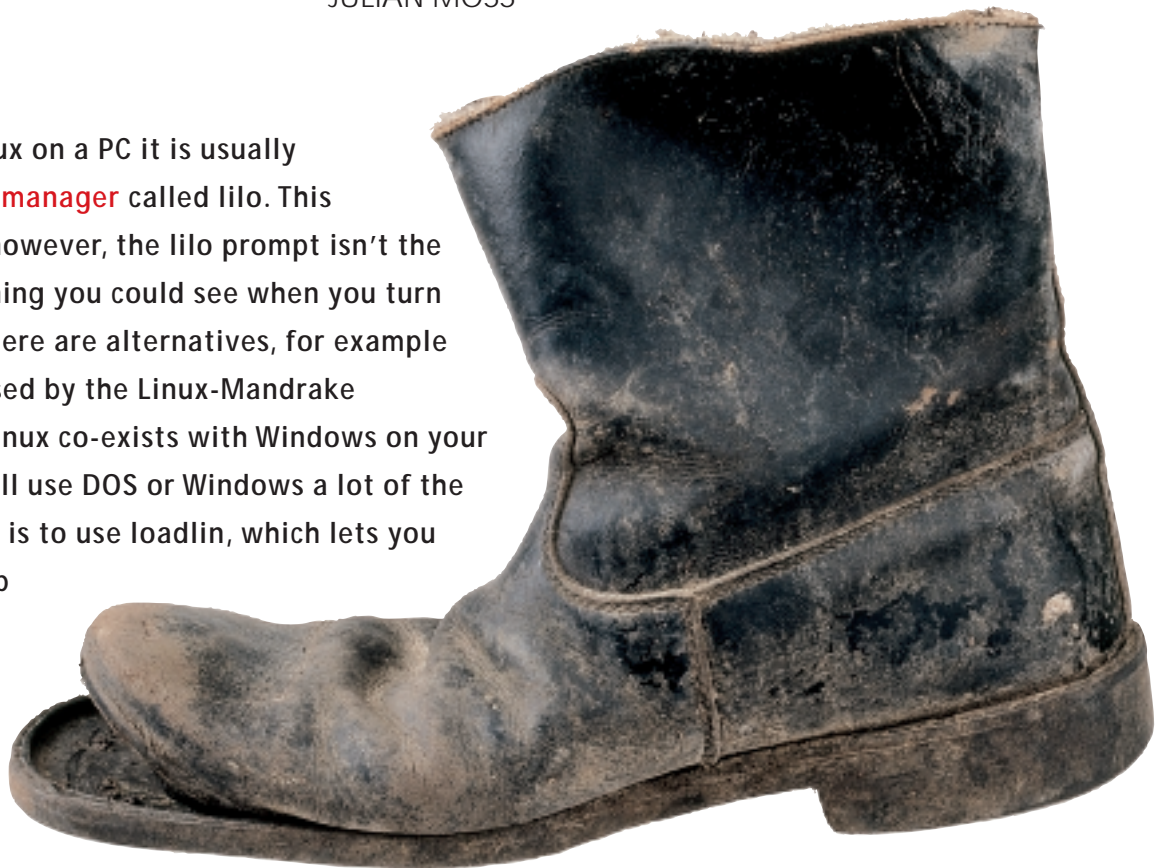


How To: Create your own boot menu

ALTERNATIVE BOOT

JULIAN MOSS

When you install Linux on a PC it is usually started using a **boot manager** called lilo. This usually works well; however, the lilo prompt isn't the most user-friendly thing you could see when you turn on your computer. There are alternatives, for example grub which is now used by the Linux-Mandrake distribution. But if Linux co-exists with Windows on your hard disk and you still use DOS or Windows a lot of the time a good solution is to use loadlin, which lets you manage your boot-up options using DOS.



The name *loadlin* is derived from "load Linux" and this describes exactly what the program does. It is an MS-DOS program that loads a Linux kernel into memory, thereby starting the boot process. It is convenient if you wish to run DOS and Windows as well as Linux. If your system is set up to boot into DOS, you can start Windows by typing "win" or you can start Linux by running "linux". Even more conveniently, you can arrange to select one of these choices from an MS-DOS boot menu.

Why would you want to do this, when *lilo* will do the same job and is a more commonly used solution? Well, for a start, *lilo* will only work if the Linux boot partition is contained within the first 1,024 **cylinders** of the hard disk. In the most common new-user scenario, in which Linux has been installed into the space made by shrinking an existing DOS partition

that occupied the whole disk, this is very often not the case, leading to the situation where the would-be newbie spends an hour installing Linux and then finds that they can't boot it.

Other benefits for the newcomer are that *loadlin* is easy to set up and allows you to change your boot options and kernel parameters using familiar DOS tools. It avoids changing the contents of the boot sector and the risk of losing the ability to access DOS and Windows. It also avoids the risk of losing the ability to boot Linux if you reinstall Windows in the future: for some reason, Microsoft doesn't think that anyone would want to use a non-Microsoft operating system and so when you install Windows, *lilo* or any other boot manager you are using are overwritten without warning. Because *loadlin* is an ordinary program it won't be

overwritten (unless you format the Windows partition, of course.)

Even some more experienced Linux users like to use *loadlin*. If you like experimenting with new Linux kernels you can easily create a system using batch files or a boot menu that lets you choose which of many kernels to use at boot-up. Or you can specify a different kernel on the command line. So all in all, if you use DOS at all, *loadlin* is a very flexible tool for loading Linux.

Installation



Where can you find *loadlin*? Well, if you have a set of CDs from a full distribution you'll probably find it on the first disk in a directory called something like */dosutils*. If you have a cover CD version like Linux Mandrake 7.1 from our issue 1 cover disk, you may not. If you can't find it, don't worry. We've included a copy on this month's cover CD for you.

Most distributions supply *loadlin* uncompressed in its own directory, ready to use, but it's possible to find it as a compressed archive called *loadlin16.tgz*. In that case you'll need to extract the files from the archive first using a command similar to:

```
tar xzf /mnt/cdrom/dosutils/loadlin16.tgz
```

Create a directory on your DOS disk called "linux" or something equally appropriate. (Keep the name to eight characters or less because you will be running this under real-mode DOS where there is no long filename support.) Copy to this directory the files *loadlin.exe*, *linux.bat*, and *test.par*. You could also copy the *doc* directory, which contains the full documentation for the utility. (You can probably manage without the *src* directory, which contains the assembly language source code!)

Next, place a copy of your current Linux kernel in the directory. If you are running Linux at the moment you can do this by mounting your DOS drive and copying the kernel directly from its location in */boot*. If you are logged in as *root* while doing this (which you shouldn't be!) and are using a graphical file manager be very careful that you copy the kernel instead of moving it. You'll also find a copy of the kernel on the boot floppy you made when installing Linux.

The kernel usually has a name like *vmlinuz-2.2.15-4mdk*, which is one of the stock kernels for Linux-Mandrake. The exact name will depend on both the distribution and the version. When copying it to the DOS directory give it a name eight characters long or less, such as *vmlinuz*.

Configuration

Now you must create a *loadlin* parameter file. It can be called anything you like, but for the sake of example we will call it *linux.par*. The file *test.par* that is usually included with *loadlin* is an example, so you could start by editing that.

A simple parameter file will look like this:

```
C:\Linux\vmlinuz # the first value must be
                  # the name of the Linux kernel

root=/dev/hda7   # the device which is mounte
ted as root FS

ro               # mount root read-only

mem=128M        # tell the kernel to use al
l the memory
```

The first three parameters are essential, and with the comments almost self-explanatory. The first value, which you may want to change, is the DOS path to the file containing the Linux kernel. To use a different kernel you could simply create a new parameter file containing a different filename.

The second value which starts with "root=" must specify the device which is mounted as root ("/"). If you can't remember what this is from when you installed Linux you can find out (if you're in Linux at the moment) by running the command *mount*. This will list all the mounted filesystems: the device you want is the one that is listed as "on /".

The third value is standard, and ensures that root is initially mounted read-only. You may not need any parameters after that. However, a common one if your system has more than 64MB of RAM is the "mem=" parameter which tells the kernel how much memory it should use. The stock kernels in most current distributions are compiled to use a maximum of 64MB by default, so without this parameter Linux may run a lot more slowly than it could. For a full description of the parameters recognised by *loadlin* or the kernel see the file *params.doc* in the *doc* subdirectory of the *loadlin* package.

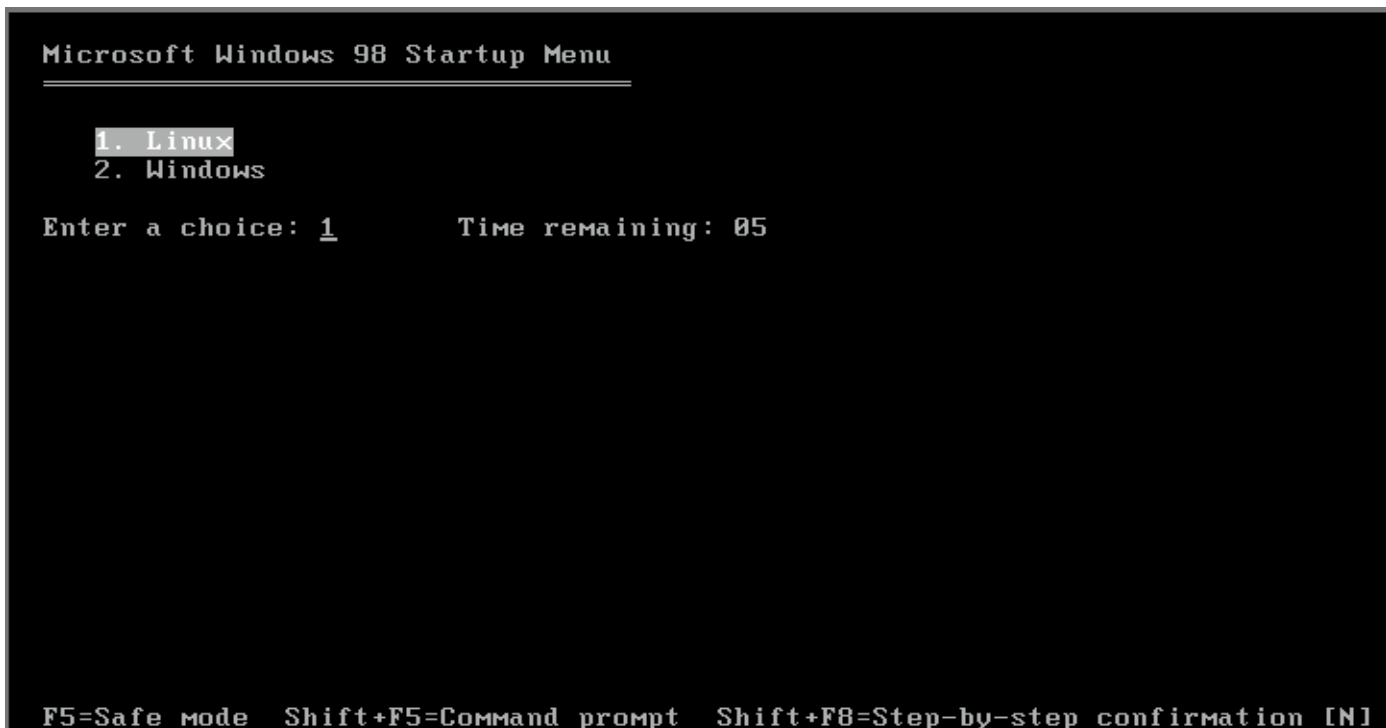
Boot manager: The PC's BIOS is designed to boot just one operating system using program code stored in the first sector (or boot sector) of the first disk **partition**.

A boot manager is a program that replaces the boot-up code in the boot sector and presents you with a choice of operating systems to boot from. Depending on your choice, it then loads the boot-up code from the appropriate partition.

BIOS: This stands for "Basic Input-Output System". It is program code stored in permanent read-only memory which is executed by the computer at start-up and enables it to access the main peripherals and load the operating system's boot-up code.

Partition: A hard disk is organised into one or more areas known as partitions. These partitions are then formatted for use by the operating system you want to use.

Cylinder: A hard disk is made up of one or more disks or platters coated with magnetic material, on to which data is recorded on concentric tracks. Although many modern hard disks have just a single platter, older ones were made up of several. A surface linking all the same-numbered tracks of all the platters would form a cylinder. From this the term is derived.



Starting Linux

Having done all this, and making sure that you are at a real DOS prompt and not a virtual one under Windows (booting Linux in the middle of a Windows session is not recommended, at least for the health of Windows) you can start Linux by executing the command:

```
LOADLIN @linux.par
```

assuming that you are in the directory containing both *loadlin.exe* and the parameter file and that the latter is called *linux.par*. If not, modify the paths accordingly.

In the *loadlin* directory you should see a batch file named *linux.bat*. You can edit this to contain the correct invocation of the above command, using full paths to both the program and its parameter file, and put it somewhere in the DOS search path. This will enable you to start Linux by just typing *linux*.

Boot menu

But there's an even better way. MS-DOS supports a facility that enables users to construct their own boot-up menus using commands in the file *config.sys*. You can use this facility to create a menu that lets you choose between Windows and Linux at boot-up. A simple example looks like this:

```
[MENU]
MENUITEM=LIN, Linux
MENUITEM=WIN, Windows
MENUDEFAULT=LIN, 10
[WIN]
REM OPTIONS NEEDED BY WINDOWS (IF ANY) GO HERE
[LIN]
SHELL=C:\Linux\loadlin.exe @C:\Linux\linux.par
[COMMON]
```

This example creates a boot menu with just two options, Windows and Linux (see Figure 1) and makes Linux the default (of course.) If Linux is chosen, DOS executes the commands under the section headed [LIN] which was associated with this option. The section contains a SHELL command that runs *loadlin* with the appropriate parameter file. As with the batch file previously mentioned, you'll need to change this line if the paths and parameter file name on your system are not the same.

Diehard Linux fans may object to the fact that the Microsoft Windows start-up logo briefly appears (which can be changed) or the fact that the menu is titled "Microsoft Windows 98 Startup Menu." But then diehard Linux fans wouldn't be running Windows in the first place, would they?

Conclusion

loadlin has many more options. For example, if you want to experiment with different Linux kernels you can run it using different parameter files. Or you can override the kernel file name parameter using a command line argument, like this:

```
LOADLIN @linux.par image=path_to_kernel
```

You can even run *loadlin* without a parameter file at all, specifying all the options on the command line (though be warned that DOS command lines are limited to 128 characters in length.) For more information see the *loadlin* documentation. However, the basic method of operation described here will be sufficient for most people's needs. If you run Windows or DOS on your computer as well as Linux you will probably find *loadlin* to be a very useful utility. ■

An MS-DOS start-up menu offering a choice of Linux or Windows