

Sat Apr 2 19:32:34 PST 1994

This is an unofficial HOWTO that describes how to use 'setup' and 'pkgtool' to upgrade your Linux system.

One of the most commonly asked questions about Slackware (especially after any major release) is: How do I upgrade my system?

Until recently, it seemed there were two routes you could go. You could back up everything you wanted to save, and then reinstall from scratch. As anyone who's done this can attest, it's a tedious process and there's almost always a couple important files that don't get backed up. The other option was to install the new packages over the old system. This works, but any old files that aren't overwritten by new ones will be left in your filesystem. This wastes space, and makes system administration difficult, especially when configuration files that are ignored by the new binaries are left behind.

The new versions of 'pkgtool' (a package maintenance tool developed for the Slackware distribution) should have provide a clean upgrade path from earlier versions of Slackware. Since it can now remove packages from your hard drive while running on a self-contained Linux filesystem loaded into a ramdisk, it can remove *any* files from your system, including ones that were difficult or impossible to remove while running on the harddrive. Files such as the shell, shared libraries, init, and other crucial system files.

Here's how you'd upgrade to a newer version of Slackware from any previous version that supports package information files in /var/adm/packages. (If your system puts these files elsewhere, you might still be able to do this by creating a symbolic link from the package informataion directory to /var/adm/packages)

1. Back up important files, or take your chances. ;^)

Odds are you'll come through ok. However, there are two important exceptions to this rule. The first (and most obvious) is when a package overwrites a file you meant to keep with a new one. The second, and possibly more serious situation is when the system needs to replace an existing file with a symbolic link. It *will* replace the file, whether it's a simple file, a file that's chmoded 444, or a directory filled with other subdirectories, each containing part of your Doctoral dissertation. So, be careful.

2. Make a list of the packages you plan to replace.
3. Use a bootkernel disk to boot one of the root/install disks. Log in as root.
4. Mount your root Linux partitions under /mnt while logged into the install disk. The method used here differs depending on what filesystem you're using for Linux. Here are some examples:

How to mount an ext2fs partition:

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

Replace this with the name of your root partition.

Similarly, if the partition was of type xiafs, you would use this command:

```
mount /dev/hda1 /mnt -t xiafs
```

If you're using UMSDOS (the system that allows you to install onto an

existing MS-DOS filesystem), this is the command you would use:

```
mount /dev/hda1 /mnt -t umsdos
```

If you've got other partitions that are part of your Linux filesystem, mount them after you've mounted that root partition. The method is the same - for example, here's how you'd mount an ext2fs /usr partition:

```
mount /dev/hda2 /mnt/usr -t ext2
```

5. Once the partition has been mounted, we need to activate swap space if the system has less than 8 MB of memory. (If you have 8 or more MB, you may go on to step 6)

You may use either a swap partition or a swapfile. To get a quick listing of your partition information, you can always type 'fdisk -l'. Doing this on my machine provides the following information:

Disk /dev/hda: 15 heads, 17 sectors, 1001 cylinders
Units = cylinders of 255 * 512 bytes

Device	Boot	Begin	Start	End	Blocks	Id	System
/dev/hda1		10	10	90	10327+	1	DOS 12-bit FAT
/dev/hda2		91	91	1000	116025	5	Extended
/dev/hda3	*	1	1	9	1139	a	OPUS
/dev/hda5	*	91	91	1000	116016+	6	DOS 16-bit >=32M

Disk /dev/hdb: 16 heads, 31 sectors, 967 cylinders
Units = cylinders of 496 * 512 bytes

Device	Boot	Begin	Start	End	Blocks	Id	System
/dev/hdb1	*	1	1	921	228392+	6	DOS 16-bit >=32M
/dev/hdb2		922	922	966	11160	82	Linux swap

From this display, you can see that /dev/hdb2 has been designated as the Linux swap partition. If the partition has not been previously prepared with mkswap, here's how that would be done:

```
mkswap /dev/hdb2 11160
```

To activate the swap partition, you would type:

```
swapon /dev/hdb2
```

6. Remove the packages! To do this, type 'pkgtool' and select the option remove installed packages. You'll be given a list of packages that you've installed - just select the list of packages that you plan to replace.

If you're using one of the full-color versions of pkgtool, you select the packages to remove by removing up and down through the list with '+' and '-' and toggling packages to remove with the spacebar. Once you've toggled all the packages you want to remove, hit ENTER to remove them.

If you're using one of the tty based versions of pkgtool, you'll have to type in the names of the packages you wish to remove. Separate each name with a space. Don't worry about how long the line ends up - just keep typing in the names until you've entered them all, and then hit ENTER to remove them.

That's it! Now you've cleaned up the old packages and you're ready to install the new ones. Type 'setup' and proceed to install the new packages as normal.

Although it never hurts to play it safe and remove all packages from the bootdisk, almost all of them can be removed using pkgtool from your hard drive. The A series is the important exception here.

I wish everyone good luck with this! :^)

Patrick Volkerding
volkerdi@ftp.cdrom.com