Kernelupgrade (nicht Genkernel)

Installation eines neuen Kernel's

Hat man „sys-kernel/gentoo-sources“ mit dem Useflag „symlink“ gebaut (empfohlen), ist nach einem Systemupdate in diesem Schritt ja auch die Sources aktualisiert werden schon der Link „usr/src/linux“ auf die neue Kernelversion gesetzt. Man kann diesen aber auch mit „eselect kernel“ setzen. Jetzt kopiert man zuerst die alte Kernelconfig.

```
 cp /usr/src/linux-2.6.36-gentoo-r5/.config /usr/src/linux/.config
```

Danach überprüft und aktualisiert man diese mit

```
make oldconfig</pre>
Bei den folgenden Schritten kann man die ENTERTASTE einfach gedrückt halten.
Man die folgenden Fragen natürlich auch durchgehen und geg. Anpassungen
durchgehen. Ist die Config OK, kann man den neuen Kernel wie gewohnt bauen
lassen, und den Neuen Kernel nach /boot/ kopieren.
<code>make -j9 && make modules_install
mount /dev/sda1 /boot/
cp arch/x86_64/boot/bzImage /boot/mein_neuer_kernel
```

Event. noch die die Configdatei von Grub anpassen:

```
nano /boot/grub/menu.lst
```

Eine viel sicherere Upgrade-Methode ist es, die Konfigurationsdatei wie gerade gezeigt zu kopieren und dann einfach `make menuconfig` auszuführen. Dies vermeidet die soeben erwähnten Probleme von make oldconfig, da make menuconfig Ihre vorherige Konfiguration so weit wie möglich in das Menü lädt. Was Sie nun tun müssen, ist durch jede einzelne Option zu gehen und nach neuen

Abschnitten, Entfernungen und so weiter zu schauen. Durch Benutzen von menuconfig erhalten Sie einen Kontext für alle neuen Änderungen und können ganz einfach die neuen Auswahlmöglichkeiten sowie deren Hilfebildschirme anschauen. Sie können dies sogar für Upgrades wie 2.6.8 zu 2.6.9 verwenden; stellen Sie sicher, dass Sie sich vorsichtig durch die Optionen lesen. Sobald Sie fertig sind, kompilieren und installieren Sie Ihren Kernel wie gewohnt.

Externe Module erneut installieren

Wenn Sie Kernelmodule verwenden, die nicht im Baum der Kernelquellen enthalten sind, sondern an anderer Stelle im Portage-Baum bereitgestellt werden (z.B. ALSA-Treiber und NVIDIA- oder ATI-Grafiktreiber), müssen Sie diese nach Aktualisierung des Kernels erneut installieren. Dies wird einfach durch das erneute emergen der betreffenden Pakete erreicht. Wir liefern Ihnen ein einfaches Tool (`sys-kernel/module-rebuild`), welches alle Kernelmodule erneut erstellt, die Sie mit separaten Ebuilds für den Kernel unter '`/usr/src/linux`' installiert haben. Die Benutzung ist sehr einfach. Nachdem sie es emerged haben führen sie einfach module-rebuild populate aus um die Datenbank mit einer Liste von Paketen zu füllen die neu gebaut werden müssen nachdem der Kernel aktualisiert wurde. Nachdem Sie damit fertig sind ihren Kernel zu upgraden oder neu zu kompilieren starten Sie module-rebuild rebuild, um die Treiber neu zu kompilieren, bevor Sie neustarten.

Für weitere Informationen führen Sie module-rebuild ohne irgendwelche Optionen aus um eine Liste von Parametern zu bekommen, die dem Programm übergeben werden können.

```
module-rebuild populate
module-rebuild rebuild
```

Nachdem alle externen Module wie NVIDIA oder Virtualbox neu gebaut wurden, kann man sein System beruhigt neu starten.

Alte Kernel und Sources löschen

Nach dem letzten Abschnitt sind Sie möglicherweise mit Ihrem neuen Kernel glücklich sein und sehen keine Notwendigkeit darin, ältere Kernel zu behalten. Um einfach alle Quellen eines bestimmten Kernels außer des neuesten zu entfernen, können Sie sich die prune-Option von emerge zu Nutze machen. Wir führen das Beispiel mit den gentoo-sources fort:

```
emerge -P gentoo-sources
```

Das ganze wird aber auch bei einem `emerge -a –depclean` erledigt.

Folgende Dinge müssen manuell entfernt werden:

- /usr/src/linux-<alte_version>
- /lib/modules/<alte_version>

Weiter Informationen

- Kernelupgradeguide
- Kernelupgrade einmal anders

From:
https://deepdoc.at/dokuwiki/ - **DEEPDOC.AT - enjoy your brain**

Permanent link:
https://deepdoc.at/dokuwiki/doku.php?id=gentoo:kernelupgrade_nicht_genkernel

Last update: **2017/04/01 19:21**