

Official Slackware HOWTO

Slackware Linux CD-ROM Installation HOWTO Patrick Volkerding <volkerdi AT slackware.com> v13.1, 2010-05-18

This document covers installation of the Slackware(R) distribution of the Linux operating system from the Slackware CD-ROM.

1. Introduction

Linux is a multiuser, multitasking operating system that was developed by Linus Torvalds and hundreds of volunteers around the world working over the Internet.

The Linux operating system now runs on several machine architectures, including ARMs, Intel 80x86, Sparc, 68K, PowerPC, DEC Alpha, MIPS, and others. The x86 Slackware distribution of Linux runs on most PC processors compatible with the Intel 486 or better, including (but not limited to) the Intel 486, Celeron, Pentium I, MMX, Pro, II, III, Xeon, 4, M, D, Core, Core 2, Core i7, and Atom; AMD 486, K5, K6, K6-II, K6-III, Duron, Athlon, Athlon XP, Athlon MP, Athlon 64, Sempron, Phenom, Phenom II, and Neo; Cyrix 486, 5x86, 6x86, M-II; Via Cyrix III, Via C3, Via Nano; Transmeta Crusoe and Efficeon. Essentially anything that's x86 and 32-bit (with at least i486 opcodes) will do for the 32-bit x86 edition of Slackware, or 64-bit and supporting x86_64 extensions (also known as AMD64, EM64T, or Intel 64) for the x86_64 edition of Slackware.

Linux is modeled after the UNIX(R) operating system. The Slackware distribution contains a full program development system with support for C, C++, Fortran-77, LISP, and other languages, full TCP/IP networking with NFS, PPP, CIFS/SMB (Samba), a full implementation of the X Window System, and much more.

1.1. Sources of Documentation

If you're new to Slackware, you'll be happy to know there is a *lot* of documentation and help available both on the Internet and on the CD-ROM itself.

A great source of general documentation about Linux is the Linux Documentation Project, online at: <http://tldp.org/>

Here you will find a collection of documents known as the "Linux HOWTOs" as well as other useful guides.

For additional help with Slackware, check out the Slackware forum at linuxquestions.org.

2. Hardware Requirements

Most PC hardware will work fine with Slackware, but some Plug-and-Play devices can be tricky to set up. In some cases you can work around this by letting DOS initialize the card and then starting

Slackware with the Loadlin utility. Setting the computer's BIOS to configure Plug-and-Play cards also may help – to do this, change the “Plug and Play OS” option to “no”.

Here's a basic list of what you'll need to install Slackware:

128 megabytes (128MB) or more of RAM. If you have less RAM than this, you might still be able to install, but if so don't expect the best possible experience.

You also will need some disk space to install Slackware. For a complete installation, you'll probably want to devote a 10GB *or larger* partition completely to Slackware (you'll need almost 6GB for a full default installation, and then you'll want extra space when you're done). If you haven't installed Slackware before, you may have to experiment. If you've got the drive space, more is going to be better than not enough. Also, you can always install only the first software set (the A series containing only the basic system utilities) and then install more software later once your system is running.

If you use SCSI, Slackware supports most SCSI controllers. The “huge” kernels support as much of the boot hardware as possible, including several hardware RAID controllers, Fiber Channel controllers, software RAID in linear and RAID 0 through 6 and RAID 10, LVM (Logical Volume Manager), and kernel support required to have fully encrypted systems.

To install from the DVD or CD-ROM, you'll need a supported drive. These days, the chances that your drive is supported by the install kernels is excellent. But, if not, you can always use a USB stick and install via the network. Or, use a floppy disk to install using PXE and the network. See the docs in `usb-and-pxe-installers` and the `etherboot` directory within for instructions.

3. Slackware Space Requirements

Slackware divides the installable software into categories. (in the old days when people installed Linux from floppy disks, these were often referred to as “disk sets”) Only the A series category (containing the base Linux OS) is mandatory, but you can't do very much on a system that only has the A series installed. Here's an overview of the software categories available for installation, along with the (approximate) amount of drive space needed to install the entire set:

- A The base Slackware system. (310 MB)
- AP Linux applications. (290 MB)
- D Program development tools. (600 MB)
- E GNU Emacs. (100 MB)
- F FAQs and HOWTOs for common tasks. (35 MB)
- K Linux 2.6.33.4 kernel source. (445 MB)
- KDE The KDE desktop environment and applications. (925 MB)
- KDEI Language support for KDE. (800 MB)
- L System libraries. (950 MB)
- N Networking applications and utilities. (325 MB)
- T TeX typesetting language. (285 MB)
- TCL Tcl/Tk/TclX scripting languages and tools. (15 MB)
- X X Window System graphical user interface. (300 MB)
- XAP Applications for the X Window System. (490 MB)
- Y Classic text-based BSD games. (6 MB)

If you have the disk space, we encourage you to do a full installation for best results. Otherwise,

remember that you must install the A set. You probably also want to install the AP, D, L, and N series, as well as the KDE, X, and XAP sets if you wish to run the X Window System. The Y series is fun, but not required.

3.1 Preparing a Partition for Slackware

If you plan to install Slackware onto its own hard drive partition (this offers optimal performance), then you'll need to prepare one or more partitions for it.

A partition is a section of a hard drive that has been set aside for use by an operating system. You can have up to four primary partitions on a single hard drive. If you need more than that, you can make what is called an extended partition. This is actually a way to make one of the primary partitions contain several sub-partitions.

Usually there won't be any free space on your hard drive. Instead, you will have already partitioned it for the use of other operating systems, such as MS-DOS or Windows. Before you can make your Linux partitions, you'll need to remove one or more of your existing drive partitions to make room for it. Removing a partition destroys the data on it, so you'll want to back it up first.

If you've got a large partition that you'd like to shrink to make space for Slackware you might consider using GParted, a partition editor that allows resizing and moving of existing partitions. They have a Live CD and USB image that allows running the program on a minimal OS, as well as versions to boot from PXE or the hard drive. Bootable images with GParted may be found here:

<http://gparted.sourceforge.net/index.php>

There's also the regular version of GNU parted that does the same thing from the command line. It is included in the installer, and as a package in the L series.

If you plan to repartition your system manually, you'll need to back up the data on any partitions you plan to change. The usual tool for deleting/creating partitions is the fdisk program. Most PC operating systems have a version of this tool, and if you're running DOS or Windows it's probably best to use the repartitioning tool from that OS. Usually DOS uses the entire drive. Use DOS fdisk to delete the partition. Then create a smaller primary DOS partition, leaving enough space to install Linux. Preferably this should be more than 6GB. If your machine doesn't have a lot of RAM, you'll want another partition for swap space. The swap partition should be equal to the amount of RAM your machine has, but should in any case be at least 128MB. If you don't have that much drive space to spare, the more the better to avoid running out of virtual RAM (especially if you plan on using a graphical desktop). You'll then need to reinstall DOS or Windows on your new DOS partition, and then restore your backup.

We'll go into more detail about partitioning later, and you don't need to create any new partitions yet - just make sure you have enough free space on the drive to do an installation (more than 6GB is ideal), or that you have some idea about which existing partition you can use for to install on.

3.2 Booting the Slackware CD-ROM

If your machine has a bootable CD-ROM drive (you may need to configure this in the system's BIOS settings) then you'll be able to directly boot the first CD-ROM. If not, then see the files in the usb-and-

pxe-installers directory for information about alternative methods of booting the installer. Also, don't neglect to read the CHANGES_AND_HINTS.TXT file, which is probably the most accurate piece of documentation to ship with Slackware (thanks Robby!).

Now it's time to boot the disc. Put the Slackware installation CD-ROM in your machine's CD-ROM drive and reboot to load the disc. You'll get an initial information screen and a prompt (called the "boot:" prompt) at the bottom of the screen. This is where you'll enter the name of the kernel that you want to boot with. With most systems you'll want to use the default kernel, called `hugesmp.s`. Even on a machine with only a single one-core processor, it is recommended to use this kernel if your machine can run it. Otherwise use the `huge.s` kernel, which should support any 486 or better.

To boot the `hugesmp.s` kernel, just enter `hugesmp.s` on the boot prompt:

```
boot: hugesmp.s
```

(actually, since the `hugesmp.s` kernel is the default, you could have just hit ENTER and the machine would go ahead and load the `hugesmp.s` kernel for you)

If you've got some non-standard hardware in your machine (or if `hugesmp.s` doesn't work, and you're beginning to suspect you need a different kernel), then you'll have to try `huge.s`. If, for some reason, that still will not boot and you know that your hardware should be supported by the 2.6.33.4 kernel, contact [volkerdi at slackware dot com](mailto:volkerdi@slackware.com) and I will see what I can do.

These are the kernels shipped in Slackware:

`hugesmp.s` This is the default installation kernel. If possible,

```
you can save a bit of RAM later (and some ugly warnings at
boot time or when trying to load modules when the driver is
already built-in) by switching to a generic kernel. In this
case that would be gensmp.s, which is a similar kernel but
without filesystems and many of the less common drive
controllers built in. To support these (at the very least
your root filesystem), an initrd (actually an initramfs)
is required when a generic kernel is used. Previous
versions of Slackware used an ext2 filesystem for this, but
now a filesystem-less dynamic kernel-based directory
structure is used. A big advantage of this is that the size
usable by the initrd is only limited by the amount of RAM in
the machine. A disadvantage is that the generic kernels no
longer include any filesystems besides romfs, so old
initrd.gz files are not usable (they would have needed new
modules anyway), and it is trickier to get a custom binaries
or modules or whatever into the installer for guru-install
purposes. It's not impossible though -- think tar to/from a
device such as a USB stick, or leveraging ROMFS.
```

`gensmp.s` The trimmed down, more modular version of `hugesmp.s`. This

```
can be switched to, after setting up an initrd and
reinstalling LILO. It is packaged as a .txz, and can be
found on the installed system as:
```

```
/boot/vmlinuz-generic-smp-2.6.33.4-smp
```

huge.s This is the 486-compatible single processor version of the

```
hugesmp.s kernel. Try this if hugesmp.s does not work on
your machine.
```

generic.s The trimmed down, more modular version of huge.s. Found on

```
the system as:
/boot/vmlinuz-generic-2.6.33.4
This also requires using an initrd.
```

speakup.s This is like the huge.s (486 compatible loaded kernel), but

```
has support for Speakup and all the SCSI, RAID, LVM, and other
features of huge.s. There is no corresponding generic kernel
for speakup.s, but the vanilla linux sources may be patched
with the speakup sources in source/k (this will probably work
on any recent kernel). After that, whatever customizations are
needed should be easily adjusted. The speakup.s kernel is
used to support hardware speech synthesizers as well as
software one like festival (though these require additional
programs that are not yet shipped with Slackware).
```

```
For more information about speakup and its drivers check out:
http://www.linux-speakup.org.
To use this, you'll need to specify one of the supported
synthesizers on the kernel's boot prompt:
```

```
speakup.s speakup.synth=synth
```

```
where 'synth' is one of the supported speech synthesizers:
acntpc, acntsa, apollo, audptr, bns, decext, decpc,
dectlk, dtlk, dummy, keypc, ltlk, soft, spkout, txprt.
```

```
A serial port may be specified with an option like this:
```

```
speakup.s speakup.synth=decext speakup.ser=1
```

```
Note that speakup serial ports are numbered starting with
one (1, 2, 3) rather than the more typical 0, 1, 2 numbering
usually seen on Linux.
```

Note that if you use the huge (non-SMP kernel) and plan to compile any third party kernel modules, you may need to apply the kernel patch in `/extra/linux-2.6...` or, you could just `cd` to the kernel sources, run “make menuconfig”, make sure that SMP (and the `-smp` suffix) are turned off, and recompile the kernel with “make”. But, that's for later – after the install.

Once you've entered your kernel choice and hit ENTER, the kernel and install program will load from

the DVD or CD-ROM, and you'll arrive at the Linux login prompt. (You're running Linux now. Congratulations! 😊)

To log into the system, enter the name of the superuser account and hit Enter:

```
root
```

Since there is no password on the install CD, you will be logged in right away.

3.3 Using Linux fdisk to create Linux partitions

At this point, you should have a large chunk of unpartitioned space on your hard drive that you'll be making into partitions for Slackware.

Now you're ready to create your root Linux partition. To do this, you'll use the Linux version of fdisk.

To need to partition a hard drive, you need to specify the name of the device when you start fdisk. For example:

```
fdisk /dev/sda (Repartition the first hard drive) fdisk /dev/sdb (Repartition the second hard drive)
```

NOTE: If you prefer, you may also try a newer menu-driven version of Linux fdisk called 'cfdisk'. Rumor has it that MOST people do prefer cfdisk, and "newer" has to be taken in context. cfdisk has many years of testing behind it.

Once you've started fdisk, it will display a command prompt. First look at your existing partition table with the 'p' command:

```
Command (m for help): p
```

```
Disk /dev/sda: 40.0 GB, 40020664320 bytes 255 heads, 63 sectors/track, 4865 cylinders Units = cylinders of 16065 * 512 = 8225280 bytes
```

| Device | Boot | Start | End | Blocks | Id | System |
|--------|------|-------|-----|--------|----|--------|
|--------|------|-------|-----|--------|----|--------|

```
/dev/sda1 * 1 2423 19462716 c W95 FAT32 (LBA)
```

Here we can see that there is one DOS partition on the drive already, starting on the first cylinder and extending to cylinder 2423. Since the drive has 4865 cylinders, the range 2424 - 4865 is free to accept a Linux installation.

If the FAT32 partition were using the entire drive, you would have no choice but to delete it entirely (this destroys the partition), or go back and use some kind of partition resizing tool like GNU parted or Partition Magic to create some free space for the installation. If you need to delete a partition, use the 'd' command. You'll be asked which partition number you want to delete - check the partition size to make sure it's the right one.

Next, you'll want to use the 'n' command to create a primary partition. This will be your root Linux partition.

```
Command (m for help): n
```

```
Command action
  e   extended
  p   primary partition (1-4)
```

You'll want to enter 'p' to make a primary partition.

```
Partition number (1-4): 2
```

Here, you enter "2" since DOS is already using the first primary partition. Fdisk will first ask you which cylinder the partition should start on. Fdisk knows where your last partition left off and will suggest the first available cylinder on the drive as the starting point for the new partition. Go ahead and accept this value. Then, fdisk will want to know what size to make the partition. You can specify this in a couple of ways, either by entering the ending cylinder number directly, or by entering a size. In this case, we'll enter the last cylinder. Here's what the screen looks like as these figures are entered:

```
First cylinder (2424-4865): 2424
Last cylinder or +size or +sizeM or +sizeK (2424-4865): 4700
```

You have now created your primary Linux partition with a size of 18.7 GB. Next, you'll want to make a Linux swap partition. You do this the same way. First, enter another "n" to make a primary partition:

```
Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
```

Enter "p" to select a primary partition. Partition 1 is in use by DOS, and you've already used partition 2 for Linux, so you'll want to enter "3" for the new partition number:

```
Partition number (1-4): 3
```

Since this is the last partition we plan to make on this hard drive, we'll use the end cylinder this time. Here are the entries for this:

```
First cylinder (4701-4865): 4701
Last cylinder or +size or +sizeM or +sizeK (4701-4865): 4865
```

Now we need to set the type of partition to 82, used for Linux swap. The reason we didn't need to set a partition type the last time is that unless otherwise specified Linux fdisk automatically sets the type of all new partitions to 83 (Linux). To set the partition type, use the "t" command:

```
Command (m for help): t
Partition number (1-4): 3
Hex code (type L to list codes): 82
```

Now you're ready to save the updated partition table information onto your hard drive. Use the "p" command again to check the results and be sure you're satisfied with them:

```
Command (m for help): p
```

Disk /dev/sda: 40.0 GB, 40020664320 bytes 255 heads, 63 sectors/track, 4865 cylinders Units = cylinders of 16065 * 512 = 8225280 bytes

| Device | Boot | Start | End | Blocks | Id | System |
|-----------|------|---------|-----------|--------|-------|-------------|
| /dev/sda1 | 1 | 2423 | 19462716+ | c | W95 | FAT32 (LBA) |
| /dev/sda2 | 2424 | 4700 | 18720732 | 83 | Linux | /dev/sda3 |
| 4701 | 4865 | 1317332 | 82 | Linux | swap | |

This looks good, so we'll use the "w" command to write the data out to the drive's partition table. If you want to exit without updating the partition table (if you've made a mistake), then you can exit without changing anything by using the "q" command instead.

When you exit fdisk using the "w" command, fdisk recommends that you reboot the machine to be sure that the changes you've made take effect. Unless you've created extended partitions, you can go ahead and run setup without rebooting.

Note: Sometimes fdisk will give you a message like "This drive has more than 1024 cylinders" and warn about possible problems using partitions with DOS. This is because MS-DOS suffers from a limitation that only allows access to the first 1024 cylinders on a hard drive. At one time, LILO used the standard BIOS routines to read sectors, so this was a limitation of LILO, too. Luckily modern versions of LILO use the LBA32 method of accessing sectors, so this limitation no longer applies. If you see the warning from fdisk, you can safely ignore it.

4.0 Installing the Slackware distribution

Now that you have one or more Linux partitions, you are now ready to begin installing software onto your hard drive. To start the Slackware install program, enter the command "setup" and hit enter:

```
# setup
```

The installer will start up with a full-color menu on your screen with the various options needed to install Slackware. In general, you'll want to start with the ADDSWAP option. Even if you've already created and activated a swap partition manually, you'll need to run this so Slackware adds the swap partition to your /etc/fstab file. If you don't add it, your system won't use the swap space when you reboot.

Installing a typical system involves running the following options from the setup menu in this order: ADDSWAP, TARGET, SOURCE, SELECT, INSTALL, and CONFIGURE. You may also start with KEYMAP if you have a non-US keyboard layout, or with TARGET if you don't want to use a swap partition.

For the rest of this section, we'll walk through a typical installation process.

4.1 The ADDSWAP option:

First, we select the ADDSWAP option. The system will scan for partitions marked as type "Linux swap" and will ask if you want to use them for swap space. Answer YES, and the system will format the partition and then make it active for swapping. Once it's finished, setup will display a message showing the line it will add to /etc/fstab to configure the swap partition at boot time. Hit enter to

continue, and setup will go on to the TARGET section of the install.

NOTE: If you created a partition to use for swap space, but setup

doesn't see it when it scans your drives, it's possible that the partition type hasn't been set in the partition table. Use the Linux "fdisk" program to list your partitions like this:

```
# fdisk -l
```

```
Disk /dev/sda: 40.0 GB, 40020664320 bytes
255 heads, 63 sectors/track, 4865 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
```

| Device | Boot | Start | End | Blocks | Id | System |
|-----------|------|-------|------|-----------|----|-----------------|
| /dev/sda1 | | 1 | 2423 | 19462716+ | c | W95 FAT32 (LBA) |
| /dev/sda2 | | 2424 | 4700 | 18720732 | 83 | Linux |
| /dev/sda3 | | 4701 | 4865 | 1317332 | 82 | Linux |

In this case, if /dev/sda3 is meant to be a Linux swap partition, you'll need to start fdisk on drive /dev/sda:

```
# fdisk /dev/sda
```

```
Command (m for help): t
Partition number (1-4): 3
Hex code (type L to list codes): 82
```

```
Command (m for help): w
```

This will change the third partition to type 82 (Linux swap) and write the partition table out to /dev/sda.

When you run setup again, the ADDSWAP option should detect the Linux swap partition.

4.2 The TARGET option:

The next option on the setup menu is TARGET. This lets you select which partition(s) you'd like to install Slackware on, and will format them using a Linux filesystem. Depending on which kernel you chose to boot with, your filesystem choices may include ext2 (the traditional Linux filesystem), ext3 (a journaling version of ext2), and Reiserfs (the first journaling filesystem written for Linux; it stores files in a balanced tree).

When you select the TARGET option, the system will scan for "Linux" partitions on your hard drives. If it doesn't find any, you'll need to make sure that you've created partitions using the fdisk program, and that the partitions are labeled as type 83 (Linux). This is the same process shown above. If you've created one or more partitions for Slackware using Linux's fdisk program then you shouldn't have any problems, since Linux fdisk (and cfdisk) sets all new partitions to type 83 (Linux) by default.

You will see a menu listing all the Linux partitions. Use the arrow keys to select the partition you'd like

to use for your root (or primary) Linux partition and hit enter. The setup program will then ask if you'd like to format the partition, and what type of filesystem to use. If this is a new installation of Slackware, you'll need to do this. Otherwise, if you are installing software onto an existing Linux system, you don't need to format the partition. For example, the partition might be used as your /home and contains home directories that you want to keep. If you choose not to format a partition, you'll see "partition will not be reformatted" on the top of the screen as you confirm your choice, so that there can be no question about it.

There are a few options you need to know about when you format Linux partitions. First, you'll need to decide whether or not you'd like to check the partition for bad blocks when you do the format. This is usually not necessary unless you know the drive in question has problems. Checking takes quite a while longer than a normal format (and most IDE drives do self-checking anyway), so you'll probably want to just go ahead and use the "Format" menu option to format the drive without checking. If you have drive problems later on (and can't just replace the hard drive with a better one), then you might want to go back and try again using the "Check" option to map out the bad sectors on the drive.

You'll notice that the partition you just formatted is now listed as "in use." If you made some other partitions for Slackware, you'll need to go through the same process of formatting them, selecting whether or not to check for bad blocks, and setting a reasonable inode density. With these partitions there will be an additional step - you'll need to select where you'd like to put the partition in your directory tree.

MS-DOS/Windows assigns a letter such as A:, B:, C:, etc, to each device. Unlike DOS, Linux makes your devices visible somewhere under the root directory (/). You might have /dev/sda1 for your root partition (/) and put /dev/sda2 somewhere underneath it, such as under your /home directory. When prompted for a mount location, just enter a directory such as /home, and hit enter. As you format each additional partition and place it in the filesystem tree, you'll be returned to the partition selection menu. When you've prepared all of your Linux partitions, you'll go on to the SOURCE option.

4.3 The SOURCE option:

The next menu option is SOURCE, where you select the source from which to install Slackware.

SOURCE displays a menu offering the choice of installation from CD-ROM, a hard drive partition, NFS, HTTP/FTP, or a directory (mounted manually). You'll want to make sure your Slackware CD-ROM is in your drive, and select the first option:

```
"Install from a Slackware CD-ROM"
```

Next, the system will ask you if you'd like to scan for your CD-ROM drive or pick manually from a list. (unless you're trying to show off to your friends, go ahead and let setup scan for the CD-ROM drive automatically). Setup will then try to access the Slackware CD-ROM. If this is successful, setup will tell you that it found and mounted a CD-ROM on a Linux device such as /dev/sr0. If the CD-ROM was successful found, you may skip ahead to the SELECT section below, otherwise read on for some CD-ROM troubleshooting tips.

If setup is not successful in accessing the CD-ROM drive, you'll need to figure out why before you can go on. The most common reason for this is that you used a kernel that doesn't support the CD-ROM drive. If that's the case, you need to restart the installation CD-ROM and specify a kernel that contains a driver to support your CD-ROM drive (if the drive is connected to a SCSI card, for example, you'll

need to use a kernel with support for that card). You can also try switching to a different console with Alt-F2 and mounting the CD-ROM drive manually and then installing from a pre-mounted directory (if you prefer a hands-on approach).

If you have no idea which device an IDE CD-ROM drive is connected to, you should have the system scan for it. You also can look at the messages generated by the system as it boots - you should see a message that Slackware detected your CD-ROM drive along with information about what type of drive it is. You can look at these messages by using the right shift key together with the PageUp and PageDown keys to scroll the screen up and down.

For the network options, you'll need to have a network card that's supported by one of the installer's modules, and preferably a DHCP server running to make the network setup easy. The network install options are to use as your source an NFS server, an FTP server, or an HTTP server (along with an optional port). If you use DHCP to set up, odds are you'll have working name resolution and won't need to enter an IP address to specify the server (but you can if you wish, of course).

The network installation feature is intended primarily to facilitate installing to many machines on a local network. Please don't use it to bog down the Slackware mirror sites.

Thanks to Eric Hameleers for finally bringing FTP/HTTP installation support to the Slackware installer. 😊

4.4 The SELECT option:

The SELECT option lets you select software to install.

When you start the SELECT option, you'll see a menu where you can choose which categories of software you're interested in installing. The first series (called the A series) contains the base filesystem structure and binaries that are crucial for your system to boot and run properly. You must install the A series. Make sure that at least the selection for series A has an [X] next to it. Most of the other choices will also have an [X] next to them, and while you can use the cursor keys and the space bar to unselect items to save space (see the space requirements above for details), you're better off with a complete installation if you have the space for it.

Once you've selected the general categories of software you wish to install, hit enter and you'll go on to the INSTALL option.

4.5 The INSTALL option:

This option actually installs the selected packages to the hard drive.

The first question the INSTALL option will ask is what type of prompting you'd like to use during the installation process. A menu will show several options, including "full", "newbie", "menu", "expert", "custom", "tagpath", and "help". The help option gives detailed information on each of the choices.

Most people will want to use "full". Others might want "menu", "expert" or "newbie" mode. We'll cover each of these in detail now.

The first option to consider is "full". If you select this mode, then setup assumes you want to install all the packages in each selected series and installs them all without further prompting. This is fast and easy. Of course, depending on which software categories you've chosen, this can use a lot of drive

space. If you use this option, you should be installing to a partition with at least 6GB free (and hopefully more like 20GB or so) to insure that you don't run out of drive space during the installation process. Because Linux allows you to split your installation across multiple partitions, the installer cannot know ahead of time whether the packages you've chosen to install will fit your partitioning scheme. Therefore, it is up to you to make sure that there is enough room.

The “newbie” mode (which was formerly known as “normal” mode) installs all of the required packages in each series. For each of the non-required packages (one by one) you'll get a menu where you can answer YES (install the package), NO (do not install the package), or SKIP (skip ahead to the next series). You'll also see a description of what the package does and how much space it will require to help you decide whether you need it or not. The “newbie” mode is verbose, requires input after each package, and is VERY tedious. It certainly takes a lot longer to install using newbie mode, and (in spite of the name), it is easier to make mistakes in newbie mode than by simply doing a full installation. Still, using it is a good way to get a basic education about what software goes into the system since you actually get a chance to read the package descriptions. With a full installation most of the package descriptions will fly by too quickly to read.

If you can decide which packages you want from less information, the “menu” or “expert” options are a good choice, and go much faster than a “newbie” mode installation. These options display a menu before installing each series and let you toggle items on or off with the spacebar. In this Slackware release, the “menu” and “expert” install modes act the same, and both options are kept only for consistency. The “expert” mode lets you toggle packages individually, allowing the user to make good or bad decisions, like turning off crucial packages or installing a package that's part of a larger set of software without installing the other parts. If you know exactly what you need, the “expert” mode offers the maximum amount of flexibility. If you don't know what you need, using the “full” mode is strongly suggested.

The “custom” and “tagpath” options are only used if you've created “tagfiles” for installation. In the first directory of each disk set is a file called “tagfile” containing a list of all the packages in that series, as well as a flag marking whether the package should be installed automatically, skipped, or the user should be prompted to decide. This is useful for situations where you need to install large numbers of machines (such as in a computer lab), but most users will not need to create tagfiles. If you are interested in using them, look at one of the tagfiles with an editor.

If you're new to Slackware, and you have enough drive space, you'll probably want to select the “full” option as the easiest way to install. Otherwise, the “menu” option is another good choice for most beginners. If you think you need (or would just like to see) the extra information offered by the “newbie” mode, go ahead and use that. Don't say you weren't warned about the extra time it requires, though, especially when installing the fragments that make up modular X. Trust us, you'll be better off selecting “full”.

Once you have selected a prompting mode, the system begins the installation process. If you've chosen “menu” or “expert” mode, you'll see a menu of software to choose from right away - use the arrow keys and spacebar to pick what you need, and then hit enter to install it. If you've chosen the “newbie” mode, the installation will begin immediately, continuing until it finds optional packages. You'll get a selection menu for each of these. If you selected “full”, now it's time to sit back and watch the packages install.

If you've selected too much software, it's possible that your hard drive may run out of space during installation. If this happens, you'll know it because you'll see error messages on the screen as setup tries to install the packages. In such a case, your only choice is to reinstall selecting less software. You can avoid this problem by choosing a reasonable amount of software to begin with, and installing

more software later once your system is running. Installing software on a running Slackware system is as easy as it is during the initial installation – just type the following command to mount the Slackware CD-ROM:

```
mount /dev/cdrom /mnt/cdrom
```

Then go to the directory with the packages you want to install, and use the `install-packages` script:

```
cd /mnt/cdrom/slackware/xap
sh install-packages
```

Other options for installing packages later on include “`installpkg`” and “`pkgtool`”. For more information about these, see the man pages (“`man installpkg`”, “`man pkgtool`”).

Once you have installed the software on your system, you'll go on to the `CONFIGURE` option.

4.6 The `CONFIGURE` option:

The setup's `CONFIGURE` option does the basic configuration your system needs, such as setting up your mouse, setting your timezone, and more.

The `CONFIGURE` option will first ensure that you've installed a usable Linux kernel on your hard drive. The installation program should automatically install the kernel used to do the initial installation. If you installed using the `speakup.s` kernel from CD-ROM, the menu will prompt you to re-insert your installation disc and hit enter, and then setup will copy the kernel from the disc to your hard drive.

NOTE: If you install a kernel on your system that doesn't boot correctly, you can still boot your system with the CD-ROM. To do this, you need to enter some information on the boot prompt. For example, if your root partition is on `/dev/hda1`, you'd enter this to boot your system:

```
huge.s root=/dev/hda1 initrd= ro
```

The “`initrd=`” option tells the kernel not to run the `/init` script on the installer image in RAM, and the “`ro`” option makes the root partition initially load as read-only so Linux can safely check the filesystem.

Once you've installed a kernel, you'll be asked if you want to make a USB bootstick for your new system. This is a very good idea if you happen to have a spare USB flash stick that you don't mind having **COMPLETELY ERASED**. 😊, so if you wish to make one, insert a USB flash memory stick when prompted and use the “`Create`” option to create a USB bootstick for your system.

Next you'll be asked what type of mouse you have. Pick the mouse type from the menu (or hit cancel if you don't have a mouse), and setup will create a `/dev/mouse` link. Most computers use a PS/2 mouse, which is the first choice.

After this, other installation scripts will run depending on which packages you've installed. For instance, if you installed the `network-*` packages you'll be asked if you want to configure your network.

4.7 LILO

LILO is the Linux Loader, a program that allows you to boot Linux (and other operating systems) directly from your hard drive. If you installed the LILO package, you now have an opportunity to set it up.

Installing LILO can be dangerous. If you make a mistake it's possible to make your hard drive unbootable. If you're new to Linux, it might be a good idea to skip LILO installation and use the bootdisk to start your system at first. You can install LILO later using the 'liloconfig' command after you've had a chance to read the information about it in /usr/doc/lilo-*. If you do decide to go ahead and install LILO, be sure you have a way to boot all the operating systems on your machine in case something goes wrong. If you can't boot Windows again, use the DOS command "FDISK /MBR" to remove LILO from your master boot record. (You can use a Windows Startup Disk for this)

The easiest way to set your machine up with LILO is to pick the "simple" choice on the LILO installation menu. This will examine your system and try to set up LILO to be able to boot Windows (DOS) and Linux partitions that it finds. If it locates the OS/2 Boot Manager, it will ask if you'd like to configure the Linux partition so that you can add it to the Boot Manager menu. (NOTE: If you use a disk overlay program for large IDE hard drives such as EZ-DRIVE, please see the warning below before installing LILO)

The "expert" option gives you much more control over the configuration of LILO. If you decide to use the "expert" option, here's how you do it. LILO uses a configuration file called /etc/lilo.conf to hold the information about your bootable partitions - the "expert" LILO installation lets you direct the construction of this file. To create the file, first select BEGIN to enter the basic information about where to install LILO. The first menu will ask if you have extra parameters you'd like passed to the Linux kernel at boot time. If you need any extra parameters enter them here.

Then you'll be asked if you wish to use the framebuffer console. The 1024x768x256 console setting is a nice one to use in most cases, but you may need to experiment to find the nicest setting for your card. Some look terrible at modes larger than 800x600 because of the default refresh rates, but at least ATI cards are known to look great at 1024x768x256. If you want to use the framebuffer console, select a mode here.

Next, decide where you want LILO installed. Usually you'll want to install LILO on the boot drive's MBR (master boot record). If you use a different boot manager (like the one that comes with OS/2) then you'll want to install LILO on your root Linux partition and then add that partition to the boot manager menu using its configuration tool. Under OS/2, this is the fdisk program.

NOTE: If you use the EZ-DRIVE utility (a diskmanager program supplied with some large IDE drives to make them usable with DOS) then do not install LILO to the MBR. If you do, you may disable EZ-DRIVE and render your disk unusable with DOS. Instead, install LILO to the superblock of your root Linux partition, and use fdisk to make the partition bootable. (With MS-DOS fdisk, this is called setting the "active" partition)

The next menu lets you set a delay before the system boots into the default operating system. If you're using LILO to boot more than one operating system (such as DOS and Linux) then you'll need to set a delay so you can pick which OS you'd like to boot. If you press the SHIFT key during the delay, LILO will display a prompt where you can type a label (typically Windows or Linux) to select which OS to boot. If you set the delay to 'Forever', the system will display a prompt at boot time and wait for you to enter a choice.

Next, you need to add entries for each operating system that LILO can boot. The first entry you make will be the machine's default operating system. You can add either a DOS, Linux, or Windows partition first. For example, let's say you select "Linux". The system will display your Linux partitions and ask which one of them you'd like to boot. Enter the name (like /dev/hda1) of your root Linux partition. Then, you'll be prompted to enter a label. This is the name you will enter at the boot time LILO prompt to select which partition you want to boot. A good choice for this is "Linux".

Adding a DOS or Windows partition is similar. To add a Windows partition to the LILO configuration file, select the Windows option. The system will display your FAT/NTFS partitions and ask which one of them you'd like to boot with LILO. Enter the name of your primary Windows partition. Then enter a label for the partition, like "Windows". Once you've added all of your bootable partitions, install LILO by selecting the "Install" option.

4.8 Networking

Another configuration menu allows you to configure your machine's networking setup. First, enter a hostname for your machine. The default hostname after installation is "darkstar," but you can enter any name you like. Next, you'll be asked to provide a domain name. If you're running a stand-alone machine (possibly using a dialup link to an Internet Service Provider) then you can pick any name you like. The default domain name is "example.net". If you are going to add the machine to a local network, you'll need to use the same domain name as the rest of the machines on your network. If you're not sure what this is, contact your network administrator for help. Once you've specified the hostname and domain name, you'll be asked which type of setup you would like: "static IP", "DHCP", or "loopback".

Loopback

This is the simplest type of setup, defining only a mechanism for the machine to contact itself. If you do not have an Ethernet card, use this selection. This is also the correct selection if you'll be using a PCMCIA (laptop) Ethernet card and want to set up your networking in /etc/pcmcia/network.opts. (you could also configure a PCMCIA card using the "static IP" or "DHCP" options, but in that case will not be able to "hotplug" the card) Finally, this is the right option to use if you have a modem, and will be connecting via dialout and PPP. You'll select loopback now, and then set up your phone connection later using pppsetup or kppp.

Static IP

If your machine has an Ethernet card with a static IP address assigned to it, you can use this option to set it up. You'll be prompted to enter your machine's IP address, netmask, the gateway IP address, and the nameserver IP address. If you don't know what numbers you should be using, ask the person in charge of the network to help. After entering your information, you'll be asked if you want to probe for your network card. This is a good idea, so say yes. Confirm that the settings are correct, and your networking will be configured to use a static IP address.

DHCP

DHCP stands for Dynamic Host Configuration Protocol, and is a system where your machine contacts a server to obtain its IP and DNS information. This is the usual way to get an IP address with broadband connections like cable modems (although some more expensive business-class broadband connections may assign static IP addresses). It is very easy to set up a DHCP connection – just select the option. Some providers will give you a DHCP hostname (Cox is one that does) that you'll also need to enter in order to identify yourself to the network. If you don't have a DHCP hostname, just leave it blank and hit ENTER. After entering your information, you'll be asked if you want to probe for your network card. This is a good idea, so say yes. Confirm that the settings are correct, and your networking will be configured to use DHCP.

Once you've completed all the configuration menus, you can exit setup and reboot your machine. Simply press ctrl-alt-delete and the kernel will kill any programs that are running, unmount your filesystems, and restart the machine.

5. Booting the installed Slackware system

If you've installed LILO, make sure you don't have a disk in your floppy drive – when your machine reboots it should start LILO. Otherwise, insert the bootdisk made for your system during the configuration process and use it to boot. Also, make sure to remove the CD-ROM to avoid booting it, or disable your machine's CD-ROM booting feature in the BIOS settings.

The kernel will go through the startup process, detecting your hardware, checking your partitions and starting various processes. Eventually you'll be given a login prompt:

darkstar login:

Log into the new system as “root”.

```
Welcome to Linux 2.6.33.4.
```

```
darkstar login: root
Last login: Tue May 18 15:36:23 2010 on tty3.
```

```
Linux 2.6.33.4.
You have new mail.
```

```
darkstar: ~#
```

6. Post-installation configuration

Once the system is running, most of the work is complete. However, there are still a few programs you'll need to configure. We'll cover the most important of these in this section.

6.1 /etc/rc.d/rc.modules

This file contains a list of Linux kernel modules. A kernel module is like a device driver under DOS. You can think of the `/etc/rc.d/rc.modules` file as similar to DOS's `CONFIG.SYS`. The file specifies which modules the system needs to load to support the machine's hardware. After booting your machine, you may find that some of your hardware isn't detected (usually an Ethernet card). To provide the support, you'll need to load the correct kernel module. Note that modern Linux kernels include a feature that allows the kernel to load its own modules, called `udev`. This will load many modules automatically without any need to edit `rc.modules`, and when using `udev` it might be better to tell it how to load the modules you want automatically rather than loading them at boot time with `rc.modules`. This is an advanced topic, and outside the scope of this document. If you're interested in this, "`man udev`" is a good place to start reading. In any case, it's best to not edit `rc.modules` unless you find that the modules you want to use are not being loaded automatically by `udev`. You can see a list of the modules that were loaded with the "`lsmod`" command. Likewise, in the majority of cases "`alsaconf`" is not required to configure sound. Rather, the "`alsamixer`" tool is used to unmute the Master and PCM channels and turn up the volume, and the "`alsactl store`" is used to save the sound defaults.

There's a lot more information out there about kernel modules, including lists of module names and the cards they support, as well as extra options you can add to the module lines to configure the hardware in different ways. The kernel's documentation in `/usr/src/linux/Documentation` has a lot of good information, as does the information shipped with `udev` (found under `/usr/doc/udev-*`).

6.2 Configuring the X Window System

Configuring X can be a complex task. The reason for this is the vast numbers of video cards available for the PC architecture, most of which use different programming interfaces. Luckily, X has come a long way since the early days of X386, where monitor modelines had to be tediously calculated. With most hardware, X can now be run with NO configuration file or additional driver! But you still might want to make a configuration file if you'll be using a third party video driver (the installer for that may offer to make it for you), or if you just want to have greater control over the details of the X configuration.

To try X without a configuration file, just type "`startx`" at a command line. If you're satisfied with the result, then you're done. If you would like X to start automatically at boot, see the `/etc/inittab` file once you've tested "`startx`" to be sure that X is working.

If this doesn't work with your card, or if you'd like to take advantage of the high-performance features of your video card such as hardware acceleration or 3-D hardware rendering, then you'll need to reconfigure X.

To configure X, you'll need to make an `/etc/X11/xorg.conf` file. This file contains lots of details about your video hardware, mouse, and monitor. It's a very complex configuration file, but fortunately there are several programs to help create one for you. We'll mention a few of them here:

Xorg -configure

Modern versions of X provide a simple way to create an initial `xorg.conf` file that often will work without any additional configuration, or, at the very least, provide a good base from which to customize the file. To run this command, enter the following in a root terminal:

```
# Xorg -configure
```

The X server probes for available hardware and creates an initial `xorg.conf` file located in the `/root` directory. You can then use this initial file to test the configuration by entering the following:

```
# Xorg -config /root/xorg.conf.new
```

This will load the initial `xorg.conf.new` file and run the X server. If you see the default black and gray checkered background with a mouse cursor appear, then the configuration was successful. To exit the X server, just press `Ctrl+Alt+Backspace` simultaneously. Once back at the command line, you can copy this `xorg.conf.new` file to `/etc/X11/xorg.conf` and begin making any manual edits necessary to customize your setup.

```
xorgsetup
```

This is a simple menu driven frontend that's similar in feel to the Slackware installer. It simply tells the X server to take a look at the card, and then set up the best initial configuration file it can make based on the information it gathers. The generated `/etc/X11/xorg.conf` file should be a good starting point for most systems (and should work without modification).

6.3 Hardware acceleration with X

If you've used `xorgsetup` or `X -configure` to configure for your card, and it's one that can take advantage of X's direct rendering support, you'll certainly want to enable this. Check your `/etc/X11/xorg.conf` and make sure that the `glx` module is loaded:

```
Load      "glx"
```

This line will probably already be in place.

6.4 User Accounts

You should make a user account for yourself. Using "root" as your everyday account is dangerous, and is considered bad form (at the very least) since you can accidentally damage your system if you mistype a command. If you're logged in as a normal user, the effects of bad commands will be much more limited. Normally you'll only log in as root to perform system administration tasks, such as setting or changing the root password, installing, configuring, or removing system software, and creating or deleting user accounts.

To make an account for yourself, use the 'adduser' program. To start it, type 'adduser' at a prompt and follow the instructions. Going with the default selections for user ID, group ID, and shell should be just fine for most users. You'll want to add your user to the `cdrom`, `audio`, `video` `plugdev` (pluggable devices like USB cameras and flash memory) and `scanner` groups if you have a computer with

multimedia peripherals and want to be able to access these. Add these group names, comma separated, at the following prompt:

```
Additional groups (comma separated) []:
```

Passwords and security

When choosing passwords for a Linux system that is connected to a network you should pick a strong password. However, passwords only help protect a system from remote trespassing. It's easy to gain access to a system if someone has physical access to the console.

If you forget the root password, you can use the install disc to mount your root partition and edit the files containing the password information. If you have a bootable optical drive, you can use the first installation CD-ROM or the DVD as a rescue disk.

At the prompt, you can manually mount the root Linux partition from your hard drive ("fdisk -l" will give you a list) and remove the root password. For example, if your root linux partition is /dev/hda2, here are the commands to use after logging into the install disc as "root":

```
mount /dev/hda2 /mnt  
cd /mnt/etc
```

Next, you'll need to edit the "shadow" file to remove root's password. Editors which might be available include "vi", "emacs", "pico", and "nano". "vi" and "emacs" might be more of an adventure than you need unless you've used them before. The "pico" and "nano" editors are easy for beginners to use.

```
pico shadow
```

At the top of the file, you'll see a line starting with root. Right after root, you'll notice the encrypted password information between two colons. Here's how root's line in /etc/shadow might look:

```
root:EnCl6vi6y2KjU:10266:0:::::
```

To remove root's password, you use the editor to erase the scrambled text between the two colons, leaving a line that looks like this:

```
root::10266:0:::::
```

Save the file and reboot the machine, and you'll be able to log in as root without a password. The first thing you should do is set a new password for root, especially if your machine is connected to a network.

Here are some pointers on avoiding weak passwords:

1. Never use your name (or anyone's name), birthdate, license plate,

```
or anything relating to yourself as a password. Someone trying  
to break into your machine might be able to look these things up.
```

2. Don't use a password that is any variation of your login name.
3. Do not use words from the dictionary (especially not "password" :)

or syllables of two different words concatenated together as your password. There are automated programs floating around on the net that can try them all in a short time.

4. Do not use a number (like 123456) or a password shorter than six characters.

The strongest passwords are a mix of letters, numbers, and symbols. Here are some examples of strong passwords (but don't use these 😊):

- ^5g!1? ()lsp@@9 i8#6#1*x ++c\$!jke *2zt/mn1

In practice, any password containing one or two words, a number (or two), and a symbol (or two) should be quite secure.

7. For more information

For more information, visit our web site at <http://www.slackware.com>

To shop for fine Slackware products (and help keep the project funded), please visit <http://store.slackware.com>. 😊

| | | |
|------------------|--|------------------------------------|
| Email: | info@slackware.com | (Information or general inquiries) |
| FTP: | ftp://ftp.slackware.com | (Updates) |
| WWW: | http://www.slackware.com | (News) |
| Security issues: | security@slackware.com | |
| General Hotline: | volkerdi@slackware.com | |

8. Trademarks

Slackware is a registered trademark of Slackware Linux, Inc. Linux is a registered trademark of Linus Torvalds. All trademarks are property of their respective owners.

From:
<https://docs.slackware.com/> - **SlackDocs**

Permanent link:
<https://docs.slackware.com/playground:howto>

Last update: **2012/08/19 23:36 (UTC)**

