Linux – Kernel neu konfigurieren und installieren

Klaus Knopper <kernel@knopper.net>

Der **Kernel** ist die Schnittstelle zwischen Hardware und Anwenderprogramm unter Unix. Hierin befinden sich die meisten **Gerätetreiber**, die **virtuelle Speicherverwaltung**, der **Prozess-Scheduler**, die **Dateisysteme** und vieles andere mehr. Benutzerprozesse haben ausschließlich über den Kernel Zugriff auf Systemresourcen wie am Rechner angeschlossene Peripherie.

Unter **Linux** ist der Systemkern im Quelltext verfügbar, und kann vom Anwender verändert, neu konfiguriert, optimiert und (vom Administrator) installiert werden. Viele Bestandteile des Kernels, die nicht permanent gebraucht werden, können als zur Laufzeit nachladbare **Module** gebaut werden, und werden vom Kernel-Modullader bei Bedarf initialisiert oder entfernt. Bereits beim Systemstart benötigte Module wie Festplattencontroller-Treiber (IDE oder SCSI) sowie das Dateisystem der Root-Partition (meist ext2) sollten jedoch unbedingt fest in den Kernel eincompiliert sein.

Die Kernel-Versionsnummer setzt sich immer aus drei Zahlen zusammen:

1. Die Haupt-Versionsnummer.
2. Die Releasenummer. Ist diese geradzahlig, so handelt es sich um ein stabiles Kernelrelease. Ist sie ungeradzahlig, so handelt es sich um einen experimentellen Entwicklerkernel, in dem zwar viele neue Features gegenüber der stabilen Version implementiert sind, jedoch noch keine ausreichenden Tests zur Verifikation der Stabilität stattgefunden haben. Für den Produktionseinsatz sollten stets die stabilen Kernel (gerade Releasenummer) eingesetzt werden.
3. Der Patchlevel. In unregelmäßigen Abständen gibt es sowohl für die stabilen Anwenderkernel als auch für die Entwicklerkernel Updates, die neue Funktionalitäten oder Fehlerbereinigungen enthalten.

Die folgende Tabelle listet die Schritte auf, die zur Neukonfiguration und Installation des Linux-Kernels erforderlich sind. Die meisten sind durch das Haupt-**Makefile** des Kernel-Sourcebaums automatisiert, einige jedoch erfordern manuellen Eingriff, da z.B. das Bootmenü nach den Wünschen des Anwenders konfiguriert wird.

1.	Aktuelle Kernel-Quelltexte (**tar**-Archiv, **rpm**-Paket o.ä.) beschaffen und in das Verzeichnis `/usr/src/linux` entpacken.
2.	In das Verzeichnis `/usr/src/linux` wechseln.
3.	Kernel-Konfigurationsmenü aufrufen mit `make menuconfig` (textorientiert) oder `make xconfig` (graphisch).
4.	Einstellen der gewünschten Konfigurationen.
5.	Abhängigkeiten generieren mit `make depend`.
6.	Komprimiertes Kernelimage generieren und Module übersetzen mit `make bzImage modules`.
7.	Module installieren mit `make modules_install`.
8.	Kernel in das Verzeichnis `/boot` kopieren, z.B. `cp arch/i386/boot/bzImage /boot/vmlinuz-neu`.
9.	Eintrag in `/etc/lilo.conf` z.B. mit `image=/boot/vmlinuz-neu` und `label=linux-neu`.
10.	`lilo` aufrufen, um den Master-Boot-Record neu zu schreiben.
11.	Reboot
12.	Falls das neue Kernel-Image nicht als Default in `/etc/lilo.conf` eingetragen wurde, muss es beim LILO-Bootprompt angegeben werden (hier: `linux-neu`).