

Li(d)os

13 de agosto de 2002

This is a description of how to install FreeDos on a Linux Kernel and run the former as the only user shell - you can run several Dos'es on one computer this way. This document can be downloaded in a printable pdf version.

1. Motivation

Two challenges at once were too much: Linux is too heavy, bloated, and so on, old Hardware, small hardware is beauty and you have to use FreeDos or something like this to use them. Such said, and agreed, but frankly, DOS is not where my strengths lie.

Let's make a DOSBeowulf-Cluster, ten '386 are faster than one PentiumIV, was the other challenge, however, a mathematician does not want to become a Unix Administrator to do such harmless thing like massive parallelization! Dos would be so fine for clustering...

2. Involvement

Since years I'm experimenting with old hardware, varied hardware and making Linux slim. In reality I want an auto-installable Linux distribution with three hardware requirement levels: small ($\leq 486/66\text{Mhz}$, 6MByte), midi ($< \text{Pentium III}$, 64MByte) and the rest.

I use Debian Gnu Linux and it's fascinating good for anything one wants to do. Put I'm realistic: you have to know a lot about the distribution to get the power out of it, you get lost in almost 10.000 packages, *and* it needs tweaking to get it *really* to work - lot's of packages do not interact correctly or even act correctly, even if they would be very useful.

There is a small network clustering package collection I'm working on, maybe it will be called "club", and "smaco" is a "small-computer" setup, which brings a fast character-cell-gui menu to the untrained user, instead of the heavy X-windows graphical interface.

Now, as a third element I took up the challenge: My answer to reducing weight of the OS and to using a DOS interface is a heavily stripped down kernel which runs, just like the DOS kernel, FreeDOS as the init process.

Su, instead of a Debian Gnu Linux OS, you get a DOS Linux OS. As this gets you (the user) into heavy troubles, troubles is Lios in spanish, I call it: Li(d)os - Linux Dos.

3. Implementation Details

What you get from this website is a linux kernel without any Networking Code compiled in, this reduces size. It has 80386 processor support, as well as a Math-Coprocessor emulation, so you can run it on any (intel 32Bit) hardware. The PCI bus is supported, but not EISA and Microchannel. No sound support, no USB support, just IDE Disk Drives and Floppies, Serial and parallel ports. The kernel is not modular, it's one static executable without loadable parts. All this is to accomplish minimum size and fuss.

Any hardware configuration has to be done on the boot-loader commandline, or within the "append" statement in the `/etc/lilo.conf` file.

The compiled in file systems are: FAT (all types), ext2, ext3-extension, Reiser filesystem, and CD-ROM. In principle you can use any of them to launch the system.

Instead of launching the SysV init system, a Bourne-Shell script is launched as the "init" process by the kernel at startup, which set's up a basic environment (PATH and TERM), and then spawns a dosemu process in a new virtual terminal.

I leave a bash-shell running on the first terminal, so unixers can troubleshoot the system there ways, but it can be eliminated by commenting out two lines in the script at `/usr/local/bin/dosinit`.

4. Installation

4.1. Overview

1. Prime the computer with a minimum Debian Gnu Installation
2. Install FreeDos for use with Dosemu
3. Install the Li(d)os Kernel and scripts
4. Reboot and get your trouble delivered

4.2. Debian Installation

You need a minimal Debian Gnu Linux Installation on a disk partition. The partition can be as small as 300MByte. You can do it even on approximately 120MByte, but you'll require skills for this.

Install everything in one and only one root-partition “/”. After the normal installation be sure to convert the ext2 partition to ext3. You can do this with the following commands (assuming /dev/hda1 holds the root filesystems):

```
tune2fs -j /dev/hda1
tune2fs -i 0 /dev/hda1
tune2fs -c 0 /dev/hda1
```

Edit the file /etc/fstab, the line containing the root mount point has to look like the following example:

```
/dev/hda1 / auto defaults 0 0
```

In short, you replace the third word: “ext2”, with “auto”.

Swapspace is not needed in a DOS environment, is it? If you have 16MByte RAM or more it should not matter, if it is less I guess you should use a small swap partition.

When the installation program asks you to create an additional user, you can safely say “no”.

You'll need to install the following debian packages: dosemu, reiserfsprogs. Any package in the “Optional” or “Extra” section can safely be omitted. Even standard packages concerned with networking can safely be removed.

Do *not* use the tasksel-option to install packages, effectively do not even use dselect, or just to remove unnecessary packages to make space on a small disk. In any other case just log in as root and do “apt-get install dosemu reiserfsprogs”, but do this after the FreeDos Installation, than come back here and install dosemu.

4.3. FreeDos Installation

FreeDos is not bundled with DosEmu – Licence Issues. Find here a copy of Herbert Xu's instructions to get it installed (taken from the dosemu README.Debian):

Initially dosemu is configured to use FreeDOS. Until a FreeDOS package can be made that meets requirements of the Debian distribution, you will have to download it separately. It can be fetched from

```
ftp://ftp.dosemu.org/dosemu/dosemu-freedos-bin.tgz
```

To install it:

```
cd /usr/lib
mkdir -p dosemu freedos
ln -s ../freedos dosemu
tar xzf /path/to/dosemu-freedos-bin.tgz dosemu/freedos
chown -R root.root freedos
chmod -R og=rX freedos rm dosemu/freedos
```

4.4. Install Li(d)os

You downloaded

1. The Linux-Kernel-Package
2. The init script
3. The dosemu launch script

Now copy it with the help of a floppy or by any other means to the partition where Li(d)os lives. “dosinit” and “dosspawn” are to be copied to the /usr/local/bin/ directory. They must be executable you can achieve this by:

```
chmod +x /usr/local/bin/dos*
```

Modify the file /etc/lilo.conf by finding a line of the following form:

```
# append=""
```

and replacing it by this one:

```
append="init=/usr/local/bin/dosinit rw"
```

To install the Kernel issue the command:

```
dpkg -i kernel-image-2.4.18-micro_10.00.Custom_i386.deb
```

Now that you have successfully installed Li(d)os, you can decide to go through /etc/dosemu/dosemu.conf, especially to tell dosemu if you have a mouse.

If you install Li(d)os on a very small system, you should either disable the unix shell completely in the dosinit script, or install a bourne-shell substitute with less cholesterol like ash (apt-get install ash, then ln -fs /bin/ash /bin/sh).

After that you can reboot and enjoy your new operating system.

5. Usage

When Li(d)os comes up, it ask's you annoying questions, which you can answer by simply pressing the "Enter" key. Then you are thrown into the FreeCommand shell – that's where you wanted to be.

If you want to use the unix shell which is running on the first virtual terminal, press **Ctrl+Alt+F1**, to go back to DOS use **Alt+F2**, or **Ctrl+Alt+F2**.

If you want a second (third, fourth, ...) DOS terminal, execute

```
unix dosspawn
```

from a DOS terminal.

Jump between the DOS terminals with **Ctrl+Alt+F n** , where n is the number of the terminal.

If you want to shut down the computer, just switch it off. If it makes you feel better go to the unix shell and execute:

```
exec init 0
```

do:

```
unix exec init 0
```

Then wait until the kernel says "Power down".

I know of no easy way to do this if you use the "very small computer" approach.

You have to check the filesystem manually every than and now if you want to maintain track of errors on the C: drive.

You can execute any installed unix command (shrug). Maybe you will have to specify environment variables and directory path's completely, and most likely programs will mess up your display. Anyway

```
unix commandline
```

will execute the *commandline* in a unix shell and spit the output to the current console. You can eventually clean up the terminal with

```
unix reset
```

You can spawn dos sessions on serial lines and use your old 286 with a terminal emulator as a second interface to your computer, but don't ask me how to accomplish this.

Be aware, that each dos terminal uses the same hard disk. So if you edit the same file in two terminals, call programs that create some fixed tempfiles, etc. you'll experiment strange things...

6. It is not so cool as I thought it would be

My standard laptop kernel package is about 1Mbyte larger than the size optimized Li(d)os kernel. However it is still 800kByte BIG! I remember with sadness the 160kByte kernels from my first Slackware Linux almost a decade ago. What the h... did they put into it nowadays?!

Also Dosemu is quite big, I saw a 16Meg Virtual Size and 2Meg Resident size on my machine, so memory consumption is much bigger than I would like it to be. Success stories from small machines are highly appreciated to relieve my grieve, as well as hints about:

- What's still inside that bloats the kernel up?
- Would recompiling Dosemu from source make it smaller? What switches have to be twiddled?

7. Notes

If the computer is fairly fast, a Reiser Filesystem would be the best choice as the root filesystem. I don't know how Reiser behaves on slow hardware - maybe we can start benchmarking it. Actually you cannot install Debian directly on a ReiserFS, look at other sources how to do it manually.

Dosemu needs the /proc filesystem. This is the only reason for the dosinit script: it calls the unix "mount" call to mount /proc. If you want to partition your harddisk in /usr and/or /var you'll have to mount them too there.

I chose to use ext3 (or reiser), because they would allow rebooting without a filesystem check, what's especially interesting if a cluster node goes up and down frequently. It's up to the user to schedule checks of the filesystem if not you'll get into problems with your data, whatever, you are a dos-user and used to this.

If a 8MByte Memory footprint cannot be hold with dosemu and Linux, a bigger kernel with more functionality would be surely a better shot than the actual minimalistic kernel.

It should be easy, to set up user separation and access control. when launching a new terminal it could be done with "su - user" or something like this. Dosemu provides a "Home-Drive" option.

Can somebody please throw in what is needed to shut up the two questions asked by FreeDos at startup.

8. Disclaimer

The software used and described in this document is Copyright by their authors, I deserve no Credits for it, I'm just another excited user - Congratulations folks!

Almost everything stated and explained in this document is in direct contradiccion to good practice, best knowledge, standard system administration and security.

I take no responsibility for damage resulting from following this instruccions or using the software described.

You are warned: do not apply this procedures on a computer where you have valuable data which you cannot afford to loose.

If your disk erases, your computer burns, your house falls down, it's your fault, not mine. If you complain with me, I'll throw your mail away, laught at you, or even can read it and give you a hint about what could have been the problem, if I am in mood to do so.

Once again: !!!Use this information and Software at your own risk!!!