

# POCKET MINI-DISTRIBUTION LINUX

CHRISTIAN PERLE



There are thousands of tools and utilities for Linux. Out of the Box chooses the pick of the bunch and suggests a little program each month that we feel is either absolutely indispensable or unduly ignored. To keep in line with the main focus of this issue we are bending our rule a bit this time round and devoting ourselves entirely to just one, entire distribution - but one which nevertheless runs from a single diskette: HAL91.



Mini-distributions, even if they are often largely ignored in the shadows of their bigger siblings, are useful tools. The spectrum ranges from the special distribution, which system administrators like to use

for diagnosis and correcting faults, to the almost-complete Linux desktop for older hardware (see the Other Mini-distributions box also).

HAL91 is specially conceived for somewhat older computers and is an ideal playground for all those who want to control their system without any graphical tools. This distribution can also be used as an emergency system, too, or general installation diskette (with a little manual work).

The requirements to run HAL91 turn out to be correspondingly modest. It can be started with any processor from the 80386 onwards, equipped with at least 8MB RAM. It was also important to the developers not to use the **higher density** diskette format, since a few drives have problems with this. HAL91 was developed by Øyvind Kolås and since January 2000 it has been undergoing refinement by myself (Christian Perle).

**Higher density:** Formatting a diskette with more than the usual 1.44MB capacity (about 1.72MB).

**Disk-Image:** The image of a complete diskette as a file. With suitable programs the diskette including **Bootsector** can be created from this.

**Bootsector:** The first sector on a diskette or another data medium which can contain executable code to start an operating system (boot).

**dd:** This Unix command serves for direct read/write of block-oriented devices. The data read can if necessary be converted into the format.

**BIOS:** Basic Input/Output System. This minimal system sits permanently in the computer and ensures it is possible to load an operating system from diskette or hard drive.

## What do you need?

On the HAL91 homepage <http://home.tu-clausthal.de/~incp/hal91/> you will find the **Disk-Image** `hal91.img`. If you do not yet have Linux on your computer and have to install from DOS/Windows, you will also find the DOS program `rawrite2.exe` there.

## How do you install it?

Since we are dealing with a ready-made diskette image, all we have to do is write this on a formatted diskette. If you already have Linux up and running, the following **dd** command (to be entered by the root administrator) is sufficient:

```
dd if=hal91.img of=/dev/fd0
```

Under DOS/Windows you should instead use `rawrite2.exe`:

```
rawrite2 -f hal91.img -d a:
```

## Start me up

To boot from the diskette, you may need to change the boot sequence in the **BIOS** setup of your computer to `A;C:`. With the HAL91 diskette in the drive, re-start your computer. After about 60 seconds, the mini-Linux is loaded and announces itself with a logo in the console (Figure 1). The diskette can then be taken out of the drive after booting because the system runs completely in the RAM.

As in the 'big' distributions there are also several virtual consoles available in HAL91 (reached via `ALT-F1` to `ALT-F4`). There is no need to log in, as a **Shell** with *root* privileges is running on all consoles.

**Shell:** One of the most important components of every Unix system - the command line controlled user interface of the system.

**Kernel:** The operating system acts as the interface between hardware and any processes running. It also makes multitasking and memory management available. The real Linux is only the kernel.

**ext2:** The Second Extended Filesystem is now the most commonly used file system under Linux. It provides files and directories with rights and assigns them to owners and groups.

**Manpage:** Manpages (short for Manual pages) are an online reference handbook for Unix commands. They are called up using `man` commands. There are no manpages included in HAL91 due to lack of space.

**Mount:** Under Unix systems, data media are not assigned drive letters, but mounted in the file system. A directory provided for this (the Mountpoint) serves for access to the content of the data medium.

**Shell script:** A text file with commands which are automatically processed in sequence by the shell.

**Nullmodem cable:** A cable to connect two computers directly via the serial interface. Unlike normal serial cables, in this case the send and receive lines are cross connected.

**PPP:** The Point to Point Protocol connects two computers via a serial line (modem or null modem) with the TCP/IP protocol.

### Listing 1: Extract from the kernel messages

```
hda: TOSHIBA MK1924FCV, 518MB w/128kB Cache, CHS=1053/16/63
ide0 at 0x1f0-0x1f7,0x3f6 on irq 14
Floppy drive(s): fd0 is 1.44M
FDC 0 is an 8272A
PPP: version 2.2.0 (dynamic channel allocation)
```

Especially with older computers, there is rarely anyone who actually remembers what hardware is actually inside them. HAL91 helps to identify many components. So IDE hard disks, ATAPI CD-ROM drives, NE2000-compatible network cards and

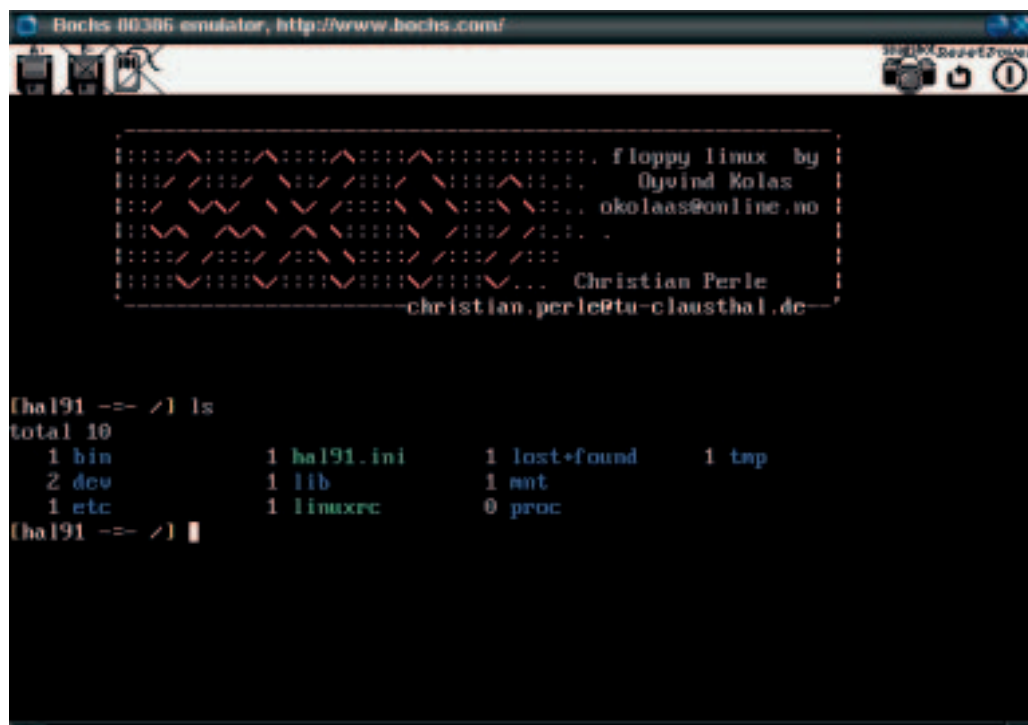


Figure 1:  
The HAL91 console

### Other Mini-distributions

Of course, there are other mini-Linuxes apart from HAL91. Three specialised examples are worth mentioning in this respect: *tomsrtbt* (Tom's Root Boot Disk) is especially good for use as a rescue system, while with *muLinux* an attempt is made to capture as many applications as possible on one diskette, and *fli4l* provides a complete ISDN router solution on a diskette. Reference sources are <http://www.toms.net/rb/home.html> for *tomsrtbt*, <http://sunsite.dk/mulinux/> for *muLinux* and <http://www.fli4l.de/> for *fli4l*.

diverse PCI devices can be recognised and used from time to time.

The commands *dmesg* to display the Kernel messages and *cat /proc/pci*, which lists the PCI devices. Listing 1 shows an extract from the kernel messages on a Toshiba 100CS Notebook.

These messages are telling us that an IDE hard drive (*hda*, thus the first device on the first IDE controller) which is a Toshiba with 518MB, split into 1053 cylinders, 16 write/read heads and 63 sectors per cylinder, has been found. The next line shows that only one IDE controller (*ide0*) is present. There is also a diskette drive (*fd0*), which is connected to the controller *FDC 0*. The last line of the extract relates the support that exists in the kernel for PPP.

### What else?

The command *e2fsck* for patching up the **ext2** file systems, is included, with which you might be able to resuscitate wounded Linux installations. You must, however, first read the associated **Manpage**.

So that you can also **mount** hard disk partitions, diskettes and CD-ROMs, the commands *mount* and *umount* are included. *ext2* (Linux), *vfat* (for long filenames from Windows 95/98) and *iso9660* (for data CDs) are all included.

### Small, but networked

HAL91 can also make contact with the world around it, to a limited extent. If the computer contains an NE2000-compatible network card, this can be configured using the commands *ifconfig* and *route*, to work with *telnet*, *ftp* or *ncp* on remote computers or to transfer files. The program *ncp* has already been discussed in an earlier Out of the Box article (Linux Magazine Issue 3 p.114). You can find

an example of using *ifconfig* and *route* in the **Shell script** *init.net* in the */bin* directory.

If there is no network card, you can still make use of the **Nullmodem cable**. Using the shell script *ppp-nullmodem* - which can also be found in the */bin* directory - make a **PPP** connection between two computers running under HAL91. This will involve having to swap the IP address used in the script on one of the two computers, so that the *pppd* commands look as follows:

```
pppd /dev/ttyS1 115200 asyncmap 0 noauth pers2
ist local passive nodefaultroute 192.168.0.12
:192.168.0.2
```

on one and

```
pppd /dev/ttyS1 115200 asyncmap 0 noauth pers2
ist local passive nodefaultroute 192.168.0.22
:192.168.0.1
```

on the other computer.

### An editor for all seasons

A text editor is also part of the package with HAL91. The Unix standard editor *vi* was quite deliberately chosen for this. This may not be as easy to learn as other editors, but it has the most functions in proportion to its small memory space requirement. A good introduction can be found at <http://www.infobound.com/vi.html>. Experiment with this editor, it's worth it!

From this point of view, HAL91 is also a suitable system for dry runs with Linux. So long as nothing else is mounted, everything runs in the RAM and is, unbreakable. Finally, it is also possible to find out with no risk what happens if *libc.so.5* is deleted.

### For the advanced student

Having once discovered this toy/tool, you may well want to make your own adaptations. HAL91 is based on kernel 2.0.36 and *libc 5.3.12*. To compile your own kernel for this distribution it is best to stay in line with the existing configuration, which is stored on the diskette as the file *kconf*. In any event, support for the *Initial Ramdisk (initrd)* should be part of the compilation.

To swap programs you will have to change the content of the compressed RAM disk (*initrd.gz*). To do this, copy the file onto a hard disk partition and uncompress with *gunzip*, in order to then mount it via the **Loop Device**.

Now you can potter about to your heart's content in the file, deleting or adding programs. When you do though, bear in mind that the programs can, at the most, be linked to *libc 5.x*, *libm 5.x* or *libtermcap*. Also, after unmounting and re-compression using *gzip -9* the file must not be too big for the diskette. That also sets the limits for HAL91: A graphical user interface with X11 thus falls victim to the requirements of space.

**Loop device:** With the loop device it is possible to mount files like partitions. A syntax example of this reads *mount -t ext2 -o loop initrd /mnt* - where the file *initrd* is mounted as *ext2* file system on the directory */mnt*.

**Router:** A router is used for passing on (forwarding) IP packets to specific destination IP addresses. From its Routing Table, it knows which route a packet should take, depending on the destination address.