

Erstellen einer Boot-Disk:

Es werden beim 2.6.er min. 2 Disketten benötigt. Eine auf der der Kernel und der Boot-Loader (lilo) installiert ist und eine auf der ein minimales Root-System installiert ist. (Es geht auch beides auf einer Diskette ... aber hierbei gibt es deutliche Platzprobleme)

Diskette 1: (boot)

Kernel soweit wie möglich abspecken (floppy- Support nicht vergessen) `"/boot/bzImage-boot"`

Dateisystem auf die Diskette Bringen:

```
mke2fs -N 24 -m 0 /dev/fd0
```

→ Hierbei eine Gesamt-Inode-Anzahl von 24 und 0% reservierung für root

Diskette nach /mnt mounten:

```
mount -o dev /dev/fd0 /mnt
```

Nun die Wichtigen Ordner erstellen:

```
rm -rf /mnt/lost+found; mkdir /mnt/{boot,dev}
```

Nun den Kernel (wenn er passt) auf die Diskette Kopieren:

```
cp /boot/bzImage-boot /mnt/; sync
```

Jetzt die Lilo-Konfiguration:

```
vi /mnt/lilo.conf
```

```
boot      = /dev/fd0  
read-write  
backup    = /dev/null  
compact  
image     = bzImage-boot  
label     = Bootdiskette  
root      = /dev/fd0  
append    = "init=/bin/sh"
```

Nun noch einige Änderungen am Kernel:

```
rdev /mnt/bzImage-boot /dev/fd0 → Rootsystem des Kernels auf /dev/fd0  
rdev -r /mnt/bzImage-boot 49152 → Nach dem Laden des Kernels, vor dem mounten des Root-Dateisystems, soll eine Pause gemacht werden, um die Diskette wechseln zu können.
```

Jetzt den Lilo auf der Diskette Installieren:

```
lilo -v -C /mnt/lilo.conf -r /mnt
```

Diskette unmounten und Fertig:

Diskette 2. (Root)

Ein Dateisystem Image erstellen (ca 3,5 MB)

```
dd if=/dev/zero of=/root/root-fs.img bs=1K count=3500
```

Dateisystem in das Image:

```
mke2fs -m 0 -N 2000 /root/root-fs.img
```

Das Image nach /mnt mounten:

```
mount -o dev,loop -t ext2 /root/root-fs.img /mnt
```

Ordner-Struktur erstellen:

```
mkdir /mnt/{dev,proc,etc,sbin,bin,lib,mnt,usr}
```

Nun müßte jeder Befehl der Später verwendet werden soll, die Shell und die Benötigten Bibliotheken in dieses System Kopiert werden. Wegen der Platzprobleme entscheiden wir uns hier für Busybox.

```
cp /bin/busybox /mnt/bin
```

```
ldd /bin/busybox
```

```
linux-gate.so.1 => (0xffffe000)  
libcrypt.so.1 => /lib/tls/i686/cmov/libcrypt.so.1 (0xb7f6c000)  
libm.so.6 => /lib/tls/i686/cmov/libm.so.6 (0xb7f47000)  
libc.so.6 => /lib/tls/i686/cmov/libc.so.6 (0xb7e15000)  
/lib/ld-linux.so.2 (0xb7fae000)
```

```
mkdir -p /mnt/lib/tls/i686/cmov  
cp /lib/tls/i686/cmov/libcrypt.so.1 \  
    /mnt/lib/tls/i686/cmov/libcrypt.so.1  
cp /lib/tls/i686/cmov/libm.so.6 /mnt/lib/tls/i686/cmov/libm.so.6  
cp /lib/tls/i686/cmov/libc.so.6 /mnt/lib/tls/i686/cmov/libc.so.6  
cp /lib/ld-linux.so.2 /mnt/lib/ld-linux.so.2
```

Busybox beinhaltet alle nötigen Befehle, je nach dem über welchen Namen man busybox aufruft. Also müssen ein Paar Links erstellt werden.

```
cd /mnt/bin  
ln -s busybox sh
```

Nun noch die Nötigen Geräte- Dateien in das Mini-System Kopieren

```
cp -rvdp /dev/hd* /mnt/dev/  
cp -rvdp /dev/sd* /mnt/dev/  
cp -rvdp /dev/fd* /mnt/dev/  
cp -rvdp /dev/tty{1..6} /mnt/dev/  
cp -rvdp /dev/ram* /mnt/dev/  
cp -rvdp /dev/{console,kmem,mem,null} /mnt/dev
```

Das ganze Unmounten und dann das Image komprimieren:

```
umount /mnt  
dd if=/root/root-fs.img bs=1K | gzip -v9 > root-fs.img.gz
```

Das ganze nun mit dd auf die 2. Diskette schreiben:

```
dd if=/root/root-fs.img.gz of=/dev/fd0
```

FERTIG !!!!