Mit zehn Schritten zum eigenen Kernel:

1. Den aktuellen Kernel herunterladen und entpacken
2. make clean
3. make xconfig
4. Die Datei 'Makefile' bearbeiten
5. make dep
6. make bzImage
7. make modules
8. make modules_install
9. mv /usr/src/linux/arch/i386/boot/bzImage /boot/kernel2.4.19-xyz
10. LILO oder GRUB konfigurieren

1. Den aktuellen Kernel herunterladen und entpacken

Zuerst sollten Sie sich einen aktuellen Kernel besorgen. Oft wird der Quellcode des Kernels auch schon bei der Installation von Linux auf die Festplatte kopiert. Schauen Sie dazu im Verzeichnis '`/usr/src`' nach. Falls sich hier ein Verzeichnis mit dem Namen '`linux`' befindet, wechseln Sie in dieses Verzeichnis und lassen sich den Verzeichnisinhalt anzeigen. Wenn die Quellen des Linux-Kernels also schon vorhanden sind, können sie gleich bei Schritt 2 weitermachen.

Vielleicht möchten Sie aber den neuesten Kernel installieren, den es zur Zeit gibt. Den finden Sie auf der Internetseite von Kernel.org. Hier können Sie sich jederzeit den aktuellen Kernel herunterladen. Achten Sie aber unbedingt darauf, dass Kernel wie z.B. 2.5.18, d.h. Kernel, welche in der Versionsnummer an vorletzter Stelle eine ungerade Zahl haben (in diesem Fall die 5), Entwickler-Kernel sind, welche zwar sehr aktuell, aber doch auch recht fehleranfällig sind. Stabile Kernel haben als vorletzte Zahl in der Versionsnummer eine gerade Zahl, wie z.B. der Kernel 2.4.19, welchen Sie auf der Homepage von Kernel.org herunterladen können. Beachten Sie auch, dass der Kernel zur Zeit in zwei verschiedenen Formaten angeboten wird. Einmal als GZIP-gepackter Kernel (Endung .gz), sowie auch als mit dem neueren BZIP2 komprimierter Kernel (Endung .bz2). Sie sollten sich für den Kernel mit der Endung '.bz2' entscheiden, da die Downloadgröße kleiner ist und auf jedem aktuellen Linux-System auch das entsprechende Programm zum Entpacken der Datei vorhanden ist.

Den Kernel sollten Sie nun in das Verzeichnis '`/usr/src`' entpacken. Dabei wird ein neues Verzeichnis mit dem Namen '`linux`' angelegt. Sollten Sie vorhaben, die Quellen von mehreren verschiedenen Kernels zu entpacken, so empfielt es sich, den Namen '`linux`' des Verzeichnisses umzubenennen, z.B. in '`linux-2.4.19`' und dann einen symbolischen Link mit dem Namen '`linux`' auf das aktuelle Kernel-Verzeichnis zu legen. Dazu gibt es den Befehl '`ln -s ALTER_NAME NEUER_NAME`', der Befehl sieht dann z.B. so aus: '`ln -s linux-2.4.19 linux`'.

2. make clean

Nun sollten Sie den Befehl 'make clean' ausführen. Vergewissern Sie sich, dass Sie sich im Kernel-Verzeichnis befinden. Dorthin gelangen Sie mit '`cd /usr/src/linux`'. Durch '`make clean`' werden temporär im Kernel-Verzeichnis angelegte Dateien gelöscht und dafür gesorgt, dass der Kernel später auch wirklich neu kompiliert wird und nicht bereits kompilierte Teile wiederverwendet werden.

3. make xconfig

Jetzt muss der Kernel noch konfiguriert werden. Man hat die Wahl zwischen einer textbasierten Abfrage, einer menügesteuerten Konfiguration im Textmodus, sowie einer menügesteuerten Konfiguration im Grafikmodus.

Die textbasierte Abfrage wird mit 'make config' aufgerufen. Sie hat den Nachteil, dass dabei ein Parameter nach dem anderen angezeigt wird und jedesmal mit [y]es, [n]o oder [m]odule bestätigt werden muss. Hat man einmal eine Frage falsch beantwortet, so muss man die gesamte Konfiguration wieder von vorne anfangen. Als der Linux-Kernel noch bei der Version 1.2 war, war dies die einzige Möglichkeit, den Kernel zu konfigurieren, damals gab es allerdings auch noch nicht so viele Parameter wie heute, die konfiguriert werden mussten.

Die menügesteuerte Konfiguration im Textmodus wird mit 'make menuconfig' gestartet. Hier hat man den Vorteil, dass die Konfiguration nicht so linear wie bei der textbasierten Abfrage abläuft. Man hat die Möglichkeit, zuvor gewählte Optionen auch im Nachhinein noch zu verändern. Diese Variante ist auch übersichtlicher, da die Optionen in mehreren Bereichen zusammengefasst sind. Weiterhin hat man die Möglichkeit, die gerade erstellte Konfiguration unter einem beliebigen Namen zu speichern und dann jederzeit wieder aufzurufen, wenn z.B. nur kleine Änderungen am Kernel vorgenommen werden sollen.

Die menügesteuerte Konfiguration Grafikmodus kann nur aufgerufen werden, wenn das Grafiksystem X11 gestartet ist (Befehl: '`startx`'). X11 startet dann eine grafische Benutzeroberfläche wie z.B. KDE, Gnome, FVWM oder einen anderen Windowmanager. In einer Konsole geben Sie nun im Kernel-Verzeichnis den Befehl '`make xconfig`' ein. Daraufhin erscheint ein Dialogfenster. Hier können alle Einstellungen gemacht werden. Die meisten Optionen werden erläutert, wenn Sie auf den dazugehörigen '`Help`'-Knopf klicken.

Bei der Konfiguration sollten Sie möglichst viel als Modul konfigurieren, damit die Größe des Kernels möglichst klein gehalten wird.

4. Die Datei 'Makefile' bearbeiten

Wenn die Konfiguration erledigt ist, sollten Sie die Datei '`Makefile`' im Kernel-Verzeichnis bearbeiten. Sie enthält innerhalb der ersten Zeilen unter anderem die Bezeichnung des Kernels, sowie ein Feld, wo Sie die Bezeichnung noch erweitern können. Geben Sie in dieses Feld hinter dem '='-Zeichen z.B. '`-xyz`' ein, so heißt der Kernel von nun an z.B. '`2.4.19-xyz`'. Somit können Sie Ihren ganz persönlichen Kernel erstellen. Diese Bezeichnung bestimmt übrigens auch, in welchem Verzeichnis die Kernel-Module gespeichert werden. Aber dazu später mehr.

5. make dep

Nun müssen Sie '`make dep`' eingeben. Dies sorgt dafür, dass Abhängigkeiten im Kernel richtig aufgelöst werden.

6. make bzImage

Durch Eingabe von '`make bzImage`' wird der eigentliche Kompiliervorgang gestartet. Dies kann je nach Prozessor- und Arbeitsspeicherausstattung des Computers durchaus etwas länger dauern. Zum Schluss der Kompilierung erscheint (hoffentlich) eine Meldung, dass der Vorgang erfolgreich

war und der Ort, wo der Kernel jetzt gespeichert wurde, sowie dessen Größe. Normalerweise kann der Kernel nun unter '/usr/src/linux/arch/i386/boot/bzImage' gefunden werden.

7. make modules

Der Kernel selber ist nun fertig, es fehlen allerdings noch die Module, die der Kernel benötigt, um z.B. auf bestimmte Geräte zuzugreifen. Diese werden mit dem Befehl 'make modules' erstellt. Auch dies dauert wieder eine Weile.

8. make modules_install

Nun sind die Module erstellt, aber noch nicht am richtigen Ort. Der Befehl 'make modules_install' sorgt dafür, dass im Verzeichnis '/lib/modules/' ein Verzeichnis erstellt wird, in dem die Module gespeichert werden. Der Name dieses Verzeichnisses hängt vom Namen des Kernels ab. Heißt der Kernel z.B. '2.4.19-xyz', so befindet sich im Verzeichnis '/lib/modules' ein Unterverzeichnis mit dem Namen '2.4.19-xyz', welches (in weiteren Unterverzeichnissen) nun alle Module enthält.

9. mv /usr/src/linux/arch/i386/boot/bzImage /boot/kernel2.4.19-xyz

Nun müssen Sie noch den Kernel, welcher sich in der Regel im Verzeichnis '/usr/src/linux/arch/i386/boot' unter dem Namen 'bzImage' befindet noch an die Stelle verschieben, an der sich die Kernel auf Ihrem Linux-System befinden. Im Normalfall ist dies das Verzeichnis '/boot'. Sollte es bei Ihnen nicht vorhanden sein, erstellen Sie es am besten mit 'mkdir /boot'. Nun wird der Kernel in dieses Verzeichnis verschoben. Dazu gibt es den Befehl 'mv' (eine Abkürzung von '**m**o**v**e' = bewegen). Der vollständige Befehl lautet nun z.B. 'mv /usr/src/linux/arch/i386/boot/bzImage /boot/kernel2.4.19-xyz'.

10. LILO oder GRUB konfigurieren

Je nachdem, welchen Bootmanager Sie verwenden (ich persönlich bevorzuge LILO), müssen Sie diesen noch konfigurieren, damit der Bootmanager auch weiß, welchen Kernel er laden soll. Die Konfigurationsdatei für LILO befindet sich im Verzeichnis '/etc' und heißt 'lilo.conf'. Laden Sie diese in einen beliebigen Texteditor und fügen Sie in etwa die folgenden Zeilen an (Erläuterung folgt):

```
image=/boot/kernel2.4.19-xyz
label=MeinKernel
root=/dev/hda1
```

Die Zeile mit 'image=' enthält den Namen des Kernels mit vollständigem Pfad auf der Festplatte.

Die Zeile 'label=' enthält eine beliebige Bezeichnung für den Kernel, die Bezeichnung darf nur nicht zu lang sein.

Die Zeile 'root=' enthält das Gerät, welches das Linux-System beinhaltet. Sollte auf dem Computer nur Linux allein installiert sein, so ist dies bei (E)IDE-Festplatten oft '/dev/hda1', bei SCSI-Festplatten '/dev/sda1'. Um herauszufinden, auf welchem Gerät Ihr Linux die Rootpartition hat, geben Sie den Befehl 'mount' ein. Die Ausgabe enthält u.a. eine Zeile mit einem

Schrägstrich '`/`'. Dies kennzeichnet die Rootpartition. In der selben Zeile finden Sie auch die Angabe des Gerätes in Form von '`/dev/hda3`' oder ähnlich. Dies ist die Angabe, welche in die Zeile mit '`root=`' in der Konfigurationsdatei von LILO eingegeben werden muss.

Abschließend darf nicht vergessen werden, den Befehl '`lilo`' aufzurufen, damit der Bootloader die Konfigurationsdatei einliest und den Bootsektor der Festplatte neuschreibt.

Wenn Sie den Computer jetzt neustarten, sollten Sie die Möglichkeit haben, den neuen Kernel zu starten. Vorsichtshalber sollten Sie immer einen Kernel parat haben, der auf jeden Fall funktioniert. Es ist schon sehr ärgerlich, wenn ein falsch konfigurierter Kernel das Booten des Linux-Systems verhindert und ein alternativer Kernel nicht vorhanden ist.