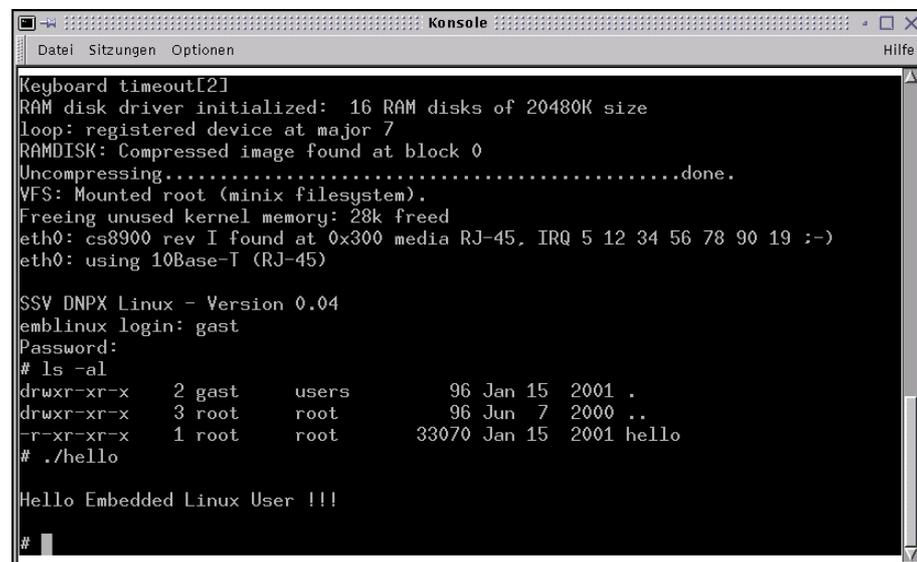


File insert into ROM-Disk

The embedded Linux of the DIL/NetPC is completely booted from a FLASH memory chip. Three components are stored in this FLASH: 1st the bootloader, 2nd the Linux- Kernel and 3rd the root file system, typically for Linux. These three elements can be available in form of only one binary image (for example DNPX.IMG) –or also as separate files under ROM-DOS. In this case it is possible to start the embedded Linux via DOS command line.



```
Keyboard timeout[2]
RAM disk driver initialized: 16 RAM disks of 20480K size
loop: registered device at major 7
RAMDISK: Compressed image found at block 0
Uncompressing.....done.
VFS: Mounted root (minix filesystem).
Freeing unused kernel memory: 28k freed
eth0: cs8900 rev I found at 0x300 media RJ-45, IRQ 5 12 34 56 78 90 19 :-)
eth0: using 10Base-T (RJ-45)

SSV DNPX Linux - Version 0.04
emblinux login: gast
Password:
# ls -al
drwxr-xr-x  2 gast  users      96 Jan 15  2001 .
drwxr-xr-x  3 root  root       96 Jun  7  2000 ..
-r-xr-xr-x  1 root  root    33070 Jan 15  2001 hello
# ./hello

Hello Embedded Linux User !!!
#
```

Figure 1: Start of a program in the directory `/home/visitor`

The three separate files use the names `LOADLIN.EXE` for the bootloader, `ZIMAGE` for the kernel and `RIMAGE.GZ` for the root file system. You will find these files on the DIL/NetPC-Starterkit CD-ROM.

The file `RIMAGE.GZ` is special. This file includes the complete file system for the embedded Linux in a compressed form. In other words: All files and directories, which are used during the runtime by the embedded Linux on the DIL/NetPC, are stored in this file. The figure 2 symbolizes this connection.

Starting the Linux system, the bootloader will always be activated. This bootloader puts the kernel into the main memory and starts it. The kernel expands itself and generates the RAM-disk. On this RAM-disk, the root file system will be generated. The file `RIMAGE.GZ` will be decompressed and copied into the empty RAM-disk.

In other words: After the boot-procedure the RAM-disk builds an exact copy of the files and directories in RIMAGE.GZ.

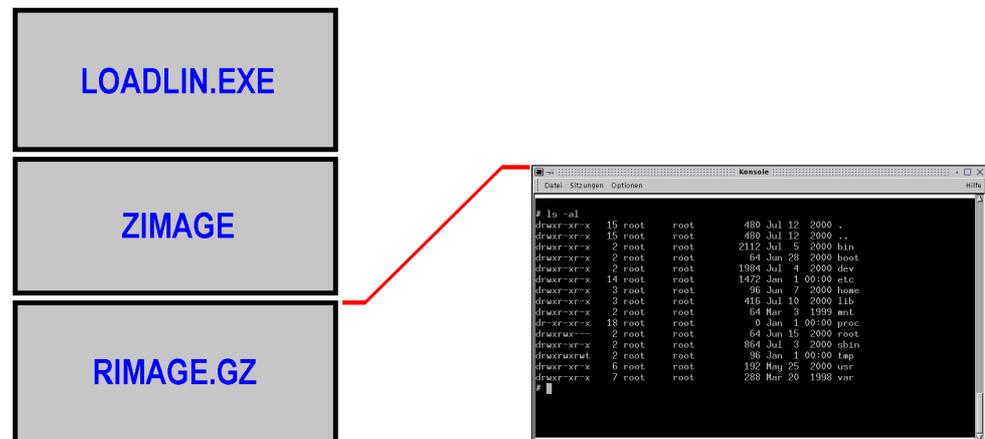


Figure 2: RIMAGE.GZ contains the complete root file system

If files will be created on the RAM-disk during runtime, these files get lost after every boot procedure. Should these files be available permanently on the DIL/NetPC –they have to be stored in RIMAGE.GZ.

A typical example is a program, which is written for the DIL/NetPC. During the software-development the executable (binary) file of this program can be transmitted via FTP onto the RAM-disk after every change for debugging and/or testing. There it will be stored and –if required, started (for example via Telnet). To the end of the development time, the program should be permanent available on the DIL/NetPC. For that, it is necessary to insert the binary file into RIMAGE.GZ.

Working on RIMAGE.GZ

The bootloader LOADLIN.EXE, the file ZIMAGE with the kernel and RIMAGE.GZ, as source for the root file system, were stored in the FLASH of the DIL/NetPC. It is not possible to change these files directly on the DIL/NetPC without further activities. As a result it is necessary to have the possibility to modify RIMAGE.GZ on a development system –if it is required. This development system must also run under Linux, so that the suitable tools and technologies are available. There are five steps necessary:

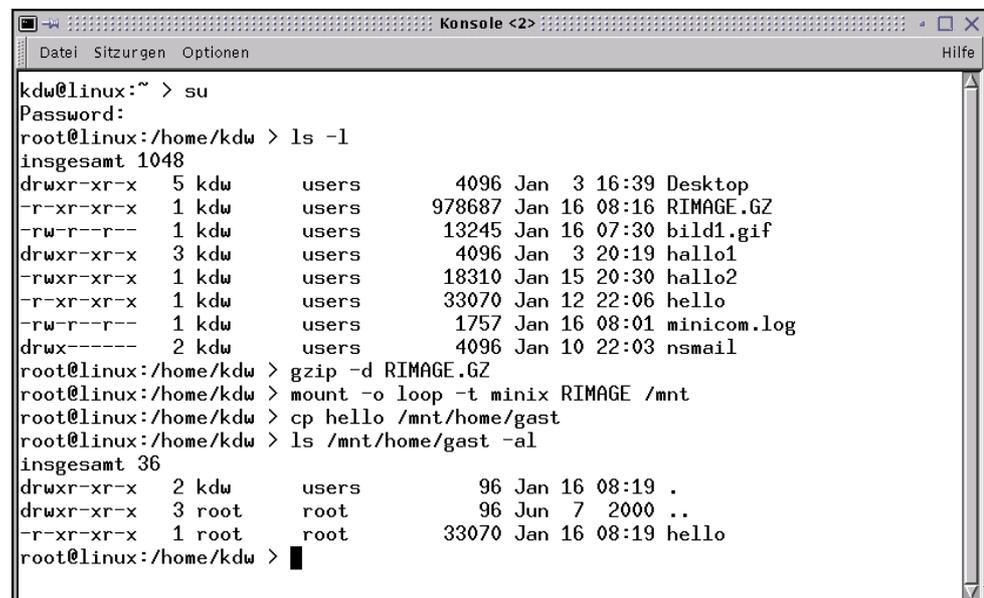
- 1.) Unpacking RIMAGE.GZ with the program gzip (decompress).
- 2.) "Mounting" the uncompressed file to allow access.
- 3.) Add or replace the requested file.

4.) RIMAGE "unmounten" again (do not forget this step).

5.) Generate a new file RIMAGE.GZ with the program gzip.

For example we insert a file in RIMAGE.GZ. This file should be started later on the DIL/NetPC as a program and show you the text "Hello embedded Linux User!!!" (see figure 1). The separate points will be described step by step and should serve as an example for own projects.

First of all, you have to guarantee that the file RIMAGE.GZ is available on your development system. You will find the original file on the Starterkit CD-ROM in the subdirectory /LINUX. Please copy the file HELLO from the CD-ROM-directory /SAMPLE/LINUX/HELLO to the development system. This file is an executable binary file for text output. If necessary ensure, by using the Linux-command `chmod +x hello`, that the file HELLO is equipped with the necessary rights (+x = Executable) for execution.



```

kdw@linux:~ > su
Password:
root@linux:/home/kdw > ls -l
insgesamt 1048
drwxr-xr-x  5 kdw  users      4096 Jan  3 16:39 Desktop
-r-xr-xr-x  1 kdw  users    978687 Jan 16 08:16 RIMAGE.GZ
-rw-r--r--  1 kdw  users    13245 Jan 16 07:30 bild1.gif
drwxr-xr-x  3 kdw  users      4096 Jan  3 20:19 hallo1
-rwxr-xr-x  1 kdw  users     18310 Jan 15 20:30 hallo2
-r-xr-xr-x  1 kdw  users     33070 Jan 12 22:06 hello
-rw-r--r--  1 kdw  users     1757 Jan 16 08:01 minicom.log
drwx----- 2 kdw  users      4096 Jan 10 22:03 nsmail
root@linux:/home/kdw > gzip -d RIMAGE.GZ
root@linux:/home/kdw > mount -o loop -t minix RIMAGE /mnt
root@linux:/home/kdw > cp hello /mnt/home/gast
root@linux:/home/kdw > ls /mnt/home/gast -al
insgesamt 36
drwxr-xr-x  2 kdw  users       96 Jan 16 08:19 .
drwxr-xr-x  3 root  root        96 Jun  7 2000 ..
-r-xr-xr-x  1 root  root     33070 Jan 16 08:19 hello
root@linux:/home/kdw >

```

Figure 3: Mounting from RIMAGE as directory /mnt

Now, please open a terminal window and log on to the development system under Linux. You have to own root rights –this means you have to log on as system administrator. Execute the following command from the subdirectory, which contains the files RIMAGE.GZ and HELLO:

```
gzip -d rimage.gz
```

The file RIMAGE.GZ will become uncompressed by the program gzip. The file RIMAGE.GZ itself will disappear within this process. For that, there is now a file RIMAGE with the approx. large of 3 Mbytes. This uncompressed file you can mount as a further file system (link up). Linux offers the mount-command with the appropriate parameters:

```
mount -or loop -t minix rimage /mnt
```

To execute the mount-command, it is necessary to have root-access. After a successful execution you decree a further directory named /mnt which you can access. Figure 3 shows some details about these steps.

If you are not familiar with the Linux file system, because you have only worked under Windows before, the mount procedure of RIMAGE may look a little confused to you. First of all, you have to clarify for yourself that a Linux file system does not know any disk drives. The entire file system basically exists only of subdirectories and files. In this case it does not matter, onto which disk drive any directory is located. The file system can be extended at any time by using the mount-command.

For this, a device with own file system, like a floppy disk or CD-ROM, will be simply added (mounted). After the mount procedure there is the added file system as an integral component of the entire Linux file system.

Under Linux you cannot only add a floppy or CD-ROM-drive by using the mount-command, but even a file which contains a file system. The file RIMAGE contains a file system, so the mount-command will append this file to the existing system. In other words: The DIL/NetPC file system stands now for disposal inside the development system as a directory /mnt.

All directories of the embedded Linux from a DIL/NetPC are now available within the /mnt directory.

Now you can copy any files into the directory /mnt. With the command:

```
cp hello /mnt/home/gast
```

For example, you can copy the file HELLO in the directory /home/visitor of the DIL/NetPC by the command:

```
ls /mnt/home/gast -al
```

You can view the content of the directory /home/visitor. At the same time you are able to check which access rights were attached to HELLO.

After all changes in the configuration files were made you have to disconnect /mnt, and as a result RIMAGE also, again from the file system (unmount). For that, it is necessary to use the following command:

```
umount /mnt
```

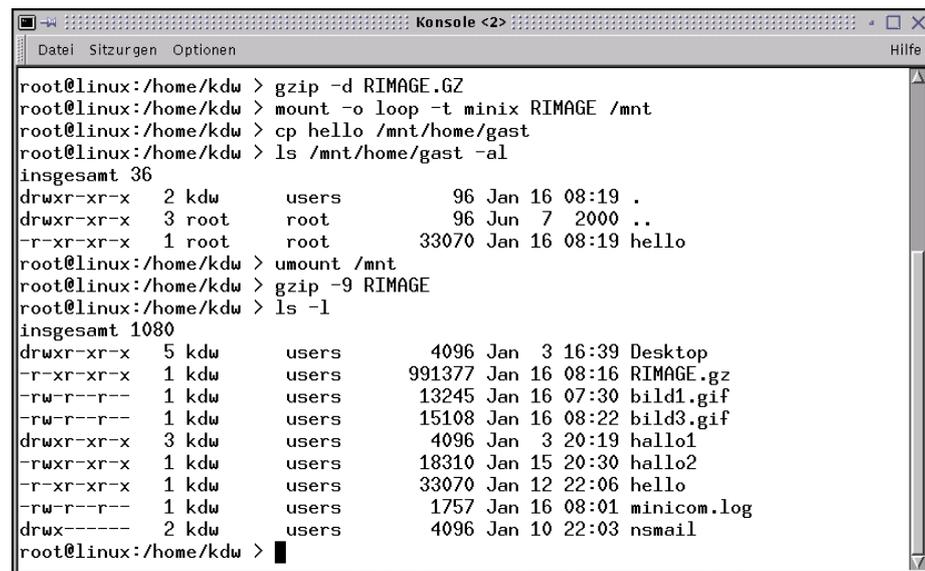
Figure 4 acts as an example. After that, RIMAGE is not coupled anymore onto the Linux file system of the development system. Now, by using the gzip-program a new compressed RIMAGE.GZ-file can be made. For that, the following command is necessary:

```
gzip -9 rimage
```

By the parameter "-9" the gzip-program is informed that it has to be compressed in according to the best possible "minimum file length".

Note: If you forget the unmount-instruction before you execute the gzip-command, the file system in RIMAGE.GZ will be damaged. This problem is not immediately noticeable. You will also not receive any error report. Only later on the DIL/NetPC it can result considerable problems.

Please execute the individual steps with the greatest possible care.



```

root@linux:/home/kdw > gzip -d RIMAGE.GZ
root@linux:/home/kdw > mount -o loop -t minix RIMAGE /mnt
root@linux:/home/kdw > cp hello /mnt/home/gast
root@linux:/home/kdw > ls /mnt/home/gast -al
insgesamt 36
drwxr-xr-x  2 kdw  users      96 Jan 16 08:19 .
drwxr-xr-x  3 root  root      96 Jun  7 2000 ..
-r-xr-xr-x  1 root  root    33070 Jan 16 08:19 hello
root@linux:/home/kdw > umount /mnt
root@linux:/home/kdw > gzip -9 RIMAGE
root@linux:/home/kdw > ls -l
insgesamt 1080
drwxr-xr-x  5 kdw  users    4096 Jan  3 16:39 Desktop
-r-xr-xr-x  1 kdw  users   991377 Jan 16 08:16 RIMAGE.gz
-rw-r--r--  1 kdw  users   13245 Jan 16 07:30 bild1.gif
-rw-r--r--  1 kdw  users   15108 Jan 16 08:22 bild3.gif
drwxr-xr-x  3 kdw  users    4096 Jan  3 20:19 hallo1
-rwxr-xr-x  1 kdw  users   18310 Jan 15 20:30 hallo2
-r-xr-xr-x  1 kdw  users   33070 Jan 12 22:06 hello
-rw-r--r--  1 kdw  users    1757 Jan 16 08:01 minicom.log
drwx----- 2 kdw  users    4096 Jan 10 22:03 nsmail
root@linux:/home/kdw >

```

Figure 4: Unmount and creating of a new file RIMAGE.GZ

Table 1 shows the individual steps again in a clear form. Through that, the configuration changes for a DIL/NetPC under embedded Linux should be practicable without problems.

Command	Function
<code>gzip -d rimage.gz</code>	RIMAGE.GZ uncompress
<code>mount -o loop -t minix rimage /mnt</code>	mount file system
<code>df /mnt</code>	show memory space
<code>cp hello /mnt/home/gast</code>	copy ...
<code>ls /mnt/home/gast -al</code>	test directory...
<code>umount /mnt</code>	unmount file system
<code>gzip -9 rimage</code>	compress RIMAGE.GZ

Table 1: Overview of the inputs

As long as you boot the DIL/NetPC embedded Linux via ROM-DOS, you are able

to transmit the new file RIMAGE.GZ by YMODEM-record via a terminal program to the DIL/NetPC. The figure 5 shows this step for `minicom`, a terminal program under Linux.

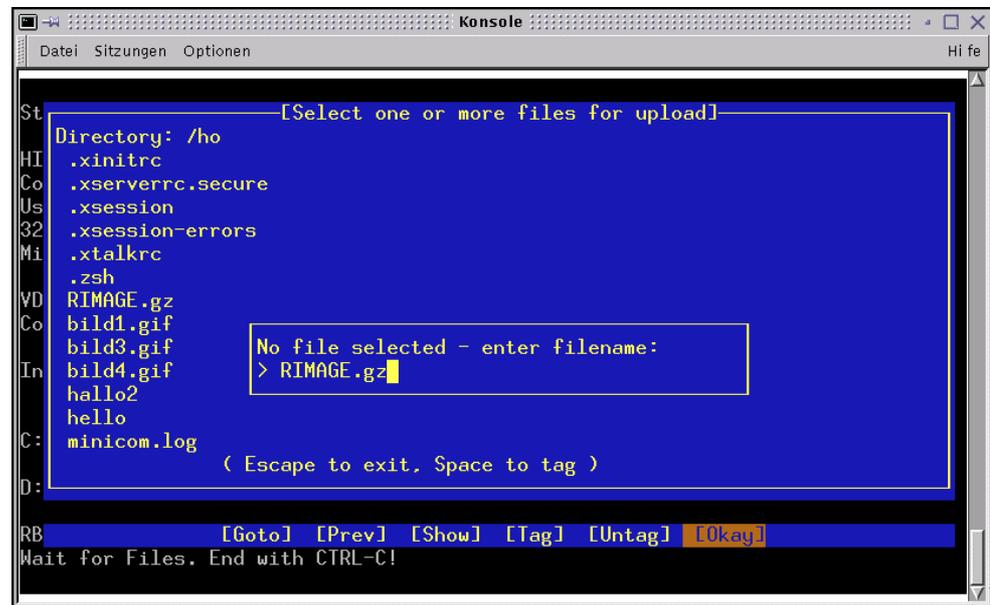


Figure 5: File transfer by `minicom`

For the DOS-free Linux-boot procedure first of all the new RIMAGE.GZ-file has to be transferred into DNPX.IMG. Then DNPX.IMG has to be generated and programmed completely new into the FLASH.

General Note

If you use Linux on your development system about a desktop like KDE, you should avoid absolutely, to copy the files to the subdir `/mnt` by using a file manager like KDF for example. For copying use definitely the Linux-`cp` (copy)-command as command line into a console.

We have found out, that in the case of use KDF for file copying and the following use of the `umount`-command the file system in RIMAGE.GZ is damaged. This mistake is noticeable in such way, that the file HELLO on the DIL/NetPC has only an approx. file length of 1.800 bytes. An execution of this file as program was also not possible.

Figures

- Figure 1: Start of a program in the directory `/home/visitor`
- Figure 2: RIMAGE.GZ contains the complete root file system
- Figure 3: Mounting from RIMAGE as directory `/mnt`
- Figure 4: Unmount and creating of a new file RIMAGE.GZ
- Figure 5: File transfer by `minicom`

Tables

Table 1: Overview of the inputs

Contact

SSV Embedded Systems
Heisterbergallee 72
D-30453 Hannover
Tel. +49-(0)511-40000-0
Fax. +49-(0)511-40000-40
E-mail: sales@ist1.de
Internet: www.ssv-embedded.de

Document History (Emblinx7e.doc)

Revision	Date		Name
1.00	28.08.2001	First Version.	JNE

This document is meant only for the internal application. The contents of this document can change any time without announcement. There is taken over no guarantee for the accuracy of the statements. Copyright © **SSV EMBEDDED SYSTEMS 2001**. All rights reserved.

INFORMATION PROVIDED IN THIS DOCUMENT IS PROVIDED 'AS IS' WITHOUT WARRANTY OF ANY KIND. The user assumes the entire risk as to the accuracy and the use of this document.