Der aktuelle Preisverfall von Flash-Medien fordert es geradezu heraus, das Lieblings-Live-Linux tauglich für den Schlüsselbund zu machen. Wir erklären den Boot-Vorgang vom USB-Medium und zeigen die Konfiguration.

VON **MATTIAS SCHLENKER**

LINUX VON USB-STICK UND FESTPLATTE BOOTEN

Pocket-Pinguin

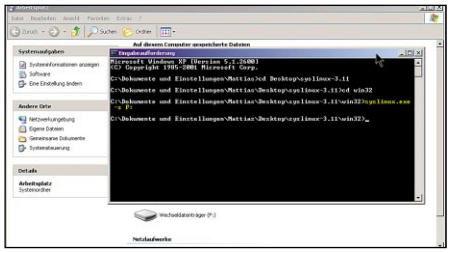
Seit etwa fünf Jahren ersetzen USB-Sticks nach und nach die gute alte Floppy. Mit heute kaum vorstellbaren Kapazitäten von 8 Megabyte hatten sie aus dem Stand weg fünffachen Speicher im kompakten Format und bargen ein deutlich geringeres Risiko, vom Bürostuhl überrollt zu werden. Doch in einem Punkt konnten USB-Sticks Floppies nicht ablösen: als Boot-Medium. Obwohl fast jedes BIOS das Starten vom USB-Wechseldatenträger anbietet, sind die beliebtesten Startmedien noch immer Floppy und CD. Beide sind nicht ideal. Die 1,44 Megabyte reichen gerade für ein DOS nebst BIOS-Update, und der optische Datenträger ist nicht nur empfindlich, sondern muss bereits bei kleinen Konfigurationsänderungen umständlich neu erstellt werden.

Dass der USB-Stick lange Zeit ein Schattendasein als Boot-Medium fristete, lag in erster Linie an zwei Faktoren: Ein DOS auf dem Stick ist höchstens für BIOS-Updates sinnvoll (Dell lieferte eine zeitlang sogar entsprechende Sticks mit seinen Rechnern aus), während andere Betriebssysteme häufig nur sehr aufwändig auf dem USB-Medium unterzubringen waren. Und gerade während der Anfangszeit der von USB bootfähigen Rechner vereitelten Fehler im BIOS häufig den Start.

Mit Linux und moderner Hardware treten Probleme deutlich seltener auf: Zum einen erlaubt es die offene Architektur, vielen Live-Distributionen derart angepasste Kernel und Ramdisks mitzugeben, dass der Stick auch wirklich vom System gefunden wird, und zum anderen kennen moderne BIOSe nicht nur USB-Floppy-Laufwerke, sondern auch USB-Zip oder USB-Festplatte und akzeptieren so häufig sogar in mehrere Partitionen unterteilte Sticks.

Stückchenweise

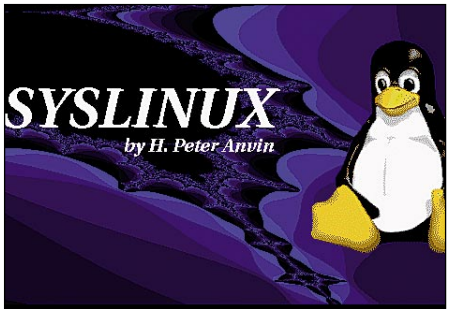
Der Boot-Vorgang von USB besteht aus mehr Schritten als bei anderen Medien: Zunächst wird vom BIOS entweder ein Floppy-Laufwerk oder eine am IDE-Bus angeschlossene Festplatte emuliert. Von diesem Laufwerk wird der in den ersten 446 Bytes befindliche Bootloader geladen. Bei DOS und dem von uns präferierten Bootloader *Syslinux* verweist er meist auf eine als „aktiv“ markierte Partition. Die wiederum enthält einen eigenen Bootloader oder einen Verweis auf eine zu startende Datei in der Dateisystems-Zuordnungstabelle. In unserem Fall ist das die Datei *syslinux.sys*, die mit dem Werkzeug *syslinux* den USB-Stick oder die USB-Festplatte geschrieben wurde. Syslinux lädt dann eine Konfiguration und kann sogar ein einfaches



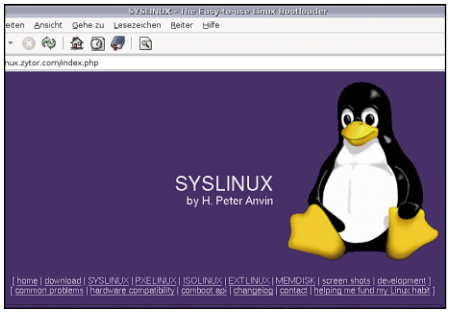
Vielseitig: mit der beiliegenden *syslinux.exe* können auch unter Windows bootfähige USB-Sticks erstellt werden.

Menü präsentieren. Die Vielzahl der Schritte und die Tatsache, dass so manch ein BIOS beim Start von USB an der längst überwunden geglaubten 1024-Zylinder-Grenze scheitert - die Boot-Datei muss innerhalb der ersten 1024-Zylinder der *Festplatte* liegen - macht Booten von USB schwieriger als von Festplatte.

Hat der Bootloader den Systemkern und die „initiale Ramdisk“ (*initrd*) in den Arbeitsspeicher geladen und gestartet, übernimmt der Kernel die Kontrolle. Da in fast allen Fällen auf ein komprimiertes Dateisystem auf dem



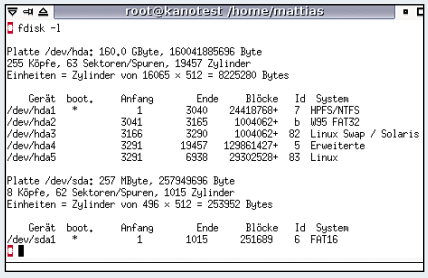
Splashscreens lassen sich aus indizierten (16 Farben) PPM-Dateien mit 640 Pixeln Breite erstellen. Dafür dient das Perl-Script *ppmtolss16*.



Unter http://syslinux.zytor.com/ finden Sie neben der aktuellsten Syslinux-Version umfangreiche Dokumentation.

In wenigen Schritten zum bootfähigen Stick

Das Mini-Linux *DamnSmallLinux* eignet sich hervorragend zur Installation auf einem USB-Stick, der zwischen 64 und 1024 Megabyte groß ist. Die folgenden Schritte können Sie entweder unter einer Live-CD oder einem auf Festplatte installierten Linux durchführen. Verwenden Sie auf jeden Fall das auf der Heft-DVD mitgelieferte *Syslinux 3.11* oder eine spätere Version. Die bei vielen Distributionen mitgelieferten 2er-Versionen erlauben keine Boot-Menüs oder sind sogar „kaputtgepatcht“



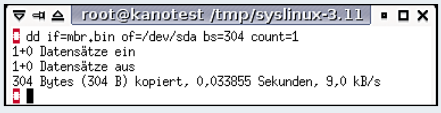
(z.B. bei der *Mandriva*-Distribution). Der Befehl *fdisk -l* zeigt die Partitionierung von allen angeschlossenen „blockorientierten“ Geräten. USB-Sticks und -Festplatten werden in der Regel als */dev/sda*, */dev/sdb* etc. oder (seltener) als */dev/uba* erkannt. Die erste Partition auf dem Stick muss vom Typ *FAT16* und als *bootfähig* (oder *aktiv*) markiert sein. Ist dies nicht der Fall, können Sie mit *fdisk /dev/sda* die Partitionierung des Sticks anpassen.

- **Formatieren Sie die Boot-Partition des Sticks. Am zuverlässigsten ist die Kommandozeile:**

```
mkfs.msdos -F 16 /dev/sda1
```

- **Entpacken Sie das Syslinux-Archiv und wechseln Sie in das entstandene Verzeichnis**

```
tar xvjf syslinux-3.11.tar.bz2
cd syslinux-3.11
```



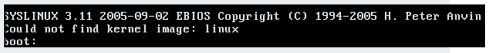
Nicht jeder Stick verfügt ab Werk über einen Master-Boot-Record (MBR). Sie können den bei Syslinux mitgelieferten mit dd kopieren oder ihn mit dem Tool ms-sys schreiben:

```
dd if=mbr.bin of=/dev/sda bs=304
count=1
```

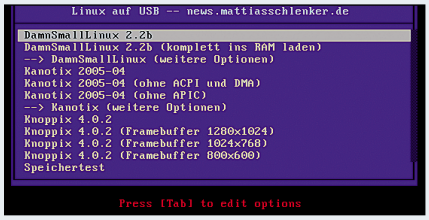
- **Schreiben Sie nun den eigentlichen Syslinux-Bootloader auf die aktive (im Idealfall die einzige) Partition des Sticks. Der Punkt leitet die relative Pfadangabe ein, schließlich befinden Sie sich im Verzeichnis syslinux-3.11 und damit außerhalb des Standardsuchpfades:**

```
./unix/syslinux -s /dev/sda1
```

- **Die Vorbereitung des Sticks ist jetzt abgeschlossen und der Bootloader vollständig geschrieben, aber noch nicht konfiguriert. Es gilt nun, das BIOS so einzustellen, dass von USB gestartet werden kann. Bei vielen Systemen muss dafür *USB Keyboard Support* oder *USB Legacy Support* aktiviert werden. USB-Sticks werden in der Regel als *USB Floppy* oder *USB Zip* angesprochen. Fehlen die USB-Geräte bei der Boot-Reihenfolge im BIOS-Setup, gelingt es meist mit *F8* oder *F11* während des Speichertests ein temporäres Boot-Medium auszuwählen. Wenigstens „SYSLINUX 3-11“ und die Fehlermeldung *Could not find kernel image* sollte den Erfolg anzeigen. Syslinux lädt und möchte nur noch konfiguriert werden.**



- **Die Verwandtschaft zum CD-Bootloader Isolinux erleichtert die Konfiguration deutlich. Kopieren Sie den Inhalt des Ordners /boot/isolinux der DamnSmallLinux-CD in das Wurzelverzeichnis des USB-Sticks und benennen Sie die Datei *isolinux.cfg* um in *syslinux.cfg*. Auch der KNOPPIX-Ordner muss auf den Stick. Die Datei isolinux.bin wird auf dem Stick nicht benötigt, Sie können diese löschen. Beim nächsten Booten lädt Syslinux den DamnSmallLinux-Splashscreen genau wie das CD-Pendant und bootet den Kernel auf simplen Druck der Eingabetaste. Da die *Initrd* (bei DamnSmall: *minirt24*) Treiber für USB 1.1 und 2.0 enthält und nach USB-Laufwerken sucht, wenn keine CD einliegt, ist keine weitere Anpassung notwendig. Ebenso mit *Kanotix*: wir mussten lediglich die Grub-Konfigurationsdatei *menu.lst* in das Format von *Syslinux* umschreiben, damit Kanotix vom 1GByte großen Stick startete.**



Übersichtliche und einfach zu nutzende Boot-Menüs sind schnell erstellt.

Auswahlmenüs mit Syslinux

Syslinux bringt ein eigenes Konfigurationsschema für Auswahlmenüs mit. Es handelt sich dabei um einfache Menüs, in denen Sie mit den Pfeiltasten navigieren können. Untermenüs sind problemlos möglich. Die Menüs werden von der Komponente *menu.c32* bereitgestellt, die Sie im Ordner *com32/modules* finden - kopieren Sie diese Datei einfach in das Wurzelverzeichnis des Sticks. Als Basis für die Konfiguration kann eine normale *syslinux.cfg* dienen. Der Trick liegt einzig und allein darin, als Standard nicht einen Linux-Kernel, sondern *menu.c32* zu laden. *menu.c32* liest die Konfigurationsdatei erneut und wertet die von der ersten Stufe ignorierten Zeilen *MENU LABEL* aus.

Im Fall von DamnSmallLinux sieht die originale Konfigurationsdatei (verkürzt wiedergegeben) so aus:

```
DEFAULT linux24
APPEND init=/etc/init lang=us vga=791
initrd=minirt24.gz ...

TIMEOUT 300
PROMPT 1
DISPLAY boot.msg

LABEL dsl
   KERNEL linux24
   APPEND init=/etc/init lang=us
vga=791 initrd=minirt24.gz ...

LABEL memtest
   KERNEL memtest
   APPEND initrd=
```

Statt des Kernels wird *menu.c32* als Default-Eintrag geladen, das Menü bekommt einen Titel, und die einzelnen Einträge werden um aussagekräftige Namen ergänzt:

```
DEFAULT menu.c32

TIMEOUT 300
PROMPT 1
DISPLAY boot.msg

MENU TITLE DamnSmallLinux

LABEL dsl
MENU LABEL DamnSmallLinux (Standard)
   KERNEL linux24
   APPEND init=/etc/init lang=us
vga=791 initrd=minirt24.gz ...

LABEL memtest
MENU LABEL Speichertest
   KERNEL memtest
   APPEND initrd=
```

Enthält die *syslinux.cfg* zu viele Einträge, ist es sinnvoll, selten Benötigte zu entfernen und in Untermenüs auszulagern. Die Übergabe eines APPEND-Paramters weist *menu.c32* an, eine bestimmte Konfigurationsdatei (hier *menu2.cfg*) zu laden:

```
LABEL menu2
MENU LABEL => Menue selten genutzte
Boot-Optionen
KERNEL menu.c32
APPEND menu2.cfg
```

Alternativ können Einträge in der *syslinux.cfg* vorhanden, aber versteckt sein. Diese sind am Bootprompt zugänglich, tauchen aber nicht im Auswahlmenü auf:

```
LABEL memtest
MENU HIDE
KERNEL memtest
APPEND initrd=
```

Stick zugegriffen wird, müssen Kernel und Ramdisk zuerst einen Treiber für den USB-Massenspeicher laden. Systeme aus Redmond scheitern meist an dieser Stelle und auch die meisten Live-Linuxe enthielten bis vor etwa zwei Jahren nicht die nötigen Treiber, was Nacharbeiten an der Initrd erforderte.

Welchen Bootloader?

Während beim Desktop-Linux längst Grub den Kampf der Bootloader für sich entscheiden konnte - der unterlegene *Lilo* ist praktisch nur noch auf Servern zu finden - dominiert bei USB-Laufwerken der einst für Disketten entwickelte *Syslinux*. *Syslinux* ist zwar unkompliziert in der Handhabung, durch die Beschränkung auf eine FAT16-Partition jedoch de facto auf Sticks mit maximal 1024 Megabyte Speicher limitiert. Größere Sticks oder USB-Festplatten erfordern deshalb die Unterteilung in mehrere Partitionen. Bei USB-Sticks ist diese Partitionierung nicht ganz unproblematisch: Einige Betriebssysteme aus Redmond erkennen nur die erste Partition. Legt man zudem eine kleine Boot-Partition am Ende des Sticks an, wird sich dieser nur noch auf Rechnern starten lassen, die USB-Sticks als Festplatte oder ZIP ansprechen. Rechner, die Sticks als Floppy erkennen, verlangen oft die Boot-Partition am Anfang des Datenträgers. Bei USB-Festplatten ist eine Partitionierung unproblematisch. Wir verwenden in der Regel eine 64 Megabyte große Boot-Partition am Anfang und eine FAT32-Partition, die den Rest der Platte überspannt. Formatiert man Letztere unter Linux, existiert keine Beschränkung auf 32 Gigabyte große Dateisysteme. Nun könnte man auf die Idee kommen, den Stick einfach mit dem „Grand Unified Bootloader“ (*Grub*) zu versehen, der auch von FAT32-Medien startet. Das haben wir ausprobiert und den Stick zunächst gemountet und eine Ordnerstruktur angelegt:

```
mount /media/sda1
mkdir -p /media/sda1/boot/grub
```

Weil das BIOS beim Booten den Stick als erste Festplatte anspricht, muss die Datei *device.map* von Hand erstellt werden:

```
echo '(hd0)  /dev/sda' >
/media/sda1/boot/grub/device.map
```

Jetzt kann der eigentliche Bootloader geschrieben werden:

```
grub-install --root-directory=/
➥ media/sda1 /dev/sda
```

Auf unseren Testsystemen bootete der so präparierte Stick nur dann sicher, wenn er im BIOS als Festplatte angesprochen werden konnte. Als USB-ZIP startete er vereinzelt und als USB-Floppy entweder überhaupt nicht, oder ein auf Festplatte befindlicher *Grub* wurde gestartet.

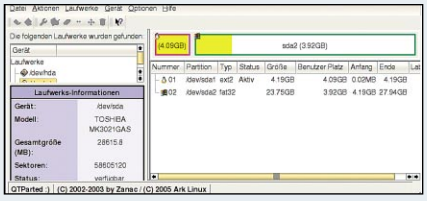
Der Grund dürfte darin liegen, dass *Grub* in diesem Fall den Stick als Floppy-Laufwerk annimmt und nach einer weiteren Festplatte sucht. *Syslinux* dagegen sucht seine Konfigurationsdateien, Kernel und Ramdisk auf dem Medium, von dem der Bootloader gestartet wurde - ganz gleich ob es sich dabei um eine emulierte Floppy oder eine Festplatte handelt.

DOS immer dabei

Kommt *Syslinux* zum Einsatz, kann mittels *memdisk* ein DOS-Disketten-Image in den Speicher geladen und gestartet werden. Das ermöglicht BIOS-Updates von Rechnern, die weder über ein Diskettenlaufwerk noch über ein installiertes Windows zum „Flashen“ ver-

Knoppix bootfähig auf USB-Festplatte

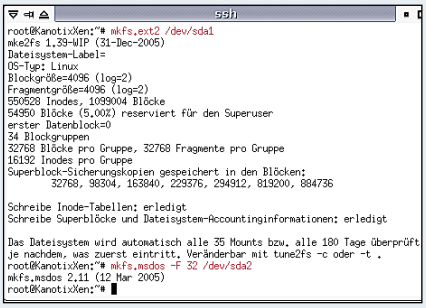
Nicht jede Distribution spielt mit: Obwohl sich FAT16 für die Syslinux-Bootpartition und FAT32 für die Partition mit den Cloop-Container-Dateien weitgehend als Standard etabliert haben, schert manch ein Distributor aus. Diesmal ist es Knoppix, das in Version 5.0 ohne FAT-Treiber seine Containerdateien auf entsprechend formatierten Medien nicht findet. Damit keine umständliche Modifikation der *initrd* notwendig ist, haben wir uns für einen kleinen Trick entschieden, um Knoppix bootfähig auf USB-Festplatte unterzubringen: Eine Linux-Partition, die mit *ext2* formatiert ist, nimmt neben dem kompletten Inhalt der Knoppix-DVD den mit Syslinux verwandten Bootloader *Extlinux* auf.
Und so bringen Sie Knoppix 5.0 auf USB-Festplatte unter:



Löschen Sie alle vorhandenen Partitionen und legen Sie, beispielsweise mit *qtparted* oder dem Befehl

- `fdisk /dev/sda`

eine etwa vier Gigabyte große Linux-Partition und eine FAT32-Partition, die den Rest der Platte überspannt, an. Markieren Sie die Linux-Partition als aktiv.
Formatieren Sie die neu angelegten Partitionen



- `mkfs.ext2 /dev/sda1`

`mkfs.msdos -F 32 /dev/sda2`

- **Kopieren Sie mit dd wie oben beschrieben den Bootsektor in den MBR der USB-Festplatte. Sie müssen sich dafür in der Shell im entpackten Syslinux-Ordner befinden:**

- `dd if=mbr.bin of=/dev/sda bs=304 count=1`

- **Mounten Sie die erste Partition der USB-Festplatte**

- `mount /dev/sda1 /media/sda1`

und kopieren Sie den gesamten Inhalt einer Knoppix-DVD auf die erste Partition der USB-Platte:

- `rsync -avP /cdrom/ /media/sda1/`

Verwenden Sie jetzt das ebenfalls im Syslinux-Tarball enthaltene Programm *extlinux* um den Bootloader zu schreiben:

- `cd /tmp/syslinux-3.11`

`./extlinux/extlinux`
`/media/sda1/boot/isolinux`

- **Benennen Sie Knoppix' Bootloader-Konfigurationsdatei**

- `/boot/isolinux/isolinux.cfg`

um in

- `/boot/isolinux/extlinux.conf`

Beim Booten von USB begrüßt Sie Extlinux statt Isolinux (CD/DVD) oder Syslinux (FAT16). Die Konfiguration von Extlinux können Sie genauso vornehmen wie bei Syslinux, allerdings müssen die Konfigurationsdateien in dem Verzeichnis liegen, das Sie bei der Installation des Bootloaders übergeben haben -- in unserem Beispiel also in /boot/isolinux.

fügen. Das Image einer DOS-Diskette entsteht auf einem Rechner mit Diskettenlaufwerk mit dem Befehl:

- `dd if=/dev/fd0 of=/tmp/dos.img`

Die entstandene Datei kopieren Sie neben der Datei *memdisk* (im gleichnamigen Ordner des Syslinux-Archivs) auf den Stick. Mit dem Eintrag

- `LABEL dos`

```
MENU LABEL DOS Floppy Image
KERNEL memdisk
APPEND initrd=dos.img
```

in der Syslinux-Konfigurationsdatei entsteht eine Startmöglichkeit für DOS. Die Image-Datei kann mittels Loopback-Treiber gemountet werden, was das Kopieren von BIOS-Images und Flash-Werkzeugen deutlich vereinfacht:

```
mount -o loop /tmp/dos.fdd /mnt/floppy
```

Für den Fall, dass Sie über kein Windows verfügen, mit dem Sie DOS-Disketten erstellen können, haben wir ein Image einer FreeDOS-Diskette auf der Heft-DVD beigelegt. Nach unserer Erfahrung arbeitet dieses gut mit Flash-Werkzeugen von Asrock und Via zusammen. Systemadministratoren erlaubt es Syslinux auf diese Weise, auf einem USB-Stick DOS-Images mit den BIOS-Updates verschiedener Rechner unterzubringen und per Bootmenü das richtige auszuwählen - das ist eine sinnvolle Alternative zum Stapel der sonst mitgeschleifter Disketten.

Weitere Informationen im Web

http://syslinux.zytor.com/
Eine ausführliche Dokumentation von Syslinux und den verwandten Bootloadern für CDs, EXT2-Partitionen und PXE-Boot hält die Projektseite bereit.

www.kernel.org/pub/linux/utils/boot/syslinux/
Die jeweils aktuellste Syslinux-Version liegt auf den Mirror-Servern des Linux-Kernels.

Fazit

Noch immer ist das Booten vom USB-Stick nicht ohne Tücken. Auf der sicheren Seite ist man mit USB-Sticks bis zu einem Gigabyte, die mit einer einzigen FAT16-Partition und Syslinux sowohl als emulierte Floppy oder als vorgetäuschte Festplatte funktionieren und so auf fast allen Rechnern starten. Fast genauso gut funktionieren USB-Festplatten, wenn sie mit einer kleinen FAT16-Boot-Partition (maximal ein Gigabyte) am Anfang versehen werden. Tückisch sind USB-Sticks mit mehr als einem Gigabyte. Partitioniert man diese, treten häufig Zugriffsprobleme unter Windows auf. Trotz vereinzelter Probleme sollten Sie sich nicht entmutigen lassen: Ein DamnSmallLinux oder Kanotix auf USB-Stick am Schlüsselbund oder das Knoppix in DVD-Version hat gute Dienste als Rettungssystem oder „Büro unterwegs“ geleistet. **:JKN**