Installing Debian Linux 1.1 from Floppy Disk

About Copyrights and Software Licenses

I'm sure you've read the licenses that come with most commercial software - they say you can only use one copy of the software on one computer. The Debian Linux System isn't like that. We encourage you to put a copy on every computer in your school or place of business. Lend it to your friends, and help them install it on their computers. You can even make thousands of copies and *sell* them - with a few restrictions. That's because Debian is based on *free software*.

Free software doesn't mean that it doesn't have a copyright, it simply means that the copyrights and licenses of individual programs do not require you to pay for the privilege of copying the programs. There are other sorts of restrictions on how you copy the software, which you can read about once you've installed the system. For example, many of the programs in the system are licensed under the *GNU* **General Public License**, or **GPL**. The GPL requires that you make the *source code* of the programs available whenever you distribute a copy of the program. Thus, we've included the source code for all of those programs in the Debian system. There are several other forms of copyright and software license used on the programs in Debian. You can find the copyrights and licenses of every program by looking in the directory **/usr/doc/copyright** once you've installed your system.

The most important legal notice is that this software comes with **no warranties**. People who write free software can't afford to be sued.

System Requirements

CPU

Your computer must have a 386, 486, Pentium, or Pentium Pro processor, or one of the clones of those processors made by manufacturers such as Cyrix, AMD, TI, IBM, etc. If your processor has letters like "sx", "sl", "slc", etc. after the number as in "386sx", that's fine. The system will *not* run on the 286 or lower processors.

I/O Bus

Your computer must use the ISA, EISA, PCI, or VL bus. The VL bus is also known as VESA Local Bus or VLB. Computers that have PCI or VLB generally have ISA or EISA slots as well. The Micro-Channel bus used in IBM PS/2 computers won't work.

RAM and Disk

You must have at least 4MB of RAM and 40MB of hard disk. If you want to install *everything* from the chess game through the printed-circuit design software, you'll need 300MB or more. The disk interfaces that emulate the "AT" hard disk interface which are often called MFM, RLL, IDE, or ATA are supported. SCSI disk controllers from many different manufacturers are supported. See the **Hardware Compatibility List** for more details.

Floppy Disk

You must have a 1.2MB or 1.44MB floppy disk drive as the "a:" drive in the system upon which you will install Linux.

Display

You should be using a VGA-compatible display interface for the console terminal. Nearly every modern display card is compatible with VGA. CGA, MDA, or HGA might work OK for text, but they won't work with the X Window System, and we haven't tested them. Use of a serial terminal for the console is not yet supported.

Other Hardware

Linux supports a large variety hardware devices such as mice, printers, scanners, modems, network cards, etc. However, none of these devices are required while installing the system.

Before You Start

Backups

Before you start, make sure to back up every file that is now on your system. The installation procedure can wipe out all of the data on a hard disk!

Information You'll Need

If your computer is connected to a network, you should ask your network's system administrator for this information:

- Your host name (you may be able to decide this on your own).
- Your domain name.
- Your computer's IP address.
- The netmask to use with your network.
- The IP address of your network.
- The broadcast address to use on your network.
- The IP address of the default gateway system you should route to, if your network *has* a gateway.
- The system on your network that you should use as a DNS (Domain Name Service) server.
- Whether you connect to the network using Ethernet.

Configuring Your System

Disks

There are some hardware details you should look into. First, decide which disk you want to place the Linux system on. You've backed that up along with your other disks already, right?

The BIOS Set-Up Menu

Your motherboard probably provides a BIOS set-up menu. Some systems start this menu if you press **DEL** while the system is booting, some require a SETUP disk, and some have other means of invoking the BIOS set-up menu. If you can start this menu, use it to control the features discussed in the following several paragraphs.

Boot Device Selection

Many BIOS set-up menus allow you to select the devices that will be used to bootstrap the system. Set this to look for a bootable operating system on **a:**, (the first floppy disk), and then **c:** (the first hard disk). Since you'll boot Linux from a floppy while installing it, it's important that the BIOS enables booting from a floppy disk.

Extended vs. Expanded Memory

If your system provides both ex*ten*ded and ex*pan*ded memory, set it so that there is as much extended and as little expanded memory as possible. Linux requires extended memory and can not use expanded memory.

Virus Protection

Disable any virus-warning features your BIOS may provide. If you have a virus-protection board or other special hardware, make sure it is disabled or physically removed while running Linux. These aren't compatible with Linux, and Linux has a better method of protecting you from viruses.

Shadow Ram

Your motherboard probably provides *shadow RAM*. You may see settings for "Video BIOS Shadow", "C800-CBFF Shadow", etc. **Disable** all shadow RAM. Shadow RAM is used to accellerate access to the ROMs on your motherboard and on some of the controller cards. Linux avoids using these ROMs once it has booted because it provides its own faster 32-bit software in place of the 16-bit programs in the ROMs. Disabling the shadow RAM may make some of it available for programs to use as normal memory. Leaving the shadow RAM enabled may interfere with Linux access to hardware devices.

Advanced Power Management

If your motherboard provides Advanced Power Management (APM), configure it so that power management is controlled by APM. Disable the doze, standby, suspend, nap, and sleep modes, and disable the hard-disk power-down timer. Linux can take over control of these modes, and can do a better job of power-management than the BIOS. The version of the operating system kernel on the installation floppies does not, however, use APM, because we've had reports of one laptop system crashing when the Linux APM driver is configured. Once you've installed Linux, you can install the *source* package and build a custom-configured version of the operating system kernel to enable APM and other features.

Other BIOS Settings to Watch Out For

If your BIOS offers something like "15-16 MB Memory Hole", please disable that. Linux expects to find memory there if you have that much RAM.

We have a report of an Intel Endeavor motherboard on which there is an option called "LFB" or "Linear Frame Buffer". This had two settings: "Disable" and "1 Megabyte". Set it to "1 Megabyte". When disabled, the installation floppy was not read correctly, and the system eventually crashed. At this writing we don’t understand what’s going on with this particular device - it just worked with that setting and not without it.

Hardware Settings to Watch Out For

If any cards provide "mapped memory", the memory should be mapped somewhere between 0xA0000 and 0xFFFFF (from 640K to just below 1 megabyte) or at an address at least 1 megabyte greater than the total amount of RAM in your system.

Windows-specific Hardware

A disturbing trend is the proliferation of *Windows* modems and printers. In some cases these are specially designed to be operated by the Microsoft Windows operating system and bear the legend *Made expecially for Windows-based computers*. This is generally done by removing the embedded processors of the hardware and shifting the work they do over to a Windows driver that is run by your computer’s main CPU. This strategy makes the hardware less expensive, but the savings are often *not* passed on to the user and this hardware may even be more expensive than equivalent devices that retain their embedded intellegence.

You should avoid windows-specific hardware for two reasons. The first is that the manufacturers do not generally make the resources available to write a Linux driver. Generally, the hardware and software interface to the device is proprietary, and documentation is not available without a non-disclosure agreement, if it is available at all. This precludes its being used for free software, since free software writers disclose the source code of their programs. The second reason is that when devices like these have had their embedded processors removed, the operating system must perform the work of the embedded processors, often at *real-time* priority, and thus the CPU is not available to run your programs while it is driving these devices. Since the typical Windows user does not multi-process as intensively as a Linux user, the manufacturers hope that the Windows user simply won’t notice the burden this hardware places on their CPU. However, any multi-processing operating system, even *Windows 95* or *NT,* is degraded when peripheral manufacturers skimp on the embedded processing power of their hardware.

You can help this situation by encouraging these manufacturers to release the documentation and other resources necessary for us to program their hardware, but the best strategy is simply to avoid this sort of hardware.

Other Closed Hardware

Some hardware manufacturers simply won’t tell us how to write drivers for their hardware, or they won’t allow us access to the documentation without a non-disclosure agreement that would prevent us from releasing the Linux source code. One example is the IBM laptop DSP sound system used in recent ThinkPad systems - some of these systems also couple the sound system to the modem. Since we haven't been granted access to the documentation on these devices, they simply won't work under Linux. You can help by asking the manufacturers of such hardware to release the documentation. If enough people ask, they will realize that Linux is an important market.

Writing the Floppy Disk Image Files to Floppy Disk

You will need the file root.bin . If your **a:** drive of the system upon which you will install Linux uses 1.2MB floppy disks, you will also need the files boot1200.bin, base12-1.bin, base12-2.bin, base12-3.bin, and base12-4.bin. If the **a:** drive of the system upon which you will install Linux uses 1.44MB floppy disks, you'll need the files boot1440.bin, base14-1.bin, base14-2.bin, and base14-3.bin. All of these are *floppy disk image files*, which means that each file contains the complete contents of a floppy disk in *raw* form. A special program is used to write the image files to floppy disk in *raw* mode.

Find 6 formatted floppy disks if you are using 1.4 megabyte floppies, or 7 if you are using 1.2 megabyte floppies. Mark these as "Installation Boot", "Installation Root", "Custom Boot", "Base 1", "Base 2", "Base 3", and if you are using 1.2 megabyte floppies: "Base 4".

Here is the filename-to-disk-label correspondence:

boot1440.bin or boot1200.bin: "Installation Boot"
root.bin: "Installation Root"
base14-1.bin or base12-1.bin: "Base 1"
and so on for "Base 2", "Base 3", and, if you are using 1.2 megabyte floppies: "Base 4".

No file is written to the Custom Boot floppy, that will be written by the Debian system while it is being installed.

Writing from a DOS, Windows, or OS-2 System

You'll find the rawrite2.exe program in the **/debian/tools** directory. There's also a rawrite2.txt file containing instructions for rawrite2.exe .

To write the floppy disk image files to the floppy disks, use the command

```
rawrite2 file drive
```

where *file* is one of the floppy disk image files, and *drive* is either **a:** or **b:**.

Writing from a Linux or Unix System

Some workstations attempt to automaticaly mount a floppy disk when you place it in the drive. You might have to disable this feature before the workstation will allow you to write a floppy in *raw mode*. Unfortunately, I don't know the command necessary to do this for your particular workstation. Ask your system administrator.

To write the floppy disk image files to the floppy disks, use the command

```
dd if=file of=/dev/fd0 bs=10k ; sync
```

where *file* is one of the floppy disk image files. */dev/fd0* is a commonly used name of the floppy-disk device, it may be different on your workstation. The command may return to the prompt before Unix has finished writing the floppy disk, so look for the disk-in-use light on the floppy drive and be sure that the light is out and the disk has stopped revolving before you remove it from the drive. On some systems, you'll have to run a command to eject the floppy from the drive.

Installing the System

The Installation Boot floppy

Place the Installation Boot floppy in the **a:** floppy drive, and reset the system by pressing **reset**, turning the system off and then on, or by pressing **Control-Alt-Del** on the keyboard. The floppy disk should be accessed, and you should then see a screen that starts with *Welcome to Debian Linux* and ends with the **boot:** prompt.

You can do two things at the **boot:** prompt. You can press the function keys F1 through F10 to view a few pages of helpful information, or you can boot the system. If you have any hardware devices that aren't made accessable from Linux correctly when Linux boots, you may find a parameter to add to the boot command line in the screens you see by pressing F3, F4, and F5. If you add any parameters to the boot command line, be sure to type the word **linux** and a space before the first parameter. If you simply press Enter, that's the same as typing **linux** without any special parameters.

If this is the first time you're booting the system, just press Enter and see if it works correctly. It probably will. If not, you can reboot later and look for any special parameters that inform the system about your hardware.

Once you press Enter, you should see the message *Loading...*, and then *Uncompressing Linux...*, and then a page or so of cryptic information about the hardware in your system. There may be a few messages in the form *can't find something*, or *something not present*, or even *this driver release depends on something*. Most of these messages are harmless. You see them because the installation boot disk is built to run on computers with many different peripheral devices. Obviously, no one computer will have every possible peripheral device, so the operating system may emit a few complaints while it looks for peripherals you don't own.

The last message on the screen should be *VFS: Insert root floppy disk to be loaded into ramdisk and press Enter*. Insert the Installation Root Disk, and then press enter.

The Installation Root Disk

The system should emit a message like *RAMDISK: Compressed image found at block 0*, or *Loading 1440 blocks into ram disk...*. You'll see a few more messages and then your a dialog box should appear on your screen.

If you see a message like *end_request: I/O error, dev 02:00, sector 0* followed by another *insert root floppy* prompt, you've pressed Enter before you inserted the disk. Do not proceed, reset the system and start again. If you proceed after that message, the system will not use a RAM disk and the installation will not run correctly.

The Color-or-Monochrome Dialog Box

If your monitor displays black-and-white, press Enter to continue with the installation. Otherwise, use the arrow key to move the cursor to the *Color* menu item and then press Enter. The display should change from black-and-white to color. Then press Enter again to continue with the installation.

The Main Menu

You may see a dialog box that says *The installation program is determining the current state of your system*. On some systems, this will go by too quickly to read. You'll see this dialog box between steps in the main menu. The installation program will check the state of the system in between each step. This checking allows you to re-start the installation without losing the work you have already done if you happen to halt your system in the middle of the installation process.

The Shell

If you are an experienced Unix or Linux user, press **LeftAlt-F2** to get to the second *virtual console*. That's the **Alt** key on the left-hand side of the space bar, and the **F2** function key, at the same time. This is a separate window running a Bourne shell clone called **ash**. At this point you are booted from the floppy disk, and there is a limited set of Unix utilities available for your use. You can see what programs are available with the command **ls /bin /sbin /usr/bin /usr/sbin**. Use the menus to perform any task that they are able to do - the shell and commands are only there in case something goes wrong. In particular, you should always use the menus, not the shell, to activate your swap partition, because the menu software can't detect that you've done this from the shell. Press **LeftAlt-F1** to get back to menus.

Last Chance

Did we tell you to back up your disks? Here's your first chance to wipe out all of the data on your disks, and your last chance to save your old system. If you haven't backed up all of your disks, remove the floppy from the drive, reset the system, and run backups.

Partition Your Hard Disks

If you have not already partitioned your disks for Linux native and Linux swap filesystems, the menu item *Next* will be *Partition a Hard Disk*. If you have already created at least one Linux Native and one Linux Swap disk partition, the *Next* menu selection will be *Initialize and Activate the Swap Disk Partition*. Whatever the *Next* menu selection is, you can use the down-arrow key to select *Partition a Hard Disk*.

The *Partition a Hard Disk* menu item presents you with a list of disk drives you can partition, and runs the *cfdisk* program, which allows you to create and edit disk partitions. The cfdisk manual page is included with this document, and you should read it now. You must create one "Linux" (type 83) disk partition, and one "Linux Swap" (type 82) partition.

Your swap partition will be used to provide virtual memory for the system and should be between 16 and 128 megabytes in size, depending on how much disk space you have and how many large programs you want to run. Linux will not use more than 128 megabytes of swap, so there's no reason to make your swap partition larger than that. a swap partition is strongly recommended, but you can do without one if you insist, and if your system has more than 16 megabytes of RAM. If you wish to do this, please select the *Do Without a Swap Partition* item from the menu.

The "Linux" disk partition will hold all of your files, and you may make it any size between 40 megabytes and the maximum size of your disk minus the size of the swap partition. If you are already familiar with Unix or Linux, you may want to make additional partitions - for example, you can make partitions that will hold the **/var**, and **/usr**, filesystems.

Initialize and Activate the Swap Disk Partition

This will be the *Next* menu item once you have created one disk partition. You have the choice of initializing and activating a new swap partition, activating a previously-initialized one, and doing without a swap partition. It's always permissible to re-initialize a swap partition, so select *Initialize and Activate the Swap Disk Partition* unless you are sure you know what you are doing. This menu choice will give you the option to scan the entire partition for un-readable disk blocks caused by defects on the surface of the hard disk platters. This is useful if you have MFM, RLL, or older SCSI disks, and never hurts. Properly-working IDE disks don't need this choice, as they have their own internal mechanism for mapping out bad disk blocks.

The swap partition provides *virtual memory* to supplement the RAM memory that you've installed in your system. It's even used for virtual memory while the system is being installed. That's why we initialize it first.

Initialize a Linux Disk Partition

At this point, the *Next* menu item should be *Initialize a Linux Disk Partition*. If it isn't, it's because you haven't completed the disk partitioning process, or you haven't made one of the menu choices dealing with your swap partition.

You can initialize a Linux Disk partition, or alternatedly you can mount a previously-initialized one.

These floppies will not upgrade an old system without removing the files - Debian provides a different procedure than using the boot floppies for upgrading existing Debian systems. Thus, if you are using old disk partitions that are not empty, you should initialize them (which erases all files) here. You must initialize any partitions that you created in the disk partitioning step. About the only reason to mount a partition without initializing it at this point would be to mount a partition upon which you have already performed some part of the installation process using this same set of installation floppies.

Select the *Next* menu item to initialize and mount the **/** disk partition. The first partition that you mount or initialize will be the one mounted as **/** (pronounced *root*). You will be offered the choice to scan the disk partition for bad blocks, as you were when you initialized the swap partition. It never hurts to scan for bad blocks, but it could take 10 minutes or more to do so if you have a large disk.

Once you've mounted the **/** partition, the *Next* menu item will be *Install the Base System* unless you've already performed some of the installation steps. You can use the arrow keys to select the menu items to initialize and/or mount disk partitions if you have any more partitions to set up. If you have created separate partitions for **/var**, **/usr**, or other filesystems, you should initialize and/or mount them now.

Install the Base System

This should be the *Next* menu step after you've mounted your **/** disk, unless you've already performed some of the installation steps on **/**. Select the *Install the Base System* menu item. There will be a pause while the system looks for a "local copy" of the base system. This search is for CD-ROM installations and will not succeed, and you'll be offered a menu of drives to use to read the base floppies. Select the appropriate drive. Feed in the Base 1, 2, and 3 (and 4 if you are using 1.2MB floppies) as requested by the program. If one of the base floppies is unreadable, you'll have to create a replacement floppy and feed all 3 (or 4) floppies into the system again. Once the floppies have all been read, the system will install the files it's read from them. This could take 10 minutes or more on slow systems, less on faster ones.

Install the Operating System Kernel

At this point, the *Next* menu item should be *Install the Operating System Kernel*. Select it, and you will be prompted to select a floppy drive and insert the Installation Boot floppy. This will copy the kernel on to the hard disk. In a later step this kernel will be used to create a custom boot floppy for your system, and to make the hard disk bootable without a floppy.

Once the kernel has been copied over to your hard disk, you will be asked to install a master boot record. If you aren't using a boot manager (and this is probably the case if you don't know what a boot manager is), answer **yes** to this question. The next question will be whether you want to boot Linux automaticaly from the hard disk when you turn on your system. This sets Linux to be the *bootable partition* - the one that will be loaded from the hard disk. If you answer **no** to this question, you can set the bootable partition later using the DOS **fdisk** program, or with the Linux **fdisk** or **activate** programs.

If you are installing Linux on a drive other than the first hard disk in your system, be sure to make a boot floppy. The boot ROM of most systems is only capable of directly booting from the first hard drive, not the second one. You can, however, work around this problem once you've installed your system. To do so, read the instructions in the directory **/usr/doc/lilo**.

Configure the Base System

At this point you've read in all of the files that make up a minimal Debian system, but you must perform some configuration before the system will run. Select the *Configure the Base System* menu item.

You'll first be asked about the keyboard you are using. You can select the United States keyboard, or the keyboards used in other locales. Next, you'll be asked to select your time zone. Look for your time zone or region of the world in the menu, and type it at the prompt. This may lead to another menu, in which you can select your actual time zone.

Next, you'll be asked if your system clock is to be set to GMT or local time. Select GMT if you will only be running Linux and Unix on your system, and select local time if you will be running another operating system such as DOS or Windows. Unix and Linux keep GMT time on the system clock and use software to convert it to the local time zone. This allows them to keep track of daylight savings time and leap years, and even allows users who are logged in from other time zones to individually set the time zone used on their terminal. If you run the system clock on GMT and your locality uses *daylight savings time*, you'll find that the system adjusts for daylight savings time properly on the days that it starts and ends.

Configure the Network

You'll have to configure the network even if you don't have a network, but you'll only have to answer the first two questions - *what is the name of your computer?*, and *is your system connected to a network?*.

If you are connected to a network, here come some questions that you may not be able to figure out on your own - check with your system administrator if you don't know:

- Your host name.
- Your domain name.
- Your computer's IP address.
- The netmask to use with your network.
- The IP address of your network.
- The broadcast address to use on your network.
- The IP address of the default gateway system you should route to, if your network *has* a gateway.
- The system on your network that you should use as a DNS (Domain Name Service) server.
- Whether you connect to the network using Ethernet.

Some technical details you might, or might not, find handy: the program will guess that the network IP address is the bitwise-AND of your system's IP address and your netmask. It will guess the broadcast address is the bitwise OR of your system's IP address with the bitwise negation of the netmask. It will guess that your gateway system is also your DNS server. If you can't find any of these answers, use the system's guesses - you can change them once the system has been installed, if necessary, by editing /etc/init.d/network .

Make a Boot Floppy

You should make a boot floppy even if you intend to boot the system from the hard disk. The reason for this is that it's possible for the hard disk bootstrap to be mis-installed, but a boot floppy will almost always work. Select *Make a Boot Floppy* from the menu and feed the system a blank floppy as directed. Make sure the floppy isn't write-protected, as the software will format and write it. Mark this the "Custom Boot" floppy and write-protect it once it has been written.

The Moment of Truth

This is what electrical engineers call the *smoke test* - what happens when you turn on a new system for the first time. Remove the floppy disk from the floppy drive, and select the *Reboot the System* menu item. If the Linux system doesn't start up, insert the Custom Boot floppy you created and reset your system. Linux should boot. You should see the same messages as when you first booted the installation boot floppy, followed by some new messages, and then a screen that prompts for your system's *root password*.

Set the Root Password

This is the password for the *super-user*, a login that bypasses all security protection on your system. It should only be used to perform system administration, and only for as short a time as possible. Do not use **root** as your personal login. You will be prompted to create a personal login as well, and that's the one you should use to send and receive e-mail and perform most of your work - not root. The reason to avoid using root's privileges is that you might be tricked into running a *trojan-horse* program - that is a program that takes advantage of your super-user power to compromise the security of your system behind your back. Any good book on Unix system administration will cover this topic in more detail - consider reading one if it's new to you. The good news is that Linux is probably more secure than other operating systems you might run on your PC. DOS and Windows, for example, give *all* programs super-user privilege. That's one reason that they have been so plagued by viruses.

If asked to enter the "old password" for root, just press return. All of the passwords you create should contain from 6 to 8 characters, and should contain both upper and lower-case characters, as well as punctuation characters.

Once you've added both logins, you'll be dropped into the **dselect** program. This program allows you to select *packages* to be installed on your system. If you have a CD-ROM or hard disk containing the additional Debian packages that you want to install on your system, this will be useful to you. Otherwise, you may want to quit dselect and start it later, once you have transported the Debian package files to your system. You must be the super-user (root) when you run dselect.

Log In

After you've quit dselect, you'll be presented with the login prompt. Log in using the personal login and password you selected. Your system is now ready to use.

Last Update

The last update of this document was made on Wed May 29 14:39:41 PDT 1996

Copyright of This Document

Trademark Acknowledgement

Trademarks that are not explicitly acknowledged here are the property of their respective holders. 386, 386sx, 486, Pentium, and Pentium Pro are the property of Intel. Windows is a trademark of Microsoft. ThinkPad and PS/2 are the property of IBM.