Hard Disk Upgrade Mini How−To

Table of Contents

Hard Disk Upgrade Mini How−To

By Yves Bellefeuille <yan@storm.ca>, and Konrad Hinsen <hinsen@cnrs−orleans.fr>

v2.11, 13 April 2000

How to copy a Linux system from one hard disk to another

This document explains how to transfer, or migrate, an entire Linux system, including LILO, from one hard disk to another.

In the following explanation, "`/dev/hda`" (first IDE hard disk) means the "old" disk, and "`/dev/hdb`" (second IDE hard disk) means the "new" disk.

Specific partitions on the "old" disk are referred to as "`/dev/hda1`", "`/dev/hda2`", and so on. Specific partitions on the "new" disk are referred to as "`/dev/hdb1`", "`/dev/hdb2`", and so on.

The explanations in this document are based on Red Hat 6.0. They have also been tested with Debian 2.1, Slackware 3.5 and SuSE 6.2; we indicate a few differences to note if you're using those distributions.

If the commands don't work properly on your system, please let us know, mentioning what distribution you're using.

---

1. Install both disks on your system

Modern systems can accept four "EIDE" devices on the hard disk controller, so there shouldn't be any problem installing both disks on your system at the same time, even if you also have other EIDE devices. Hard disks and CD−ROM drives are typical EIDE devices. Floppy drives and tape drives are usually connected to the floppy drive controller rather than to the hard disk controller.

SCSI adapters are even more flexible and can accept seven devices. If you're lucky (and rich) enough to have a

SCSI adapter, you probably already know this, and you probably know which of your devices are SCSI devices! For more information, see the SCSI How−To.

Even the oldest systems can accept two devices on the hard disk controller, so you can still install both hard disks at the same time. However, if you already have another device installed in addition to your hard disk (for example, if you have both a hard disk and a CD−ROM drive), you'll have to remove the other device to be able to install the old hard disk and the new hard disk at the same time.

You must configure the disks as "master" or "slave" by installing the disks' jumpers as appropriate. You'll often find configuration information on the disks themselves; if not, consult the manuals or the disks' manufacturers.

You must also inform the BIOS of the disks' presence and of their "geometry". Usually, you enter the BIOS setup programme by pressing a key during the system boot−up. Here's what to do for some common BIOSes:

> Acer notebooks: F2 key during Power−On Self−Test (POST)
>
> American Megatrends (AMI): Del key during Power−On Self−Test
>
> Award: Del, or Ctrl−Alt−Esc
>
> Compaq: F10 key after the square appears in the top right corner of the screen during boot−up
>
> Dell: Ctrl−Alt−Enter
>
> DTK: Esc key during Power−On Self−Test
>
> Hewlett−Packard Pavilion: F1 key during HP blue splash screen
>
> IBM Aptiva 535: F1 while the square with the wavy lines is displayed in the upper right corner during power−on
> [Does this also apply to other Aptiva models?]
>
> IBM PS/2: Ctrl−Alt−Del, then Ctrl−Alt−Ins when the cursor is in the top right corner
>
> Mr. BIOS: Ctrl−Alt−S during Power−On Self−Test
>
> Packard Bell: For some models, F1 or F2 key during Power−On Self−Test
>
> Phoenix: Ctrl−Alt−Esc, or Ctrl−Alt−S, or Ctrl−Alt−Enter
>
> Many older systems require an Installation or Reference Disk.

(We're interested in receiving information on other BIOSes to add them to this list.)

Reboot the system and login as `root`. If you use the "`su`" command to become the user `root`, use "`su -`", with the hyphen option.

2. Unmount non−Linux partitions and network drives

Some people like to mount partitions from other operating systems (DOS, Windows, OS/2, etc.) so they can use them under Linux. These partitions must be created and copied under their own operating system, and you should unmount them before copying your Linux partition. For example, if you have a DOS partition mounted at `/dos`, you must unmount it with this command:

```
umount /dos
```

Note that the command is "`umount`", without the first letter "n" in the word "unmount".

You should also unmount network drives.

---

3. Partition the new disk

Use this command to partition the new disk:

```
fdisk /dev/hdb
```

EIDE devices are identified as `hda`, `hdb`, `hdc`, and `hdd` in the `/dev` directory. Partitions on these disks can range from 1 to 16 and are also in the `/dev` directory. For example, `/dev/hda4` refers to partition 4 on hard disk `a` (first EIDE hard disk).

SCSI devices are listed as devices `sda`, `sdb`, `sdc`, `sdd`, `sde`, `sdf`, and `sdg` in the `/dev` directory. Similarly, partitions on these disks can range from 1 to 16 and are also in the `/dev` directory. For example, `/dev/sda3` refers to partition 3 on SCSI disk `a` (first SCSI hard disk).

For Linux partitions with the `ext2` file system, use system ID 83. For swap partitions, use system ID 82.

For more information on partitioning, see the Installation How−To and the Partition Mini How−To.

If your new disk has over 1024 cylinders, see the Large Disk Mini How−To. In brief, you should install all files required to boot Linux within the first 1024 cylinders. One way to do this is to create a small partition (5 Mb or so) just for the `/boot` directory at the beginning of the disk. (***Slackware only:*** The kernel is at `/vmlinuz` rather than `/boot/vmlinuz`, so you should put both the `/` directory and the `/boot` directory in this partition.)

Partitions for systems other than Linux should be created using their own `fdisk` or equivalent command rather than with Linux's `fdisk`.

---

4. Format the new disk

Use the following command to format Linux partitions using `ext2fs` on the new disk:

```
mkfs.ext2 /dev/hdb1
```

To check the disk for bad blocks (physical defects), add the `-c` option just before "`/dev/hdb1`".

If the new disk will have more than one Linux partition, format the other partitions with "`mkfs.ext2 /dev/hdb2`", "`mkfs.ext2 /dev/hdb3`", and so on. Add the `-c` option if desired.

(Note: With older distributions, the command "`mkfs -t ext2 -c /dev/hdb1`" didn't check for bad blocks under any of Red Hat, Debian or Slackware, contrary to what the `man` page stated. This has now been fixed.)

To format a swap partition, use this command:

```
mkswap /dev/hdb1
```

Again, you can add the `-c` option before "`/dev/hdb1`" to check for bad blocks.

---

5. Mount the new disk

Create a directory where you'll mount the new disk, for example `/new-disk`, and mount it there:

```
mkdir /new-disk
mount -t ext2 /dev/hdb1 /new-disk
```

If the new disk will have more than one Linux partition, mount them all under `/new-disk` with the same organization they'll have later.

Example: The new disk will have four Linux partitions, as follows:

```
/dev/hdb1:    /
/dev/hdb2:    /home
/dev/hdb3:    /var
/dev/hdb4:    /var/spool
```

Mount the four partitions under `/new-disk` as follows:

```
/dev/hdb1:    /new-disk
/dev/hdb2:    /new-disk/home
/dev/hdb3:    /new-disk/var
/dev/hdb4:    /new-disk/var/spool
```

You must create the mount points for each "level" before you mount the partitions at that level. Example:

```
mkdir /new-disk                                   [1st level]
mount -t ext2 /dev/hdb1 /new-disk

mkdir /new-disk/home                              [2nd level]
mount -t ext2 /dev/hdb2 /new-disk/home

mkdir /new-disk/var                               [2nd level also]
mount -t ext2 /dev/hdb3 /new-disk/var

mkdir /new-disk/var/spool                         [3rd level]
mount -t ext2 /dev/hdb4 /new-disk/var/spool
```

If you've created a mount point at `/new-disk/tmp`, you'll need to correct the directory's permissions to let all users access it:

```
chmod 1777 /new-disk/tmp
```

---

6. Copy the files from the old disk to the new disk

You might want to go to single–user mode before starting to copy the disk, in order to shut down the system daemons and preserve the state of the logs, and to prevent users from logging in:

```
/sbin/telinit 1
```

When copying the hard disk, you want to copy all directories and files, including links.

However, you *don't* want to copy the directory `/new-disk`, since this would copy the new disk to itself!

Furthermore, you want to create the `/proc` directory on the new disk, but you don't want to copy its contents: `/proc` is a "virtual" file system and doesn't have any actual files, but rather contains information on the processes running on the system.

Here are three different ways to copy the old disk to the new one. This may take quite a while, especially if you have a large disk or little memory. You can expect to be able to copy 10 Mb per minute, and possibly much more.

You can follow the copy's progress by using the command "`df`" from another terminal. Try "`watch df`" or "`watch ls -l /new-disk`" to see a report updated every two seconds; press `Ctrl-C` to end the display. Be aware that running the "`watch`" programme itself will slow down the copying.

i. `cp -ax / /new-disk`

   This is the simplest method, but will only work if your original Linux system is on a single disk partition.

The `−a` option preserves the original system as much as possible. The `−x` option limits `cp` to a single file system; this is necessary to avoid copying the `/new−disk` and `/proc` directories.

(***SuSE only:*** With this method only, you must also create the directory `/dev/pts` on the new disk. Use the command "`mkdir /new−disk/dev/pts`".)

(Note: When using the `−x` option, recent versions of `cp` will create the directories `/new−disk/new−disk` and `/new−disk/proc`, although the directories will be empty. If these directories are created, you should delete `/new−disk/new−disk`, and keep `/new−disk/proc`.)

ii. `cd / &echo cp −a `/bin/ls −1Ab | egrep −v "^new−disk$|^proc$"` /new−disk | sh`

(write this all on one line)

This goes to the root directory and then copies all files and directories except `/new−disk` and `/proc` to `/new−disk`. Note that the first option after `ls` is the number 1, not the letter L!

This command should work in all circumstances.

iii. `cp −a /bin /boot /dev /etc /home /lib /lost+found /mnt /root /sbin /tmp /usr /var /new−disk`

(write this all on one line)

The last directory, `/new−disk`, is the destination for the `cp` command. All the other directories are the sources. Therefore, we're copying all the directories we're listing to `/new−disk`.

With this method, you simply list yourself the directories you want to copy. Here we listed all the directories except `/new−disk` and `/proc`. If you can't use the other methods for any reason, you can always use this command to manually specify the directories you want to copy.

With this method only, if there are any files in the root directory itself, you need another command to copy them. In particular, this is required with Debian and Slackware, since these distributions put files in the root directory:

```
cp −dp /* /.* /new−disk
```

Previous versions of the Mini How−To stated that you could also use "`tar`" to copy the disk, but this method was found to have a bug. There are of course many other ways to copy the disks, but these three are the simplest, quickest, and most reliable.

After using any of these three methods, you must also create the `/proc` directory on the new disk, if it doesn't already exist:

```
mkdir /new−disk/proc
```

At this point, you may verify the file structure on the new disk, if you wish:

```
umount /new−disk
fsck.ext2 −f /dev/hdb1
mount −t ext2 /dev/hdb1 /new−disk
```

If the new disk has more than one partition, you must unmount them from the "bottom up" before running `fsck.ext2`: in the example mentioned above, you'd first unmount the 3rd level partitions, then the 2nd level partitions, and then the 1st level partition.

You may also compare the two disks, to ensure that the files were copied properly:

```
find / −path /proc −prune −o −path /new−disk −prune −o −xtype f
−exec cmp {} /new−disk{} \;
```

(write this all on one line)

(***Slackware only:*** A basic Slackware installation ("A" series only) doesn't include the "`cmp`" command, so you won't be able to run this command if you have only installed the basic files. The "`cmp`" command is in the "AP1" series.)

This will only compare regular files, not character or block special files (in the `/dev` directory), sockets, etc., since the "`cmp`" command doesn't work properly with these. We would welcome suggestions on how to verify these "special" files.

---

7. Modify `/new−disk/etc/fstab` as appropriate

If your new disk doesn't have the same partitions or organization as the old disk, modify the file `/new−disk/etc/fstab` on the new disk as appropriate.

Make sure that the disk partitions (first column) correspond to the organization you'll have with the new disk, once the old disk has been removed, and that you're only mounting one partition at "`/`" as shown in the second column.

For more information on the format of the file `/etc/fstab`, see the Linux System Administrator's Guide, section 4, under "Mounting and unmounting".

For swap partitions, use a line similar to this one:

```
/dev/hda1    swap     swap     defaults     0   0
```

8. Prepare LILO to boot the new disk

(Thanks to Rick Masters for helping with this.)

We're assuming that LILO is installed on the hard disk's Master Boot Record (MBR); this seems to be the most common configuration. You want to install LILO on what's presently the second hard disk but will become the first hard disk.

Edit the file `/new−disk/etc/lilo.conf` as follows:

```
disk=/dev/hdb bios=0x80       # Tell LILO to treat the second
                              # disk as if it were the first
                              # disk (BIOS ID 0x80).
boot=/dev/hdb                 # Install LILO on second hard
                              # disk.
map=/new-disk/boot/map        # Location of "map file".
install=/new-disk/boot/boot.b # File to copy to hard disk's
                              # boot sector.
prompt                        # Have LILO show "LILO boot:"
                              # prompt.
timeout=50                    # Boot default system after 5
                              # seconds. (Value is in tenths of
                              # seconds.)
image=/new-disk/boot/vmlinuz  # Location of Linux kernel. The
                              # actual name may include a version
                              # number, for example
                              # "vmlinuz-2.0.35".
    label=linux               # Label for Linux system.
    root=/dev/hda1            # Location of root partition on
                              # new hard disk. Modify this as
                              # appropriate for your system.
                              # Note that you must use the name
                              # of the future location, once the
                              # old disk has been removed.
    read-only                 # Mount partition read-only at
                              # first, to run fsck.
```

(***Slackware only:*** Use "`image=/new−disk/vmlinuz`".)

If you're using a SCSI hard disk, you may have to add a line with "`initrd`". See your existing file `/etc/lilo.conf`.

Install LILO on the new disk:

```
/sbin/lilo -C /new-disk/etc/lilo.conf
```

The `−C` option tells LILO what configuration file to use.

9. Make a boot diskette (optional)

If you wish, you can make a boot diskette, in case you run into problems when trying to boot the new disk.

Insert an empty diskette, format it, create a file system on it and mount it:

```
fdformat /dev/fd0H1440
mkfs.ext2 /dev/fd0
mount −t ext2 /dev/fd0 /mnt
```

(***Debian only:*** With Debian 2.x, use "`/dev/fd0u1440`" instead of "`/dev/fd0H1440`". With Debian 1.x, use "`/dev/fd0h1440`", with a lower case "h".)

(***Debian only:*** With Debian 2.x, use "`superformat`" instead of "`fdformat`". You can ignore the error "`mformat: command not found`". With Debian 1.x, if you don't have the command "`fdformat`", you can omit it if the floppy is already formatted. In this case, you should check the diskette for bad blocks by adding "`−c`" after the "`mkfs.ext2`" command.)

(***Slackware only:*** Use "`/dev/fd0u1440`" instead of "`/dev/fd0H1440`". With older versions, try "`/dev/fd0h1440`", with a lower case "h".)

(***SuSE only:*** Use "`/dev/fd0u1440`" instead of "`/dev/fd0H1440`".)

Copy all files in `/boot` to the diskette:

```
cp −dp /boot/* /mnt
```

(***Red Hat only:*** If the `/boot` directory contains both `vmlinux` and `vmlinuz` files (note the difference in the last letter), you only need to copy the `vmlinuz` files to the boot diskette. They are the same as the `vmlinux` files, except that they're compressed to save space.)

(***Slackware only:*** Copy the file `/vmlinuz` to the boot diskette; use the command "`cp /vmlinuz /mnt`".)

Create a new file `/mnt/lilo.conf` as follows:

```
boot=/dev/fd0                # Install LILO on floppy disk.
map=/mnt/map                 # Location of "map file".
install=/mnt/boot.b          # File to copy to floppy's
                             # boot sector.
prompt                       # Have LILO show "LILO boot:"
                             # prompt.
timeout=50                   # Boot default system after 5
                             # seconds. (Value is in tenths of
                             # seconds.)
image=/mnt/vmlinuz           # Location of Linux kernel on
                             # floppy. The actual name may
                             # include a version number, for
                             # example "vmlinuz−2.0.35".
    label=linux              # Label for Linux system.
    root=/dev/hda1           # Location of root partition on
```

```
                                        # new hard disk. Modify this as
                                        # appropriate for your system.
                                        # Note that you must use the name
                                        # of the future location, once the
                                        # old disk has been removed.
    read−only                           # Mount partition read−only at
                                        # first, to run fsck.
```

Install LILO on the boot diskette:

```
/sbin/lilo −C /mnt/lilo.conf
```

The `−C` option tells LILO what configuration file to use.

Unmount the diskette:

```
umount /mnt
```

---

10. Remove the old disk

Shut down the system and remove the old disk. Remember to modify the disk jumpers and the BIOS information to reflect the changes.

---

11. Reboot the system, modify LILO configuration file

Reboot the system. If you have problems, you can use the boot diskette you just make. To do so, you may have to modify your BIOS's boot−up sequence to "`A:, C:`".

You should modify the file `/etc/lilo.conf` in case you later want to run LILO again. Here's an example of how the file can look:

```
boot=/dev/hda                   # Install LILO on first hard
                                # disk.
map=/boot/map                   # Location of "map file".
install=/boot/boot.b            # File to copy to floppy's
                                # boot sector.
prompt                          # Have LILO show "LILO boot:"
                                # prompt.
timeout=50                      # Boot default system after 5
                                # seconds. (Value is in tenths of
                                # seconds.)
image=/boot/vmlinuz             # Location of Linux kernel. The
                                # actual name may include a version
                                # number, for example
                                # "vmlinuz−2.0.35".
    label=linux                 # Label for Linux system.
    root=/dev/hda1              # Location of root partition on
```

```
                                  # new hard disk. Modify this as
                                  # appropriate for your system.
               read-only          # Mount partition read-only at
                                  # first, to run fsck.
```

(***Slackware only:*** Use "`image=/vmlinuz`".)

Acknowledgements:

Thanks to Scott Christensen, Frank Damgaard, Alexandre Fornieles, David Fullerton, Igor Furlan, Jerry Gaines, Chris Gillespie, Nicola Girardi, Per Gunnar Hansö, Richard Hawes, Ralph Heimueller, Gerald Hermant, Andy Heynderickx, Paul Koning, Hannu Liljemark, Claes Månsson, Rick Masters, Jason Priebe, Josh Rabinowitz, Douglas Rhodes, Valentijn Sessink, Kragen Sitaker, Stephen Thomas, and Gerald Willman.

This document may be translated into any language. If you do so, please send a copy of the translation to Konrad Hinsen <hinsen@cnrs−orleans.fr>.