# Dr. Linux NEXT PATIENT PLEASE

BY MARIANNE WACHHOLZ

**Complicated organisms, since that's what Linux systems are, have little complaints all of their own. Dr. Linux observes the patients in the Linux newsgroups, issues prescriptions here for current problems and suggests alternative healing methods.**

## Is there room for just one more?

I have the feeling that the space on my computer is getting a bit tight. How do I find out how much hard disk space is occupied?

**Dr. Linux:** First of all, there are programs you can call using a command line, for example the command *df* (meaning *d*isk *f*ree):

```
user$ df
Filesystem   1k-blocks      Used   Available   Use%   Mounted on
/dev/hda6      3470648   2045904     1245220    62%   /
/dev/hda2       932912    637288      248236    72%   /RedHat
```

This promptly supplies you with the memory occupied by all **file systems** which you currently have **mounted**.

***mount:*** *This command integrates media (e.g. hard disk partitions and CDs) into the Linux file system. This is normally reserved for root. Before you can remove a mounted CD or diskette from the drive, an umount command is essential. And hard disk partitions, too, can be put out of reach under Linux again in this way.*

***File system:*** *The ways and means of organising data on a data carrier vary from one operating system to another and for different storage media. For example under Windows 9x there is usually an extension of the DOS file system FAT named VFAT in use, while Linux likes its data partitions to be in the ext2 file system. On data CDs on the other hand iso9660 is used.*

***Kilobyte:*** *Memory is divided into memory cells, which contain either the value 0 or 1. Such a memory cell or the data stored in it is called a Bit. Several bits can be combined into units such as a byte, word or long word: a byte for example corresponds to eight bits. This is sufficient to store one (Latin) letter. The word "space" accordingly needs five bytes of storage space. A page of text contains approx. 1500 characters and therefore unformated approx. 1500 bytes for storage. A kilobyte (kB, kByte) incidentally, corresponds not to 1000, but to 1024 bytes. One kilobyte times one kilobyte gives one Megabyte (MB, MByte), which is exactly 1,048,576 bytes. A Gigabyte (GByte), thus one kilobyte times one kilobyte times one kilobyte, is lots and lots of bytes, 1,073,741,824 to be exact.*

The memory space is displayed to you in one **kilobyte** units (*blocks*). You are given details on the whole space on the respective hard disk partition, diskette, CD, etc., with an indication of how much of it is occupied (*used*) or free (*available*); plus the memory space used as a percentage.

On checking these figures it will strike you that the used (plus the still-available) memory comes to only about 95 per cent of the total memory. This is not a program or calculation error, but a deliberate limitation of the memory space for normal mortal users. The last five per cent is only available for the superuser. This gives him or her the option of making space in a full disk by means of a pack program, which itself also needs some memory space to work.

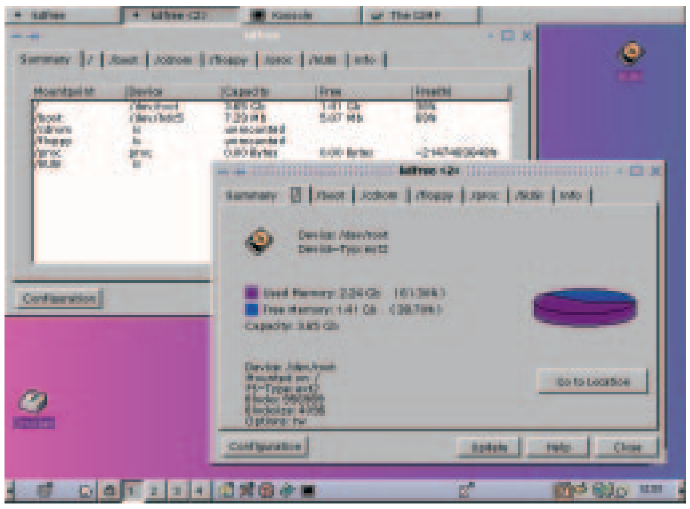
If you would prefer to have a specification in mega- or gigabytes, give *df* the option *-h* with:

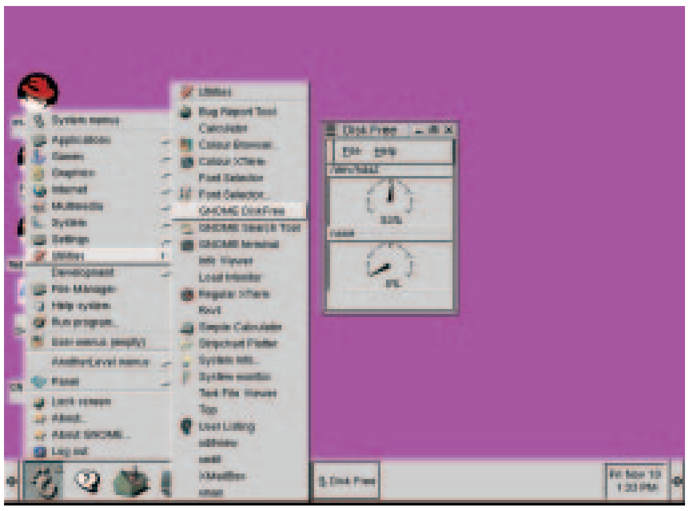
```
user$ df -h
Filesystem   Size  Used Avail Use% Mounted on
/dev/hda6    3.3G  2.0G  1.2G  62% /
/dev/hda2    911M  622M  242M  72% /Red Hat
```

For users who prefer to work with graphical interfaces, there are also a few applications available. In the KDE menu you might find, in the *Utilities* sub directory, the programs *KDFree* (Figure 1) and *KDu* (Figure 4).

If not, your distribution may supply these in a *kdu* package for later installation. With SuSE Linux this is in the series *kpa*. If you have no luck here you can download the appropriate package from *http://rpmfind.net/linux/RPM/kdu.html*.

*kdfree* offers in the first instance an overview of all data carriers which are entered in your ***/etc/fstab***. These are also selectable individually via riders, which provide a pie chart and information on the selected drive (Figure 1). As with *df* only those data carriers are taken into account which are mounted under a **mount point** in the system.





[left]
**Fig. 1: *kdfree* shows the memory allocation of individual drives as an overview or in detail**

[right]
**Fig. 2: The display of *gdiskfree* therefore looks a bit sparse**

In a direct comparison with this the program *GNOME Free Disk* (Figure 2), which you will find in the GNOME sub menu *Tools* looks a bit sparse in the graphic representation. Here round instruments show you a percentage value which represents how full your disk is. This tool can also be invoked from the command line using the command *gdiskfree*.

GNOME also offers you the option of inserting an applet into the control panel with which you can keep a constant eye on the memory space (Figure 3). To do this, in the GNOME menu, under *Panel/Add applet/Status display* select the item *Disk space*. On the command line, entering *diskusage_applet &* achieves the same result.

What is displayed are, as with *df,* on one command line the mount point (*MP:*) and after *av:* as in "*av*ailable" the available memory space in kilobyte blocks. If you click with the right mouse button on the applet icon in the panel, a few more setting can be configured via the menu entry *properties*; in particular, at this point the refresh rate of the display may be of interest.

## Memory space in directories

How can I find out how much memory space individual directories are occupying?

**Dr. Linux:** Change to a command line in the directory whose memory occupancy interests you, and enter the command:

```
user$ du -k
```

(short for: *d*isk *u*sage). You will receive an output of memory occupancy in kilobytes for the sub directories and the entire memory usage of a directory.

On the graphical user interface the aforementioned *KDu* in the sub directory *Utilities* of the KDE menu offers you the option of displaying the memory usage of directories (Figure 4).

## If you don’t ask...

My computer seems so slow to me – what’s the possible diagnosis?

**Dr. Linux:** Your Linux system naturally comes with a few programs which can be invaluable in helping you to analyse your system. I would like to introduce you to *vmstat* (*v*irtual *m*emory *stat*istics). Invoked on the command line, this produces something like the following output:

```
procs                    memory     swap        io        system            cpu
r  b w   swpd   free  buff  cache  si  so   bi   bo    in    cs   us   sy   id
0  0 0    212   4936  3104  23556   0   0    3    0   128   266    2    0   98
```

Here is how to decrypt this jumble of figures and abbreviations:

- *r* (*r*un): The higher the number you find here, the slower your system. This shows how many processes would run if you did not have to wait for machine time.
- *b* (*b*lock): Processes displayed here are waiting for specific events to be able to continue running the program, but not for machine time.
- *w*: The number at this point shows you the number of all processes currently backed up in the swap domain

***/etc/fstab:*** *This configuration file is used on system start up by the mount program to mount hard disk(partitions) and other data carriers in the file system and keep information ready for later mount actions. The first four columns in the file are the most interesting: The first specifies the device to be integrated, the second the mount point, the third the type of file system used on the data carrier. The mount options are listed in the fourth columns. The following example automatically mounts the root file system on the first partition of the first SCSI hard disk on booting, but on the other hand not the ATAPI-(IDE-)CDROM (noauto). However, later on, normal users (thanks to the option user) have the option, with the command mount /cdrom, of making any inserted data-CD accessible under /cdrom as read-only.*

```
#Devicename  mounted as   Filesystem  mount-options
/dev/sda1     /           ext2        defaults      0  0
[...]
/dev/hdc      /cdrom       iso9660     ro,noauto,user 0  0
```

***Mount point:*** *The directory in which the files of a data carrier are ”mounted” so that they can be accessed under Linux and other Unix operating systems.*

■

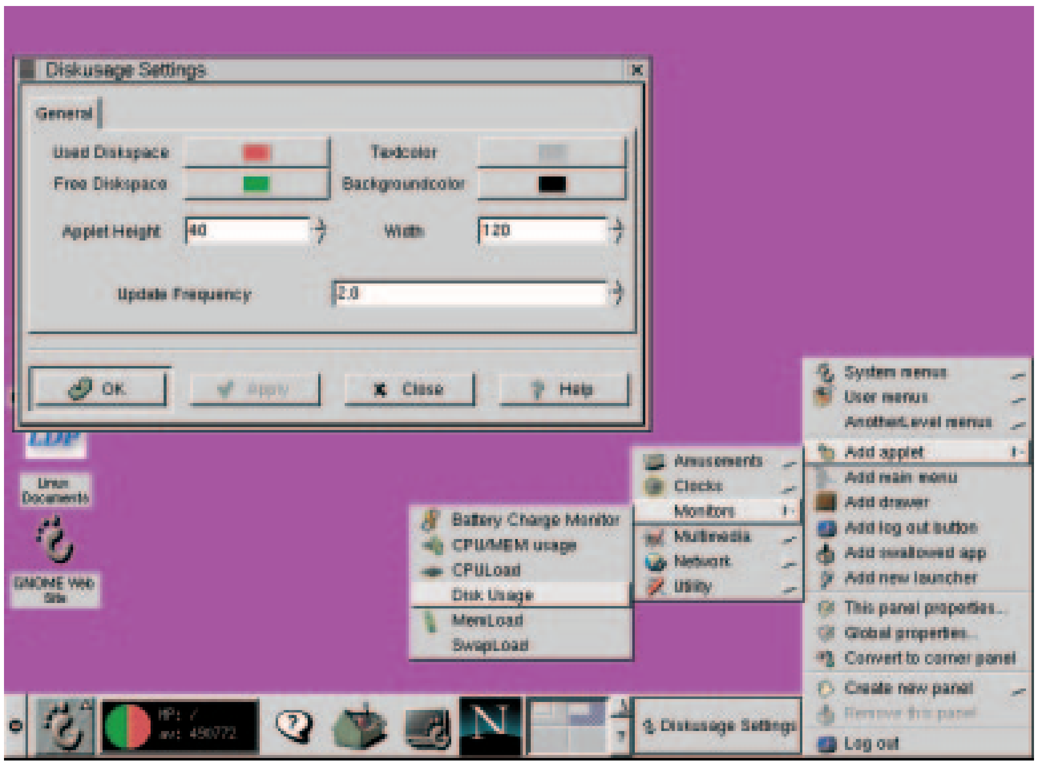
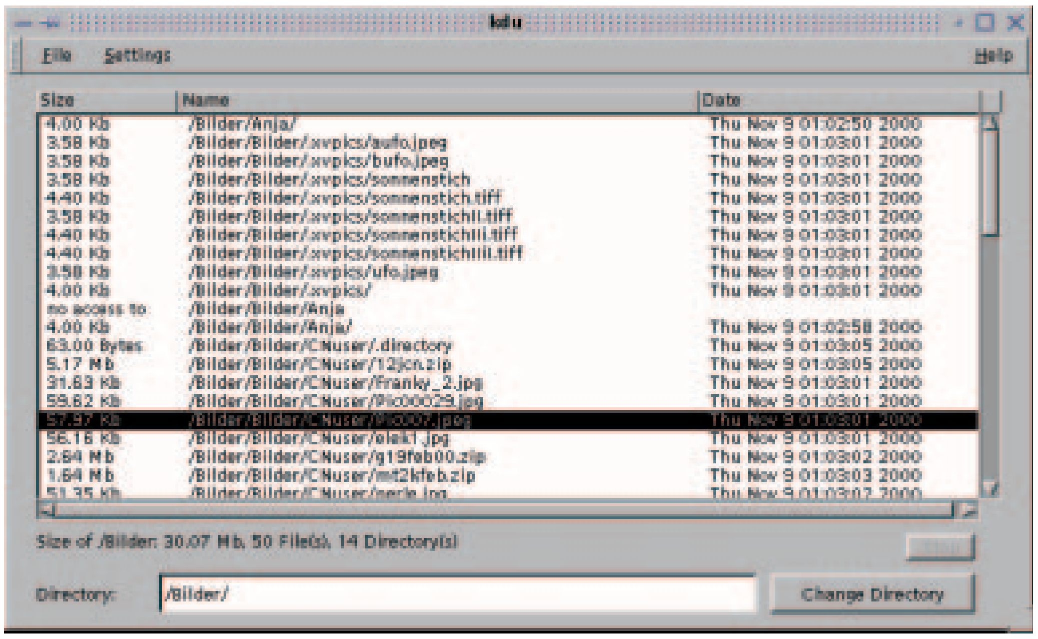


**Fig. 3: The applet *Disk space* and its configuration menu**



```
Size          Name                                          Date
4.00 Kb       /Bilder/Anja/                                 Thu Nov 9 01:02:50 2000
3.58 Kb       /Bilder/Bilder/xvpics/aufo.jpeg               Thu Nov 9 01:03:01 2000
3.58 Kb       /Bilder/Bilder/xvpics/bufo.jpeg               Thu Nov 9 01:03:01 2000
3.58 Kb       /Bilder/Bilder/xvpics/sonnenstich             Thu Nov 9 01:03:01 2000
4.40 Kb       /Bilder/Bilder/xvpics/sonnenstich.tiff        Thu Nov 9 01:03:01 2000
3.58 Kb       /Bilder/Bilder/xvpics/sonnenstichII.tiff      Thu Nov 9 01:03:01 2000
4.40 Kb       /Bilder/Bilder/xvpics/sonnenstichIII.tiff     Thu Nov 9 01:03:01 2000
4.40 Kb       /Bilder/Bilder/xvpics/sonnenstichIIII.tiff    Thu Nov 9 01:03:01 2000
3.58 Kb       /Bilder/Bilder/xvpics/ufo.jpeg                Thu Nov 9 01:03:01 2000
4.00 Kb       /Bilder/Bilder/xvpics/                        Thu Nov 9 01:03:01 2000
no access to  /Bilder/Bilder/Anja
4.00 Kb       /Bilder/Bilder/Anja/                          Thu Nov 9 01:02:58 2000
63.00 Bytes   /Bilder/Bilder/CNuser/.directory              Thu Nov 9 01:03:05 2000
5.17 Mb       /Bilder/Bilder/CNuser/12jcn.zip               Thu Nov 9 01:03:05 2000
31.83 Kb      /Bilder/Bilder/CNuser/Franky_2.jpg            Thu Nov 9 01:03:01 2000
59.62 Kb      /Bilder/Bilder/CNuser/Pic00029.jpg            Thu Nov 9 01:03:01 2000
57.97 Kb      /Bilder/Bilder/CNuser/Pic007.jpeg             Thu Nov 9 01:03:01 2000
56.16 Kb      /Bilder/Bilder/CNuser/elek1.jpg               Thu Nov 9 01:03:01 2000
2.64 Mb       /Bilder/Bilder/CNuser/g19feb00.zip            Thu Nov 9 01:03:02 2000
1.64 Mb       /Bilder/Bilder/CNuser/mt2kfeb.zip             Thu Nov 9 01:03:03 2000
51.35 Kb      /Bilder/Bilder/CNuser/nerle.jpg               Thu Nov 9 01:03:07 2000

Size of /Bilder: 30.07 Mb, 50 File(s), 14 Directory(s)

Directory: /Bilder/
```

**Fig. 4: KDu shows the memory space in directories**

- *swpd* states the swap memory currently in use in kilobytes
- *free* shows how many kilobytes of RAM are unused right now
- *buff* shows the size of the areas of memory in which the in-/output buffers are located; the unit is again the kilobyte. If data are being produced, there will not be a transfer to the hard disk for each symbol or each block. All in-/outputs are first placed in a buffer zone. If e.g. by reading, a program tries to access a block inside the file, the operating system checks whether the block sought is already in the buffer. If so, it is loaded and then made available to the program which requested it
- Under *cache* can be found in kilobytes how much integrated hard disk cache is currently available
- *si* (swap *in*) and *so* (swap out) show how many kilobytes of data per second have been loaded from the swap zone on the hard disk into the main memory or swapped from the RAM onto the disk
- *bi* (block *in*) and *bo* (block *out*) refers to e.g. hard disk activities, since the number of blocks per second is displayed here which have been sent to block devices or received from such. Block devices are for example floppies and hard disks.
- *in* shows the number of halts per second arising

from hardware demands; the so-called *Interrupts*

- *cs* (context switch) shows how often per second a switch is made from one program to the next (*multitasking*).
- *us* reflects what percentage of the processor time used is consumed by application programs, while
- *sy* shows the processor time used by the system
- *id* is the unused processor time, again as a percentage

An *id* number in the double-digit range and simultaneous massive activities in *si* and *so* can be the first indications of too little main memory. A high number for unused processor time means in such a case that the system often has to wait to access the hard disk. This can be confirmed, if *swpd* is high, *free,buff* and *cache* on the other hand stay relatively low. If *id, si* and *so* are constantly in the region of practically zero, these are serious indications of a processor which is too sluggish.

To track down problems, you should make *vmstat* produce periodic messages while these are going on. But you will probably have to be patient at this point if everything is moving as slowly as a tortoise in winter.

The following example shows a *vmstat* output with six repetitions (second figure) in one-second cycles (first figure):

```
user$ vmstat -n 1 6
  procs                    memory    swap        io     system        cpu
r  b  w   swpd   free   buff  cache  si  so    bi    bo   in    cs  us  sy  id
1  0  0    208   1556   2616  20736   0   0     3     0  130   265   2   0  97
2  0  0    208   1552   2616  20552   0   0   230     0 2019  2358  36  16  48
0  0  1    208   1544   2616  18868   0   0    91     0  830  2607  41  19  40
0  0  0    208   1544   2616  18868   0   0     0     0  102   208   0   1  99
1  0  0    208   1500   2616  18912   0   0    11     0  358   734   0   5  95
0  1  0    208   2336   2616  19188   0   0    69    60 1150   727  10   5  85
```

Anyone not wanting to go rummaging in the Man page, can use *vmstat x* to receive a brief introduction to the use of this handy tool.

## Space in the tiniest hut

When I ran up my Linux system I apparently created too small a swap partition. Can I upgrade the swap memory without repartitioning?

**Dr. Linux:** As a temporary solution, I would recommend a swap file – which you may be familiar with from Windows. Linux has two options for swapping data from RAM onto the hard disk:

- When installing your system you inevitably came across the swap partition. This is the first and the faster option
- The swap file is used less often. This works more slowly than a swap partition

Linux can manage up to 16 swap zones at once, so if necessary swap files can simply switch over

The making and activating of a swap file is done by the superuser on the command line. The former is done with the command:

```
root# dd if=/dev/zero of=Filename bs=1024 co↵
unt=Filesize_in_Kilobyte
```

This copies the file **/dev/zero** into the specified *file names*. As file size, enter a value between 40 and 131073 kilobytes. The specification of 131073 makes a file of around 128 megabytes, which is the maximum value for swap memory. The smallest manageable quantity of 40 kilobytes for making a swap area is not really sensible.

Using the command:

```
root# mkswap Filename Filesize_in_kilobytes
```

the file is given a swap identification. Only then can the file be used as swap storage. You can imagine this procedure as like the installation of a file system with the command *mkfs*.

Before you log the new file into the system, make sure that only the superuser has read-write privileges for it. Otherwise your system will questich the wrong privileges or give error messages and refuse access to the new swap area. In case of doubt use *chmod 0600 Filename* to determine the correct permissions.

Lastly, log the new memory with the command:

```
root# swapon  Filename
```

onto the system. The swap memory will be available to you immediately.

In Listing 1 you can see the acknowledgements produced by Linux when you create a swap file. Between the commands to create the file, the command *sync* is inserted at this point. This makes sure that everything is written to the hard disk, before the next command executes further actions with:

```
root# swapoff  Filename
```

you can log the additional swap memory off from the system, if you no longer need it. ■

***/dev/zero:*** *The content of the zero device, as the name indicates, consists of zeros (which go on forever). Everything written into this special file is deleted.*

■

**Listing 1: Creating and logging a swap file**

```
root@maxi:/ # dd if=/dev/zero of=swapfile bs=1024 count=131073
131073+0 Records on
131073+0 Records off
root@maxi:/ # sync
root@maxi:/ # mkswap swapfile 131073
Swap area Version 1 with a volume of 134213632 bytes is made.
root@maxi:/ # sync
root@maxi:/ # chmod 0600 swapfile
root@maxi:/ # swapon -v swapfile
swapon for swapfile
```