sure that Sendmail can resolve your hostname to a valid (i.e., parsable) domain address. If you are not connected to the Internet, or have a dial−up connection with dynamic IP addressing, add the fully qualified

domain name to the /etc/hosts file, *in addition* to the base host name; e.g., if the host name is ``bilbo'' and the domain is ``bag-end.com:''

```
192.168.0.1  bilbo.bag-end.com bilbo
```

And make sure that either the /etc/host.conf or /etc/resolv.conf file contains the line:

```
order hosts,bind
```

Caution: Do not change the ``localhost'' entry in /etc/hosts, because many programs depend on it for internal message-passing.

Sendmail takes many factors into account when resolving domain addresses. These factors, collectively, are known as, ``rulesets,'' in sendmail jargon. The program does *not* require that a domain address be canonical, or even appear to be canonical. In the example above, ``bilbo.'' (note the period) would work just as well as ``bilbo.bag-end.com.'' This and other modifications apply mainly to recent versions.

Prior to version 8.7, sendmail required that the FQDN appear first in the /etc/hosts entry. This is due to changes in the envelope address masquerade options. Consult the sendmail documents.

If you have a domain name server for *only* a local subnet, make sure that ``.'' refers to a SOA record on the server machine, and that reverse lookups (check by using nslookup) work for all machines on the subnet.

Finally, FEATURE configuration macro options like nodns, always_add_domain, and nocanonify, control how sendmail interprets host names.

The document, *Sendmail: Installation and Operation Guide*, included in the doc/ subdirectory of Sendmail source code distributions, discusses briefly how Sendmail resolves Internet addresses. Sendmail source code archives are listed at: http://www.sendmail.org/

[Chris Karakas]

7.9. How To Enable and Select Virtual Consoles.

In text mode, press the left **Alt-F1** to **Alt-F12** to select the consoles tty1 to tty12; Right **Alt-F1** gives tty13 and so on. To switch out of X you must press **Ctrl-Alt-F1**, etc; **Alt-F5** or whatever will switch back.

However, If you have a non-PC compatible system, please see the note below.

If you want to use a VC for ordinary login, it must be listed in */etc/inittab*, which controls which terminals and virtual consoles have login prompts. The X Window System needs at least one free VC in order to start.

[Note: The key sequence is actually **Ctrl** Meta **F***N*. On PC compatible systems, the right and left **Alt** keys are really synonymous with the keysymbols Meta_L and Meta_R. If the binding is different, you can determine what keys produce Meta_L and Meta_R with xkeycaps or a similar application.]

[David Charlap]

7.10. How To Set the Time Zone.

Change directory to `/usr/lib/zoneinfo/`. Get the time zone package if you don't have this directory. The source is available in `ftp://metalab.unc.edu/pub/Linux/system/admin/time/`.

Then make a symbolic link named `localtime` pointing to one of the files in this directory (or a subdirectory), and one called `posixrules` pointing to `localtime`. For example:

```
$ ln -sf US/Mountain localtime
$ ln -sf localtime posixrules
```

This change will take effect immediately try **date**.

If the system uses Red Hat-style configuration files, the respective time zone info files are `/usr/share/zoneinfo` and `/etc/localtime`.

The manual pages for **tzset** or **tzselect** describe setting the time zone. Some programs recognize the ``TZ" environment variable, but this is not POSIX-correct.

You should also make sure that your Linux kernel clock is set to the correct GMT time. Type **date -u** and check that the correct UTC time is displayed. (``*The Computer Has the Wrong Time.*")

7.11. Dial-up PPP Configuration.

This information is mainly for people who do not have a wrapper utility like **kppp** or **pppconfig**, or are not able to get those utilities to work correctly. If you need to manually configure PPP to dial in to your ISP, you will need the following information:

- The port that your modem is connected to: `/dev/ttyS0-/dev/ttyS3`, which correspond to COM1-COM4 under MS-DOS.
- The phone number of your ISP's data connection.
- The user name and password that your ISP gave you.
- The IP addresses of the primary and possibly secondary Domain Name Service that you will use when dialing in to the ISP. This assumes that you will not be using a DNS that you installed on your system.

When you have all of this information, make sure that the programs **pppd** and **chat**, at the very minimum, are installed correctly. In most current distributions, they are installed in the `/usr/sbin/` directory, and you will need to be logged in as root to use them. In addition, the following programs are also useful for configuring network connections, determining network status, and diagnosing problems: `/sbin/ifconfig`, `/sbin/route`, `/bin/ping`, `/usr/sbin/traceroute`.

These are the basic steps that you need to follow to configure PPP. You must be logged in as root.

- Make sure that the serial port and modem are operating correctly. Using a program like **minicomm** or **kermit**, you should be able to send `AT` commands to the modem and receive the `OK` string in response from the modem.
- Enter the primary and possibly secondary Domain Name Server IP addresses in the `/etc/resolv.conf` file, using dotted quad notation, with the `nameserver` label. For example:

```
order hosts,bind
nameserver 196.182.101.103
nameserver 196.182.101.104
```

The nameserver addresses in the example above are examples only. They don't correspond to actual network hosts.

The first line, `order hosts,bind`, tells your networking software, when it resolves network domain addresses, to first look in the `/etc/hosts file`, and then use the bind service; i.e., the DNS servers, which are specified on the lines that begin with nameserver.

- Locate the chat script that PPP will use to dial the modem and connect to your ISP. In many systems, this is either in the `/etc/chatscripts` or `/etc/ppp` directory, and will be called `provider` or something similar. You can store a `chat` script anywhere, provided that you tell **pppd** to use it rather than the default script. Refer to the **chat** and **pppd** manual pages, and the information below, for details. Here is a sample `chat` script:

```
ABORT        BUSY
ABORT        "NO CARRIER"
ABORT        VOICE
ABORT        "NO DIALTONE"
""           ATDT<your_isp's_phone_number>
ogin         <your_user_name>
word         <your_password>
```

This is a **chat** program for a simple, script based login. The **chat** program uses the pair of strings on each line as a match/response pair. When it starts, it sends the string ``ATDT*your_isp's_phone_number*,'' where you have substituted the actual phone number of course. It then waits for the string `ogin` (a substring of the word `login`) and sends your user name. It then waits for `word` (a substring of `password`) and sends your password. If your ISP uses a different login and password prompts, and any additional prompts, you will need to edit the script accordingly. Again, refer to the **chat** manual page for details.

If your ISP uses PAP or CHAP authentication, you will need to edit the `pap-secrets` or `chap-secrets` files in `/etc/ppp` directory as well. Refer to the manual pages for these files, as well as the instruction in the files themselves.

- The configuration of **pppd**, the program that maintains the actual connection, is usually contained in two or three separate files. The first is usually `/etc/ppp/options`, which contains options that all of your system's PPP connections will use. (Yes, you can have more than one; as many as your computer has serial ports, generally.)

Here is a sample `/etc/ppp/options` file:

```
# /etc/ppp/options

asyncmap 0
auth
crtscts
lock
noipx

# ---<End of File>---
```

The options may be given on one line or each on a separate line. Many options files are much longer, and come with a description of each option. Here, the options mean, in order, don't remap any characters between the PPP client and server; always use password, PAP, or CHAP authentication when making a connection; use the modem's hardware handshake lines for flow control; lock the serial port when in use so no other programs can access it; and do not use the IPX network protocol.

- For connection set–up on each individual serial port or PPP host, there will either be an `/etc/ppp/options.ttyS1`, for example, options file for `/etc/ttyS1`, or a file for your ISP in the `/etc/ppp/peers` directory. The default is often called `/etc/ppp/peers/provider`. Here is a sample of the default `provider` file:

```
noauth
connect "/usr/sbin/chat -v -f /etc/chatscripts/provider"
defaultroute
/dev/ttyS1
38400
persist
```

There might be an explanation of these and other options in the `/etc/ppp/peers/provider` file itself. You can also refer to the **pppd** manual page for details. Briefly, they mean: do not use PAP authentication for this connection; use the **chat** program and the `/etc/chatscripts/provider` script, which is described above, to dial the phone and log in; set the network default route to the PPP connection (so when your network software needs to resolve an network address that is not on your local machine(s), it will use the PPP connection to the Internet); use `/dev/ttyS1` as the serial port for the connection; set the modem speed to 38400; and keep the **pppd** daemon running even if the connection fails.

- That is all of the configuration you need. To actually start and stop PPP, there are often `/usr/bin/pon` and `/usr/bin/poff` scripts (in Debian), or something similar, and they are usually very simple, and only contain the command:

```
$ /usr/sbin/pppd call ${1:-provider}
```

This will start **pppd** and use the `call` option to call the server that you type on the command line, or the provider given in the `/etc/ppp/peers/provider` file if you do not specify a remote server. After making the call and logging in (about 30 seconds), you should be able to use the `/sbin/ifconfig` program to determine that the connection really did establish a PPP interface (the first will be `ppp0`, the second will be `ppp1`, etc., depending on how many simultaneous PPP connections you have. If something goes wrong, you can look at the `/var/log/ppp.log` file to determine what happened. You can also view the log as the connection is being made, by ``tailing" it in another window; that is, viewing it as pppd logs the connection's status information. To do this, use the command (again, as root):

```
$ tail -f /var/log/ppp.log
```

On some systems the PPP output is directed to `/var/log/messages`, in which case your system may not have a dedicated PPP log file.

You should be also able to **ping** one of your ISP's domain names (e.g., `mail.isp.com`) and receive a response.

These are the most basic steps for configuring a PPP connection. You will also need to take into account what other network connections may be present (for example, if there's an Ethernet connection that has already been assigned the default route), as well as various security measures at your ISP's end. If you're having trouble making the dial–up connection, usually the best way to determine what may be going wrong is to use Seyon, minicomm, kermit, or some other program to dial and log in manually to the ISP, and determine just exactly what you have to do to log in, then duplicate that in the PPP scripts.

Most Linux documentation also has additional instructions for configuring PPP connections. Refer to (``*Where Are the Linux FTP Archives?*'') (``*Where Is the Documentation?*'')

7.12. What Version of Linux and What Machine Name Is This?

Type:

```
$ uname -a
```

7.13. What Is a ``core'' File?

A `core` file is created when a program terminates unexpectedly, due to a bug, or a violation of the operating system's or hardware's protection mechanisms. The operating system kills the program and creates a `core` file that programmers can use to figure out what went wrong. It contains a detailed description of the state that the program was in when it died.

If would like to determine what program a core file came from, use the `file` command, like this:

```
$ file core
```

That will tell you the name of the program that produced the core dump. You may want to write the maintainer(s) of the program, telling them that their program *dumped core.*

[Eric Hanchrow]

7.14. How To Enable or Disable Core Dumps.

By using the **ulimit** command in **bash**, the **limit** command in **tcsh**, or the **rlimit** command in **ksh**. See the appropriate manual page for details.

This setting affects all programs run from the shell (directly or indirectly), not the whole system.

If you wish to enable or disable core dumping for all processes by default, you can change the default setting in `linux/sched.h`. Refer to definition of `INIT_TASK`, and look also in `linux/resource.h`.

PAM support optimizes the system's environment, including the amount of memory a user is allowed. In some distributions this parameter is configurable in the `/etc/security/limits.conf` file. For more

information, refer to the *Linux Administrator's Security Guide*. (``*Where Is the Documentation?*'')

7.15. How To Upgrade/Recompile a Kernel.

See the *Kernel HOWTO* or the `README` files which come with the kernel release on `ftp.cs.helsinki.fi/pub/Software/Linux/Kernel/` and mirrors. (See ``*Where Are the Linux FTP Archives?*'') You may already have a version of the kernel source code installed on your system, but if it is part of a standard distribution it is likely to be somewhat out of date (this is not a problem if you only want a custom configured kernel, but it probably is if you need to upgrade.)

With newer kernels you can (and should) make all of the following targets. Don't forget that you can specify multiple targets with one command.

```
$ make clean dep install modules modules_install
```

Also remember to update the module dependencies.

```
$ depmod -a
```

This command can be run automatically at boot time. On Debian/GNU Linux systems, the command is part of the `/etc/init.d/modutils` script, and can be linked appropriately in the `/etc/rc`*x*`.d/` directories. For more information on **depmod**, see the manual page.

Make sure you are using the most recent version of the modutils utilities, as well as all other supporting packages. Refer to the file `Documentation/Changes` in the kernel source tree for specifics, and be sure to consult the `README` file in the modutils package.

Remember that to make the new kernel boot you must run **lilo** after copying the kernel into your root partition. The Makefile in some kernels have a special `zlilo` target for this; try:

```
$ make zlilo
```

On current systems, however, you can simply copy the `zImage` or `bzImage` file (in `arch/i386/boot/` to the `/boot/` directory on the root file system, or to a floppy using the **dd** command. Refer also to the question, How do I get LILO to boot the kernel image?

Kernel version numbers with an odd minor version (ie, 1.1.x, 1.3.x) are the testing releases; stable production kernels have even minor versions (1.0.x, 1.2.x). If you want to try the testing kernels you should probably subscribe to the linux-kernel mailing list. (``*What Mailing Lists Are There?*'')

The Web site `http://www.kernelnotes.org/` has lots of information and links to other sites that provide information about Linux kernel updates.

Also refer to the questions, ``*The PCMCIA Card Doesn't Work after Upgrading the Kernel.*'' and ``*How To Get LILO to Boot the Kernel Image.*''

7.16. Can Linux Use More than 3 Serial Ports by Sharing Interrupts?

Yes, but you won't be able to use simultaneously two ordinary ports which share an interrupt (without some trickery). This is a limitation of the ISA Bus architecture.

See the *Serial HOWTO* for information about possible solutions and workarounds for this problem.

7.17. Configuring Emacs's Default Settings.

Create a file in your home directory named `.emacs` with the Emacs Lisp commands that you want to run every time Emacs starts up. You won't see the file in the directory listing. (The leading '.' tells **ls** not to display it, unless you use the `-a` command line switch with **ls**.)

Any kind of Emacs Lisp statement will work in the `.emacs` file, including entire `defuns`. Emacs uses lisp variables and statements extensively, and many of the editing functions are written in Emacs Lisp. For example, to enable word wrapping whenever you edit a file that ends with `.txt`, add the following statement. This is from the Emacs Texinfo help document (**F1–i**, then **m** `Emacs` **Return**):

```
(add-hook text-mode-hook
        '(lambda () (auto-fill-mode 1)))
```

This adds a statement that calls a *hook* function whenever a text editing mode is entered for that buffer. The value of `text-mode-hook`, which is a variable, to `auto-fill-mode`, which is a function.

If you want to turn off the menu bar at the top of each Emacs frame, add this statement:

```
(menu-bar-mode -1)
```

And if you want to include an Emacs Lisp program that someone has written, like `msb.el` (an enhanced, pop–up buffer menu), make sure the lisp file is in a directory where Emacs can find it (usually it will be named Site–lisp), and add these statements in the `.emacs` file:

```
(require 'msb)
(msb-mode 1)
```

Most tasks have several possible solutions in Emacs Lisp. Any task that can be programmed in Emacs Lisp is valid in the `.emacs` file. For more information, consult the Texinfo documentation. There is also a FAQ list for Emacs (refer to: What other FAQ's are there for Linux?).

7.18. How To Make a Rescue Floppy.

Make a file system on it with bin, etc, lib and dev directories everything you need. Install a kernel on it and arrange to have LILO boot it from the floppy (see the LILO documentation, in `lilo.u.*.ps`).

If you build the kernel (or tell LILO to tell the kernel) to have a RAM disk the same size as the floppy the RAM disk will be loaded at boot time and mounted as root in place of the floppy.

See the *Bootdisk HOWTO*.

7.19. How To Remap a Keyboard to UK, French, Etc.?

For recent kernels, get `/pub/Linux/system/Keyboards/kbd-0.90.tar.gz` from `ftp://metalab.unc.edu/`. Make sure you get the appropriate version; you have to use the right keyboard mapping package for your kernel version. For older kernels you have to edit the top–level kernel `Makefile`, in `/usr/src/linux/`. You may find more helpful information in *The Linux Keyboard and Console HOWTO*, by Andries Brouwer, at `ftp://metalab.unc.edu/pub/Linux/docs/HOWTO/`.

7.20. How To Get NUM LOCK to Default to On.

Use the **setleds** program, for example (in `/etc/rc.local` or one of the `/etc/rc.d/*` files):

```
for t in 1 2 3 4 5 6 7 8
do
 setleds +num < /dev/tty$t > /dev/null
done
```

setleds is part of the kbd package ("How do I remap my keyboard to UK, French, etc.? "). Alternatively, patch your kernel. You need to arrange for `KBD_DEFLEDS` to be defined to (`1 << VC_NUMLOCK`) when compiling `drivers/char/keyboard.c`.

7.21. How To Set (Or Reset) Initial Terminal Colors.

The following shell script should work for VGA consoles:

```
for n in 1 2 4 5 6 7 8; do
     setterm -fore yellow -bold on -back blue -store > /dev/tty$n
     done
```

Substitute your favorite colors, and use `/dev/ttyS$n` for serial terminals.

To make sure they are reset when people log out (if they've been changed):

Replace the references to `getty` (or `mingetty` or `uugetty` or whatever) in `/etc/inittab` with references to `/sbin/mygetty`.

```
#!/bin/sh
setterm -fore yellow -bold on -back blue -store > $1
exec /sbin/mingetty $@
```

[Jim Dennis]

7.22. How To Have More Than 128Mb of Swap.

Use several swap partitions or swap files. Linux kernels before version 2.2 supported up to 16 swap areas, each of up to 128Mb. Recent versions do not have this limitation.

Very old kernels only supported swap partition sizes up to 16Mb.

Linux on machines with 8KB paging, like Alpha and Sparc64, support a swap partition up to 512MB. The 128MB limitation comes from `PAGE_SIZE*BITSPERBYTE` on machines with 4KB paging, but is 512KB on machines with 8KB paging. The limit is due to the use of a single page allocation map.

The file `mm/swapfile.c` has all of the gory details.

[Peter Moulder, Gordon Weast]

7.23. How To Prevent Errors when Linking Programs with Math Functions.

Older run−time libraries included the math library in the C run−time library. It was not necessary to specify the math library separately when compiling. If the compiler generates a message like this when linking a program that uses math functions:

```
/tmp/ccDUQM4J.o: In function `main':
/tmp/ccDUQM4J.o(.text+0x19): undefined reference to `sqrt'
collect2: ld returned 1 exit status
```

You need use the `−lm` option with GCC to link with the math libraries:

```
# gcc −o program program.c −lm
```

Make sure also to use the statement `#include <math.h>` in the source file.

[Florian Schmidt]

---

8. Miscellaneous Information and Questions Answered

8.1. How To Program *XYZ* Under Linux.

Read the manuals, or a good book on Unix and the manual pages (type `man man`). There is a lot of GNU Info documentation, which is often more useful as a tutorial. Run Emacs and type **F1–i**, or type **info info** if you don't have or don't like Emacs. Note that the Emacs libc node may not exactly describe the latest Linux libc, or GNU glibc2. But the GNU project and LDP are always looking for volunteers to upgrade their library documentation.

Anyway, between the existing Texinfo documentation, and the manual pages in sections 2 and 3, should provide enough information to get started.

As with all free software, the best tutorial is the source code itself.

The latest release of the Linux manual pages, a collection of useful GNU Info documentation, and various other information related to programming Linux, can be found on `metalab.unc.edu/pub/Linux/docs/man-pages/`.

8.2. What's All This about ELF? glibc?

See the *ELF HOWTO* by Daniel Barlow. Note that this is not the file `move-to-elf`, which is a blow–by–blow account of how to upgrade to ELF manually.

Linux has two different formats for executables, object files, and object code libraries, known as, ``ELF.'' (The old format is called ``a.out.'') They have advantages, including better support for shared libraries and dynamic linking.

Both a.out and ELF binaries can coexist on a system. However, they use different shared C libraries, both of which have to be installed.

If you want to find out whether your system can run ELF binaries, look in `/lib` for a file named, ``libc.so.5.'' If it's there, you probably have ELF libraries. If you want to know whether your installation actually is ELF you can pick a representative program, like **ls**, and run file on it:

```
-chiark:~> file /bin/ls
/bin/ls: Linux/i386 impure executable (OMAGIC) - stripped

valour:~> file /bin/ls
/bin/ls: ELF 32-bit LSB executable, Intel 80386, version 1, stripped
```

There is a patch to get 1.2.x to compile using the ELF compilers, and produce ELF core dumps, at `ftp://tsx-11.mit.edu/pub/packages/GCC/`. You do not need the patch merely to run ELF binaries. 1.3.x and later do not need the patch at all.

The GNU glibc2 libraries are essentially more recent versions of ELF libraries that follow most of the same

processes for dynamic linking and loading. Upgrade information is contained in (``*How To Upgrade the Libraries without Trashing the System.*'')

8.3. How To Determine What Packages Are Installed on a System.

For distributions that use RPM format packages, use the command:

```
$ rpm -qa
```

You need to be logged in as root. You can save the output to a text file for future reference, a command like:

```
$ rpm -qa >installed-packages
```

For Debian systems, the equivalent command is:

```
$ dpkg -l
```

8.4. What Is a `.gz` File? And a `.tgz`? And `.bz2`? And... ?

`.gz` (and `.z`) files are compressed using GNU **gzip**. You need to use **gunzip** (which is a symlink to the **gzip** command that comes with most Linux installations) to unpack the file.

`.taz`, `.tar.Z`, and `.tz` are *tar* files (made with **tar**) and compressed using **compress**. The standard *nix **compress** is proprietary software, but free equivalents like ncompress exist.

`.tgz` (or `.tpz`) is a tar file compressed with **gzip**.

`.bz2` is a file compressed by the more recently introduced (and efficient) **bzip2**.

`.lsm` is a *Linux Software Map* entry, in the form of a short text file. Details about the LSM project and the LSM itself are available in the subdirectory on `ftp://metalab.unc.edu/pub/Linux/docs/`.

`.deb` is a Debian Binary Package the binary package format used by the Debian GNU/Linux distribution. It is manipulated using **dpkg** and **dpkg–deb** (available on Debian systems and from `ftp://ftp.debian.org//`).

`.rpm` is a Red Hat RPM package, which is used in the Red Hat and similar distributions.

`.sit` is a compressed Macintosh archive made with StuffIt, a commercial program. Aladdin Systems Inc., the manufacturer of StuffIt, has a free expander utility that will uncompress these archives. You can download it at `http://www.aladdinsys.com/expander/`.

The **file** command can often tell you what a file is.

If you find that **gzip** complains when you try to uncompress a file, you probably downloaded it in ASCII

mode by mistake. You must download most things in binary mode: ``get,'' to download the file.

8.5. What Does VFS Stand For?

Virtual File System. It's the abstraction layer between the user and real file systems like *ext2*, *Minix* and *MS−DOS*. Among other things, its job is to flush the read buffer when it detects a disk change on the floppy disk drive.

```
VFS: Disk change detected on device 2/0
```

8.6. What is a BogoMip?

``BogoMips'' is a combination of *Bogus* and *Mips*. MIPS stands for (depending on who you ask) *Millions of Instructions per Second*, or *Meaningless Indication of Processor Speed*.

The number printed at boot time is the result of a kernel timing calibration, used for very short delay loops by some device drivers.

According to the *BogoMips mini−HOWTO*, the rating for your machine will be:

```
                    Common BogoMips Ratings

Processor                   BogoMips         Comparison
---------                   --------         ----------
Intel 8088                  clock * 0.004    0.02
Intel/AMD 386SX             clock * 0.14     0.8
Intel/AMD 386DX             clock * 0.18     1 (definition)
Motorola 68030              clock * 0.25     1.4
Cyrix/IBM 486               clock * 0.34     1.8
Intel Pentium               clock * 0.40     2.2
Intel 486                   clock * 0.50     2.8
AMD 5x86                    clock * 0.50     2.8
Mips R4000/R4400            clock * 0.50     2.8
Nexgen Nx586                clock * 0.75     4.2
PowerPC 601                 clock * 0.84     4.7
Alpha 21064/21064A          clock * 0.99     5.5
Alpha 21066/21066A          clock * 0.99     5.5
Alpha 21164/21164A          clock * 0.99     5.5
Intel Pentium Pro           clock * 0.99     5.5
Cyrix 5x86/6x86             clock * 1.00     5.6
Intel Pentium II/III        clock * 1.00     5.6
Intel Celeron               clock * 1.00     5.6
Mips R4600                  clock * 1.00     5.6
Alpha 21264                 clock * 1.99     11.1
AMD K5/K6/K6-2/K6-III       clock * 2.00     11.1
UltraSparc II               clock * 2.00     11.1
Pentium MMX                 clock * 2.00     11.1
PowerPC 604/604e/750        clock * 2.00     11.1
Motorola 68060              clock * 2.01     11.2
Motorola 68040              Not enough data (yet).
AMD Athlon                  Not enough data (yet).
IBM S390                    Not enough data (yet).
```

If the number is wildly lower, you may have the Turbo button or CPU speed set incorrectly, or have some kind of caching problem (as described in ``*When Adding More Memory, the System Slows to a Crawl.*")

For values people have seen with other, rarer, chips, or to calculate your own BogoMips rating, please refer to the *BogoMips Mini−HOWTO*, on `ftp://metalab.unc.edu/`. (``*Where Is the Documentation?*")

[Wim van Dorst]

8.7. What Online/Free Periodicals Exist for Linux?

There are a number of recent additions to the list of periodicals devoted to Linux and free software:

- *geek news*. `http://geeknews.cjb.net/`. Headlines for articles about Linux, like the *comp.os.linux.announce* and Techweb postings, and general interest, like Associated Press stories.
- *Linux Gazette*. `http://www.linuxgazette.com/`. This is the longest−running of the on−line periodicals, and the only one that publishes source code.
- *Linux Today*. `http://www.linuxtoday.com`. News and opinion related to the Linux community, updated daily.
- *Linux Weekly News*. `http://lwn.net`. News about the Linux community, updated weekly.
- *Slashdot*. `http://www.slashdot.org`. News about the free software community and culture.
- *Freshmeat*. `http://www.freshmeat.net/`. Notices of new and updated software for Linux and other free OS's.

Please send additions to this list to the FAQ maintainer.

[Jim Dennis, Robert Kiesling]

8.8. How Many People Use Linux?

Linux is freely available, and no one is required to register with any central authority, so it is difficult to know. Several businesses survive solely on selling and supporting Linux. Linux news groups are some of the most heavily read on Usenet. Accurate numbers are hard to come by, but the number is almost certainly in the millions.

However, people can register as Linux users at the Linux Counter project, which has been in existence since 1993. In August, 1998, the project counted more than 70,000 users.

Visit the Web site at `http://counter.li.org/` and fill in the registration form. If you don't want to use the Web, send E−mail to `counter@counter.li.org` with the subject line, ``I use Linux at home,'' or ``I use Linux at work.''

The current count is posted monthly to *comp.os.linux.misc*, and is always available from the Web site.

[Harald Tveit Alvestrand]

8.9. How Many People Use Linux? (Redux.)

In 1999, International Data Corporation released its first commercial forecast of Linux sales. The report quantifies Linux vendor sales in 1996, 1997, and 1998, and forecasts through the year 2003.

To obtain the report, contact IDC at `ctoffel@idc.com`. Their Web site is `http://www.itresearch.com/`.

8.10. What Is the Best (Distribution|SCSI Card|Editor|CD–ROM Drive|....)

The ``best'' of anything depends on your particular needs. Discussions like these frequently occur on Usenet. Most often they're flame bait. Answering is generally a waste of time. Free software licensing is unrestrictive enough, that, with a little experience, you can perform your own testing on your own hosts.

A better way to phrase a specific inquiry might be: ``Where can I find....''

8.11. How Does One Pronounce Linux?

This question produces an outrageous amount of heated debate.

If you want to hear Linus himself say how he pronounces it, download `english.au` or `swedish.au` from `ftp.funet.fi/pub/Linux/PEOPLE/Linus/SillySounds/`. If you have a sound card or the PC–speaker audio driver you can hear them by typing

```
$ cat english.au >/dev/audio
```

The difference isn't in the pronunciation of Linux but in the language Linus uses to say, ``hello.''

For the benefit of those who don't have the equipment or inclination: Linus pronounces Linux approximately as Leenus, where the ``ee'' is pronounced as in ``feet,'' but rather shorter, and the ``u'' is like a much shorter version of the French ``eu'' sound in ``peur'' (pronouncing it as the ``u'' in ``put'' is probably passable).

---

9. Frequently Encountered Error Messages

9.1. Modprobe Can't Locate Module, *XXX*, and Similar Messages.

These types of messages mostly occur at boot time or shutdown. If modprobe, insmod, or rmmod complain about not being able to find a module, add the following to the `/etc/modules.conf` or `/etc/modutils/aliases` file, whichever is present on your system.

```
$ alias <module-name> off
```

And use the name of the module that appears in the error message.

[J.H.M. Dassen]

9.2. Unknown Terminal Type ``linux'' and Similar.

In early kernels the default console terminal type has changed from ``console'' to ``linux.'' You must edit `/etc/termcap` to change the line reading:

```
console|con80x25:\
```

to

```
linux|console|con80x25:\
```

(there may be an additional ``dumb'' in there if so it should be removed.)

To get the editor to work you may need type:

```
$ TERM=console
```

(for **bash** and **ksh**), or

```
$ setenv TERM console
```

for **csh** or **tcsh**.

Some programs use `/usr/lib/terminfo` instead of `/etc/termcap`. For these programs you should upgrade your terminfo package, which is part of ncurses.

The same is true for X terminal displays. If your distribution sets the `TERM` to something strange like `xterm-24-color`, you can simply reset it to a generic value from the command line:

```
$ TERM="xterm"; export TERM
```

9.3. INET: Warning: old style ioctl... called!

You are trying to use the old network configuration utilities. The new ones can be found on `ftp.linux.org.uk/pub/linux/Networking/PROGRAMS/NetTools/` (source only, I'm afraid).

Note that they cannot be used just like the old−style programs. See the *NET−2 HOWTO* for instructions on how to set up the old−style networking programs correctly. Even better, see the *NET−3 HOWTO* and upgrade your networking software.

9.4. ld: unrecognized option '−m486'

You have an old version of **ld**. Install a newer binutils package that contains an updated **ld**. Look on tsx−11.mit.edu in `/pub/linux/packages/GCC/` for `binutils−2.6.0.2.bin.tar.gz`.

9.5. GCC Says, ``Internal compiler error."

If the fault is repeatable (i.e., it always happens at the same place in the same file even after rebooting and trying again, using a stable kernel) you have discovered a bug in GCC. See the GCC Info documentation (type **F1−i** in Emacs, and select GCC from the menu) for details on how to report the error. Make sure you have the latest version, though.

Note that this is probably not a Linux−specific problem. Unless you are compiling a program many other Linux users also compile, you should not post your bug report to any of the comp.os.linux groups.

If the problem is not repeatable, you may be experiencing memory corruption. Refer to the answer: (``*Make Says, ``Error 139.'*")

9.6. Make Says, ``Error 139."

Your compiler (GCC) dumped core. You probably have a corrupted, buggy, or old version of GCC get the latest release or EGCS. Alternatively, you may be running out of swap space. Refer to: (``*The Machine Runs Very Slowly with GCC / X / ...*")

If this doesn't fix the problem, you are probably having problems with memory or disk corruption. Check that the clock rate, wait states, and refresh timing for your SIMMS and cache are correct (hardware manuals are sometimes wrong, too). If so, you may have some marginal SIMMS, or a faulty motherboard or hard disk or controller.

Linux is a very good memory tester much better than MS−DOS based memory test programs.

Reportedly, some clone x87 math coprocessors can cause problems. Try compiling a kernel with math emulation (``*How To Upgrade/Recompile a Kernel.*") `no387` kernel command line flag on the LILO prompt to force the kernel to use math emulation, or it may be able to work and still use the '387, with the math emulation compiled in but mainly unused.

More information about this problem is available on the Web at `http://www.bitwizard.nl/sig11/`.

9.7. Shell–Init: Permission Denied when I Log In.

Your root directory and all the directories up to your home directory must be readable and executable by everybody. See the manual page for **chmod** or a book on Unix for how to fix the problem.

9.8. No Utmp Entry. You Must Exec ... when Logging In.

Your `/var/run/utmp` is screwed up. You should have

```
/var/run/utmp
```

in your `/etc/rc.local` or `/etc/rc.d/*`. See, (``*I Screwed Up the System and Can't Log In to Fix It.*'') Note that the `utmp` may also be found in `/var/adm/` or `/etc/` on some older systems.

9.9. Warning––bdflush Not Running.

Modern kernels use a better strategy for writing cached disk blocks. In addition to the kernel changes, this involves replacing the old **update** program which used to write everything every 30 seconds with a more subtle daemon (actually a pair), known as **bdflush**. Get `bdflush-`*n.n*`.tar.gz` from the same place as the kernel source code (``*How To Upgrade/Recompile a Kernel.*'') and compile and install it. **bdflush** should be started before the usual boot–time file system checks. It will work fine with older kernels as well, so there's no need to keep the old **update** around.

9.10. `Warning: obsolete routing request made.`

This is nothing to worry about. The message means that your version **route** is a little out of date, compared to the kernel. You can make the message go away by getting a new version of **route** from the same place as the kernel source code. (``*How To Upgrade/Recompile a Kernel.*'')

9.11. `EXT2-fs: warning: mounting unchecked file system.`

You need to run **e2fsck** (or `fsck -t ext2` if you have the **fsck** front end program) with the **–a** option to get it to clear the ``dirty'' flag, and then cleanly unmount the partition during each shutdown.

The easiest way to do this is to get the latest **fsck**, **umount**, and **shutdown** commands, available in Rik Faith's `util-linux` package (``*Where Are the Linux FTP Archives?*'') You have to make sure that your `/etc/rc*/` scripts use them correctly.

NB: Don't try to check a file system that's mounted read/write. This includes the root partition if you don't see

```
VFS: mounted root ... read-only
```

at boot time. You must arrange to mount the root file system read/only to start with, check it if necessary, and then remount it read/write. Almost all distributions do this. If your's doesn't, read the documentation that comes with `util-linux` to find out how to do this.

Note that you need to specify the **-n** option to **mount** so it won't try to update `/etc/mtab`, since the root file system is still read-only, and this will otherwise cause it to fail.

9.12. EXT2-fs warning: maximal count reached.

This message is issued by the kernel when it mounts a file system that's marked as clean, but whose "number of mounts since check" counter has reached the predefined value. The solution is to get the latest version of the ext2fs utilities (`e2fsprogs-0.5b.tar.gz` at the time of writing) from the usual sites. (``*Where Are the Linux FTP Archives?*'')

The maximal number of mounts value can be examined and changed using the **tune2fs** program from this package.

9.13. EXT2-fs warning: checktime reached.

Kernels from 1.0 onwards support checking a file system based on the elapsed time since the last check as well as by the number of mounts. Get the latest version of the ext2fs utilities. (``*EXT2-fs warning: maximal count reached.*'')

9.14. df Says, ``Cannot read table of mounted file systems.''

There is probably something wrong with your `/etc/mtab` or `/etc/fstab` files. If you have a reasonably new version of mount, `/etc/mtab` should be emptied or deleted at boot time (in `/etc/rc.local` or `/etc/rc.d/*`), using something like

```
$ rm -f /etc/mtab*
```

Some old Linux distributions have an entry for the root partition in `/etc/mtab` made in `/etc/rc*` by using **rdev**. That is incorrect the newer versions of **mount** do this automatically.

Some old distributions also have a line in `/etc/fstab` that looks like:

```
/dev/sdb1   /root   ext2   defaults
```

The entry for `/root` should read simply `/`.

9.15. fdisk Says, ``Partition X has different physical/logical..."

If the partition number (X, above) is 1, this is the same problem as in `fdisk: Partition 1 does not start on cylinder boundary`. If the partition begins or ends on a cylinder numbered greater than 1024, this is because the standard DOS disk geometry information format in the partition table can't cope with cylinder numbers with more than 10 bits. You should see (``*How To Get Linux to Work with a Disk.*")

9.16. fdisk: Partition 1 does not start on cylinder boundary.

The version of **fdisk** that comes with many Linux systems creates partitions that fail its own validity checking. Unfortunately, if you've already installed your system, there's not much you can do about this, apart from copying the data off the partition, deleting and remaking it, and copying the data back.

You can avoid the problem by getting the latest version of **fdisk**, from Rik Faith's util−linux package (available on all the usual FTP sites). Alternatively, if you are creating a new partition 1 that starts in the first cylinder, you can do the following to get a partition that **fdisk** likes.

- Create partition 1 in the normal way. A `p' listing will produce the mismatch complaint.
- Type **u** to set sector mode and do **p** again. Copy down the number from the `End` column.
- Delete partition 1.
- While still in sector mode, re−create partition 1. Set the first sector to match the number of sectors per track. This is the sector number in the first line of the **p** output. Set the last sector to the value you wrote down in the step above.
- Type **u** to reset cylinder mode and continue with other partitions.

Ignore the message about unallocated sectors. They refer to the sectors on the first track apart from the Master Boot Record, and they are not used if you start the first partition in track 2.

9.17. fdisk Says Partition *n* Has an Odd Number of Sectors.

The PC disk partitioning scheme works in 512−byte sectors, but Linux uses 1K blocks. If you have a partition with an odd number of sectors, the last sector is wasted. Ignore the message.

9.18. Mtools Utilities Say They Cannot Initialize Drive *X*.

This means that mtools is having trouble accessing the drive. This can be due to several things.

Often this is due to the permissions on floppy drive devices (`/dev/fd0*` and `/dev/fd1*`) being incorrect. The user running mtools must have the appropriate access. See the manual page for **chmod** for details.

Most versions of mtools distributed with Linux systems (not the standard GNU version) use the contents of a file `/etc/mtools` to determine which devices and densities to use, in place of having this information

compiled into the binary. Mistakes in this file often cause problems. There is often no documentation about this.

For the easiest way to access your MS−DOS files (especially those on a hard disk partition) see How do I access files on my DOS partition or floppy? Note you should never use mtools to access files on an msdosfs mounted partition or disk!

9.19. At the Start of Booting: `Memory tight`

This means that you have an extra−large kernel, which means that Linux has to do some special memory−management magic to be able to boot itself from the BIOS. It isn't related to the amount of physical memory in your machine. Ignore the message, or compile a kernel containing only the drivers and features you need. (``*How To Upgrade/Recompile a Kernel.*'')

9.20. The System Log Says, ``end_request: I/O error, ...''

This error message, and messages like it, almost always indicate a hardware error with a hard drive.

This commonly indicates a hard drive defect. The only way to avoid further data loss is to completely shut own the system. You must also make sure that whatever data is on the drive is backed up, and restore it to a non−defective hard drive.

This error message may also indicate a bad connection to the drive, especially with home brew systems. If you install an IDE drive, *always* use new ribbon cables. It's probably is a good idea with SCSI drives, too.

In one instance, this error also seemed to coincide with a bad ground between the system board and the chassis. Be sure that all electrical connections are clean and tight before placing the blame on the hard drive itself.

[Peter Moulder, Theodore Ts'o]

9.21. ``You don't exist. Go away.''

This is not a viral infection. It comes from programs like **write**, **talk**, and **wall**, if your invoking UID doesn't correspond to a valid user (probably due to `/etc/passwd` being corrupted), or if the session (pseudoterminal, specifically) you're using isn't properly registered in the `utmp` file (probably because you invoked it in a funny way).

9.22. ``Operation not permitted.''

One or more of the file's or directory's attribute bits are set incorrectly. If the ``I'' bit is set, for example, you won't be able to change file permissions with **chmod**.

The solution is to use **lsattr** to display file and directory attributes, and **chattr** to set and unset them. The programs' documentation is contained in their manual pages.

[Paul Campbell]

9.23. *programname:* error in loading shared libraries: lib *xxx*..so. *x*: cannot open shared object file: No such file or directory.

A message like this, when the program that you're trying to run uses shared libraries, usually means one of two things: the program was either compiled on a machine that had a different set of libraries or library paths than yours; or you've upgraded your libraries but not the program.

Executable programs that are linked with dynamic libraries, expect the full pathname of each of the library files it requires. So do the shared libraries, if they rely on other libraries. This is so the shared object dependencies remain as unambiguous as possible, and also as a security measure.

Short of recompiling the executable file for the libraries on the system probably the most desirable alternative in the long run you can try to determine which libraries the executable file needs with the command: ``ldd *programname*." The output will be a list of the shared libraries on the system that the program needs to run, as well as the missing libraries. You can then add the library packages, or if the libraries already exist in a different directory, you can create a symbolic link so the program can find it. For example, if the program requires /usr/lib/libncurses.so.2, and your machine has /lib/libncurses.so.2, you can create a link where the program expects to find the library; e.g.:

```
# cd /usr/lib && ln -s /lib/libncurses.so.2 .
```

You should note, however, that creating library links like these should be considered a security risk, and the additional links you create will not be compatible with future upgrades. It's simply a quick fix for backward compatibility.

Also, it may take some guesswork to determine in exactly which of the system library directories the program expects to find a shared library file, because `ldd` will not list the paths of libraries it can't find. A program most likely will tell the run-time linker, `/lib/ld.so`, to look for shared libraries in `/lib`, `/usr/lib`, `/usr/local/lib`, or `/usr/X11R6/lib`, if it's an X client. But that doesn't mean that libraries can't be installed elsewhere. It helps to have some idea of the original library configuration before proceeding.

Also be sure to run `ldconfig` after creating the symbolic link, so that `ld.so` has an updated view of the system's libraries. You should also make certain that all of the library directories are listed in `/etc/ld.so.conf`, and perhaps in the `LD_LIBRARY_PATH` environment variable.

9.24. ``init: Id "x" respawning too fast: disabled for 5 minutes."

In most distributions this means that the system is booting by default into runlevel 5, which is supposed to respawn (re-start again after it's been exited) a graphical login via xdm, kdm, gdm, or whatever, and the

system can't locate the program.

However, ``Id'' can also indicate the absence or misconfiguration of another program, like mingetty, if init tries to respawn itself more than 10 times in 2 minutes.

Id ``x'' is the number in the leftmost column of the /etc/inittab file:

```
# Run gettys in standard runlevels
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6
```

Commenting the offending line out and then fixing the errant program and testing on the command line will allow you to see any error messages that go to standard error output (console) if the errors are not going to the system log file. Uncomment the line and restart init with ``kill -SIGHUP 1'' or ``telinit q'' to cause init to reinitialize and reread the /etc/inittab file.

Some systems, however, rewrite /etc/inittab when booting. In that case, refer to the init man page, and/or the settings in /etc/sysconfig/init.

Refer to the init and /etc/inittab man pages for detailed information.

[Carl King]

9.25. FTP server says: ``421 service not available, remote server has closed connection.''

If an FTP server won't allow logins, it is probably configured correctly, but the problem is probably with authorizing users at login. FTP servers in current distriubtions often authorize users with the Pluggable Authentication Modules library, in which case there should be an authorization file /etc/pam.d/ftp. A generic authorization file looks like this. (The line break on the first ``auth'' line is for readability. The entry is actually a single, long line).

```
#%PAM-1.0
auth       required  /lib/security/pam_listfile.so item=user \
           sense=deny file=/etc/ftpusers onerr=succeed
auth       required  /lib/security/pam_pwdb.so shadow nullok
auth       required  /lib/security/pam_shells.so
account    required  /lib/security/pam_pwdb.so
session    required  /lib/security/pam_pwdb.so
```

Also, make sure the /etc/ftpusers file, or whatever users file is named in the first ``auth'' line, is configured correctly.

Btw, the sample ftp file above is actually theftpd/ftp.pam.sample file from the ftpd-BSD-0.3.1.tar.gz package. Many thanks to David A. Madore for this much needed port.

---

10. The X Window System

10.1. Does Linux Support X?

Yes. Linux uses XFree86 (the current version is 4.0, which is based on X11R6). You need to have a video card which is supported by XFree86. See the *XFree86 HOWTO* for more details. Most Linux distributions nowadays come with an X installation. However, you can install or upgrade your own, from `"ftp://metalab.unc.edu/pub/Linux/X11/Xfree86-*"` and its mirror sites, or from `http://www.xfree86.org/`.

10.2. How To Get the X Window System to Work.

The answers to this question can, and do, fill entire books. If the installation program wasn't able to configure the X server correctly, Linux will most likely try to start the X display, fail, and drop back into text–only terminal mode.

First and foremost, make certain that you have provided, as closely as possible, the correct information to the installation program of your video hardware: the video card and monitor. Some installation programs can correctly guess a ``least common denominator'' screen configuration, like a 640–by–480 VESA–standard display, but there are many possible video hardware configurations that may not be able to display this standard.

The X Window System configuration file is called (usually) `/etc/XF86Config`, `/etc/X11/XF86Config`, or `/usr/X11R6/lib/X11/XF86Config`.

If you need to manually configure the X server, there are several possible methods:

- Try to use the **XF86Setup** program, which can help identify the correct X server and monitor timings for the video hardware.
- Make sure that the X server has the correct options. If you log in as the superuser, you should be able to use `X --probeonly` to get a listing of the video card chipset, memory, and any special graphics features. Also, refer to the manual page for the X server. (E.g.; `man X`), and try running the X server and redirecting the standard error output to a file so you can determine, after you can view text on the screen again, what error messages the server is generating; e.g., `X 2>x.error`.
- With that information, you should be able to safely refer to one of the references provided by the Linux Documentation Project. ("Where can I get the HOWTO's and other documentation? ") There are several HOWTO's on the subject, including a HOWTO to calculate video timings manually if necessary. Also, the *Installation and Getting Started* guide has a chapter with a step–by–step guide to writing a `XF86Config` file.

Also, make sure that the problem really is an incorrect `XF86Config` file, not something else like the window manager failing to start. If the X server is working correctly, you should be able to move the mouse cursor on the screen, and pressing **Ctrl–Alt–Backspace** will shut down the X server and return to the shell prompt in one of the virtual terminals.

10.3. Where To Find a Ready–Made `XF86Config` file.

If you can't seem to get X working using the guidelines above, refer to the *XFree86 HOWTO*, recent versions of *Installation and Getting Started*, and the instructions for the `XF86Setup` program. The contents of the `XF86Config` file depend on the your exact combination of video card and monitor. It can either be configured by hand, or using the **XF86Setup** utility. Read the instructions that came with XFree86, in `/usr/X11R6/lib/X11/etc`. The file you probably need to look at most is `README.Config`. You should not use the sample `XF86Config.eg` file which is included with newer versions of XFree86 verbatim, because the wrong video clock settings can damage your monitor. Please don't post to *comp.os.linux.x* asking for an `XF86Config`, and please don't answer such requests. If you have a laptop, look at the Linux Laptop Web page (``*How To Find Out If a Notebook Runs Linux.*'') Many of the installation notes also have the `XF86Config` file for the display. If you have a desktop machine, there are a few sample `XF86Config` files at `ftp://metalab.unc.edu/`. Refer also to the *XFree86 FAQ* `http://www.xfree.org/FAQ/` and the monitor timings list `http://www.xfree.org/#resources/`, and in the `/usr/X11R6/lib/X11/` directory of your X distribution.

10.4. What Desktop Environments Run on Linux?

Linux with XFree86 supports the KDE, GNOME, and commercial CDE desktop environments, and extended window managers like WindowMaker. Each uses a different set of libraries and provides varying degrees of MS Windows–like look and feel.

Information on KDE is available from `http://www.kde.org/`. The KDE environment uses the Qt graphics libraries, available from `http://www.qt.org/`. The desktop uses its own window manager, kwm, and provides a MS Windows–like look and feel.

The GNOME home page is `http://www.gnome.org/`. The environment uses the free GTK libraries, available from `http://www.gtk.org/`, and window managers like Enlightenment, `http://www.enlightenment.org/`, SawFish, `http://www.sawfish.org/`. There's also a Web page for GNOME installation and upgrade that functions much like Debian's **apt–get** utility with a friendly GUI front end. It's at: `http://www.helixcode/com/desktop/`.

The commercial CDE environment uses the Motif libraries and a variation of the Motif **mwm** window manager, **dtwm**, and provides a suite of desktop and session–management utilities. Several vendors have made the source code of Motif available and provided binary packages for Linux distributions. As a starting point, download and installation information is available at `http://www.opengroup.org/openmotif/`.

A free version of Motif, called LessTiF, is available from `http://www.lesstif.org/`.

WindowMaker, `http://www.windowmaker.org/` is a window manager that has many desktop environment–like features. It provides support for GNUstep, `http://www.gnustep.org/`, a clone of the commercial NeXTStep environment.

10.5. xterm Logins Show Up Strangely in who, finger.

The **xterm** that comes with XFree86 2.1 and earlier doesn't correctly understand the format that Linux uses for the `/var/adm/utmp` file, where the system records who is logged in. It therefore doesn't set all the information correctly. The xterms in XFree86 3.1 and later versions fix this problem.

10.6. How to Start a X Client on Another Display.

To start a X client on another system that has a running X server, use the following commands:

- Use xhost on the server system to allow the client system use the display. If the server's IP address is 192.168.20.1, enter the command:

```
$ xhost + 192.168.20.1
```

- On the client system, open a telnet connection to the server system.
- In the telnet session, start a xterm in the background with the `-display` and `-e` options. For example, if the IP address of the machine running the server is 192.168.20.1 and the client program name is named ``clientapp,'' use the following command:

```
$ xterm -display 192.168.20.1 -e clientapp &
```

[Pierre Dal Farra]

---

11. How to Get Further Assistance

11.1. If this Document Still Hasn't Answered Your Question....

Please read all of this answer before posting. I know it's a bit long, but you may be about to make a fool of yourself in front of 50,000 people and waste hundreds of hours of their time. Don't you think it's worth spending some of your time to read and follow these instructions?

If you think an answer is incomplete or inaccurate, please e−mail Robert Kiesling at `rkiesling@mainmatter.com`.

Read the appropriate Linux Documentation Project books. Refer to: (``*Where Is the Documentation?*'')

If you're a Unix or Linux newbie, read the FAQ for *comp.unix.questions*, *news.announces.newusers*, and those for any of the other *comp.unix.** groups that may be relevant.

Linux has so much in common with commercial unices, that almost everything you read there will apply to Linux. The FAQ's, like all FAQ's, be found on `ftp://rtfm.mit.edu/pub/usenet/` (the `mail-server@rtfm.mit.edu` can send you these files, if you don't have FTP access). There are mirrors of `rtfm`'s FAQ archives on various sites. Check the Introduction to *.answers posting, or look in `news-answers/introduction` in the directory above.

Check the relevant HOWTO for the subject in question, if there is one, or an appropriate old style sub−FAQ document. Check the FTP sites.

Try experimenting that's the best way to get to know Unix and Linux.

Read the documentation. Check the manual pages (type `man man` if you don't know about manual pages. Also try `man -k subject` and `apropos subject`. They often list useful and relevant, but not very obvious, manual pages.

Check the Info documentation (type **F1−i**, i.e. the **F1** function key followed by ``i'' in Emacs). This isn't just for Emacs. For example, the GCC documentation lives here as well.

There will also often be a `README` file with a package that gives installation and/or usage instructions.

Make sure you don't have a corrupted or out−of−date copy of the program in question. If possible, download it again and re−install it you probably made a mistake the first time.

Read *comp.os.linux.announce*. It often contains very important information for all Linux users. General X Window System questions belong in *comp.windows.x.i386unix*, not in *comp.os.linux.x*. But read the group first (including the FAQ), before you post. Only if you have done all of these things and are still stuck, should you post to the appropriate *comp.os.linux.** newsgroup. Make sure you read the next question first. "(What to put in a request for help.)"

11.2. What to Put in a Request for Help.

Please read the following advice carefully about how to write your posting or E−mail. Making a complete posting will greatly increase the chances that an expert or fellow user reading it will have enough information and motivation to reply.

This advice applies both to postings asking for advice and to personal E−mail sent to experts and fellow users.

Make sure you give full details of the problem, including:

- What program, exactly, you are having problems with. Include the version number if known and say where you got it. Many standard commands tell you their version number if you give them a `--version` option.
- Which Linux release you're using (Red Hat, Slackware, Debian, or whatever) and what version of that release.
- The exact and complete text of any error messages printed.
- Exactly what behavior you expected, and exactly what behavior you observed. A transcript of an example session is a good way to show this.
- The contents of any configuration files used by the program in question and any related programs.
- What version of the kernel and shared libraries you have installed. The kernel version can be found by typing ``uname -a,'' and the shared library version by typing ``ls -l /lib/libc*.''
- Details of what hardware you're running on, if it seems appropriate.

You are in little danger of making your posting too long unless you include large chunks of source code or uuencoded files, so err on the side of giving too much information.

Use a clear, detailed Subject line. Don't put things like ``doesn't work,'' ``Linux,'' ``help,'' or ``question'' in it we already know that. Save the space for the name of the program, a fragment of an error message, or summary of the unusual behavior.

Put a summary paragraph at the top of your posting.

At the bottom of your posting, ask for responses by email and say you'll post a summary. Back this up by using ``Followup-To: poster.'' Then, actually post the summary in a few days or a week or so. Don't just concatenate the replies you got summarize. Putting the word ``SUMMARY'' in your summary's Subject line is also a good idea. Consider submitting the summary to *comp.os.linux.announce*.

Make sure your posting doesn't have an inappropriate References: header line. This marks your article as part of the thread of the article referred to, which will often cause it to be junked by readers, along with the rest of a boring thread.

You might like to say in your posting that you've read this FAQ and the appropriate HOWTO's this may make people less likely to skip your posting.

Remember that you should not post E−mail sent to you personally without the sender's permission.

11.3. How To Email Someone about Your Problem.

Try to find the author or developer of whatever program or component is causing you difficulty. If you have a contact point for your Linux distribution, you should use it.

Please put everything in your E−mail message that you would put in a posting asking for help.

Finally, remember that, despite the fact that most of the Linux community are very helpful and responsive to E−mailed questions, you're likely asking for help from unpaid volunteers, so you have no right to expect an answer.

---

12. Acknowledgments and Administrivia

12.1. Where To Send Comments.

Please send comments to the maintainer: rkiesling@mainmatter.com.

Contributions to the FAQ may be in any format. Comments and corrections are gratefully received. Again, that email address is: rkiesling@mainmatter.com.

If you wish to refer to a question in the FAQ, it's better to do so by the question heading instead of number. The question numbers are generated automagically, and I don't see them in the source file.

I prefer comments in English to patch files context diff is not my first language.

12.2. Formats in Which This FAQ Is Available.

This document is available as an ASCII text file, an HTML World Wide Web page, Postscript, PDF, and as a USENET news posting.

Section and item numbers are generated with Perl. HTML is generated from SGML source using the Jade DSSSL interpreter by James Clark. Text versions are generated using lynx and edited with **sed**, which are part of most Linux distributions.

The Usenet version is posted regularly to *news.answers*, *comp.answers*, and *comp.os.linux.misc*. It is archived at ftp://rtfm.mit.edu/pub/usenet-by-hierarchy/comp/os/linux/misc/.

For Postscript and PDF versions, please contact the FAQ maintainer.

If you would like to receive the archived version of the FAQ by E-mail, send the following in the body of an E-mail message to mail-server@rtfm.mit.edu:

```
send faqs/linux/faq
```

Text, HTML, and SGML versions are available from the Linux archives at metalab.unc.edu, and from http://www.linuxdoc.org/, but they may be out of date, owing to lack of time on the LDP maintainers' parts.

The latest text and HTML versions are available at http://www.mainmatter.com and directly from the FAQ maintainer, rkiesling@mainmatter.com.

12.3. Authorship and Acknowledgments.

This FAQ is compiled and maintained by Robert Kiesling, rkiesling@mainmatter.com, with assistance and comments from Linux activists all over the world.

Freddy Contreras, the_blur_oc@hotmail.com, designed and GPL'd the Linux Frequently Asked Questions logos that appear on www.mainmatter.com.

Special thanks are due to Matt Welsh, who moderated *comp.os.linux.announce* and *comp.os.linux.answers*, coordinated the HOWTO's and wrote substantial portions of many of them, Greg Hankins the former Linux Documentation Project HOWTO maintainer, Lars Wirzenius and Mikko Rauhala, the former and current moderators of *comp.os.linux.announce*, Marc−Michel Corsini, who wrote the original Linux FAQ, and Ian Jackson, the previous FAQ maintainer. Thanks also to Roman Maurer for his many updates and additions, especially with European Web sites, translations, and general miscellany.

Last but not least, thanks to Linus Torvalds and the other contributors to Linux for giving us something to talk about!

12.4. Disclaimer and Copyright.

The GNU Free Documentation License

Copyright (c) 2001 Robert Kiesling. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections, with no Front−Cover Texts, and with no Back−Cover Texts. A copy of the license is included at:
http://www.gnu.org/licenses/licenses.html#FDL.

I would be happy to answer any questions regarding the copyright. My email address is:
rkiesling@mainmatter.com.