



## Benutzung der Boot-Konfigurationen

### Kickstart-Dateien auf Festplatte

Auf einem Classic-Amiga gab es ein ROM, welches die grundlegenden Libraries enthielt: Exec, Dos, Intuition, usw. Das ROM ermöglichte es auf einer Disk zu booten, ohne daß ein System installiert war - "ohne Startup-Sequence" zu booten war die gängige Methode um am System "Erste Hilfe" zu leisten, indem man eine einfache Shell benutzte.

Uboot, das AmigaOne-ROM, initialisiert die Hardware und bootet ein Betriebssystem, stellt aber nicht alle nötigen Libraries bereit um das eigentliche AmigaOS zu booten - es kann ein System nicht "ohne Startup-Sequence" booten wie auf einem Classic. Immerhin stellt es Schlüssel-Komponenten (ExecSG, FFS2, usw.) zur Verfügung, welche wie alle anderen auf der Festplatte ausgelagert sind, und sich im Verzeichnis SYS:Kickstart befinden müssen. Der Vorteil dieser Vorgehensweise ist, daß man jetzt das Betriebssystem updaten kann, indem man einfach das entsprechende Modul auf der Festplatte ersetzt, anstatt ein neues ROM zu kaufen oder dieses jedesmal zu flashen. Der Nachteil ist, daß man die Boot-Partition leicht unbrauchbar machen kann, indem man beispielsweise eine falsche Library-Version installiert, ein Modul löscht, usw.

### Die Konfiguration der Kickstart-Module

Ist beim Versuch OS4 zu aktualisieren etwas schief gegangen? Das System bootet nicht mehr? Kein Problem !

Als guter "Amiganer" hat man dafür eine Backup-Konfiguration. Tatsächlich ist es möglich so viele Kickstart-Konfigurationen zu erstellen, wie man für nötig hält.

Um zu verstehen, wie das funktioniert, ist es nötig sich einmal im Verzeichnis SYS:Kickstart die Datei "Kicklayout" anzusehen; eine Textdatei, die man einfach mit dem bevorzugten Editor verändern kann und die die Konfiguration für den SLB (Second Level Bootloader) enthält. Das Layout ist einfach: wenn eine Zeile mit dem Wort "LABEL" beginnt, definiert sie den Namen der folgenden Konfiguration, alle Zeilen, die mit "MODULE" beginnen, kennzeichnen die Dateinamen der Kickstart-Module, die durch diese Konfiguration geladen werden sollen. Weiterhin gibt es eine Zeile "EXEC Kickstart/loader", die man aber im Normalfall nicht ändert - es ist das Programm, welches die Module in den Speicher lädt, bevor sie ausgeführt werden.

#### Beispiel:

```

LABEL Default
EXEC Kickstart/loader

MODULE Kickstart/kernel
MODULE Kickstart/a1ide.device.kmod
... usw.

```

Um mehrere Konfigurationen zu erstellen, muß man nur die Datei "Kicklayout" ändern. Als Beispiel kann man eine Default-Konfiguration erstellen, die automatisch geladen wird, wenn man das System bootet, zusätzlich eine Backup-Konfiguration, die immer funktioniert, eine weitere Konfiguration um Module zu testen, die man gerade entwickelt hat (falls man Entwickler ist) usw.

Im "Boot selector"-Menü von Uboot kann man auswählen, welche Konfiguration man gerade verwenden möchte, indem man den entsprechenden Namen im "Configuration"-Feld einträgt. Läßt man diese Einstellung in Uboot unverändert, so wird die "Default"-Konfiguration aus der Kicklayout-Datei geladen.

### Wie erstellt man mehrere Konfigurationen?

Man muß nur mehrere Konfigurations-Abschnitte in der Kicklayout-Datei erstellen, die jeweils eine Liste der zu ladenden Module beinhalten. Nehmen wir ein Beispiel: man hat ein Update des OS4-Kernel heruntergeladen und möchte ihn installieren; um auf der sicheren Seite zu sein, läßt man erstmal die "Default"-Konfiguration unverändert, die man bisher verwendet hat. Dann muß man nur alle Zeilen der "Default"-Konfiguration kopieren und am Ende der Kicklayout-Datei einfügen, den Dateinamen des gewünschten Kernels abändern und der neuen Konfiguration einen Namen geben:

```

LABEL Default
EXEC Kickstart/loader

```

```
MODULE Kickstart/kernel
....usw.
```

```
LABEL update
EXEC Kickstart/loader
MODULE Kickstart/updated-kernel
....usw.
```

Jetzt kann man in Uboot die "Update"-Konfiguration wählen und dem SLB damit mitteilen, daß er diese Konfiguration laden soll.

## Einfaches Management der Konfigurationen

Wenn man mehrere Konfigurationen erstellt hat, wird man sehen, daß mehrere Versionen des gleichen Moduls das Kickstart-Verzeichnis überfüllen können:

```
kernel
kernelv2
kernel-51.0
kernel-01-01-2004
....usw.
```

Das macht es schwieriger den Inhalt des Verzeichnisses zu lesen und man macht häufiger Fehler. Um Abhilfe zu schaffen, empfiehlt es sich, ein Unterverzeichnis pro Konfiguration anzulegen, welches das Erzeugen neuer Konfigurationen vereinfacht und was den Vergleich zweier Konfigurationen erleichtert, um die jeweils neueren Module zu finden.

Das Kickstart-Verzeichnis sieht dann so aus:

```
Kickstart (dir)
  ConfigOK (dir)
    module1.kmod
    module2.kmod
  ...
  Default (dir)
    module1.kmod
    module2.kmod
  ...
  Debug (dir)
    module1.kmod
  ...
```

## Schlußfolgerung

Dieses Vorgehen erlaubt es dem Anwender eine falsche Konfiguration leicht zu korrigieren. Wenn man schwerwiegendere Fehlfunktionen z.B. der Festplatte hat, wird der SLB die Kickstart-Module natürlich nicht lesen können. In diesem Fall ist es nicht möglich das Bootmenü zu erreichen um "ohne Startup-Sequence" zu booten. Immerhin kann man dann von der OS4-CD booten und so versuchen das Problem zu lösen. Hoffen wir, daß es eines Tages einen größeren Markt gibt und wir wieder ein erweitertes ROM haben können...

Geschrieben von:  
Philippe 'Elwood' Ferrucci

Redigiert vom Intuitionbase-Team

Übersetzt von:  
Philippe Bourdin

published: 24th June 2004

©2004-2011 IntuitionBase