

# ARCH LINUX BootSplash-HowTo

Copyright © 2003 Dennis Herbrich <dennis@archlinux.org>

Thu Aug 7 10:07:38 2003

## Zusammenfassung

Dieser kurze Leitfaden erklärt dem interessierten Nutzer wie sein System konfiguriert werden muss damit dieses einen hochauflösenden Splashscreen während des Bootvorgangs anzeigt und wie eigene Bilder als Hintergrund für die virtuellen Framebuffer-Konsolen benutzt werden können. Eine Anleitung wie der Kernel zu patchen ist wird ebenfalls ausreichend diskutiert.

## Inhaltsverzeichnis

<b>1</b>	<b>Vorbereitungen</b>	<b>1</b>
1.1	Was brauche ich alles? . . . . .	1
1.2	Wo bekomme ich das alles her? . . . . .	2
<b>2</b>	<b>Kernel-Einstellungen</b>	<b>2</b>
2.1	Wie patche ich den Kernel richtig? . . . . .	2
2.2	Eine Auflösung auswählen . . . . .	5
<b>3</b>	<b>Userspace</b>	<b>5</b>
3.1	Erstellung eines Themes . . . . .	5
3.2	Installation der userspace Programme . . . . .	6
3.3	Installation eines Themes . . . . .	6
<b>4</b>	<b>Hintergründe für virtuelle Konsolen</b>	<b>7</b>
4.1	Konfigurationsdateien . . . . .	7
4.2	Aktivieren der Hintergründe . . . . .	7
<b>5</b>	<b>Feedback</b>	<b>8</b>

# 1 Vorbereitungen

## 1.1 Was brauche ich alles?

Um das schnieke Splashscreen Feature zu verwenden benötigen Sie:

- den Kernelpatch für ihre Kernelversion
- ein JPG Bild in der richtigen Auflösung welches als Splashscreen genutzt werden soll (Duh!)
- die splash screen utilities
- eine Grafikkarte, die die VESA Framebuffer Kernelerweiterungen unterstützt
- wie üblich etwas freie Zeit

## 1.2 Wo bekomme ich das alles her?

Leider bietet die offizielle boot splash Homepage unter <http://www.boot splash.org> (noch) keinen Patch für den aktuellen 2.4.21 Kernel an, den ARCH LINUX schon seit längerem benutzt. Aber keine Sorge, Sie können den Patch und alles andere was Sie brauchen werden von <http://archlinux.veloxis.de/howtos/boot splash/downloads/>

herunterladen. Dort befindet sich auch ein Paket mit einigen 1024x768 Bildern die Sie nutzen können. Wenn Sie eine höhere Auflösung fahren wollen schlage ich vor, dass Sie die Homepage von e1904 unter <http://www.geocities.com/e1904/> besuchen, wo ich de facto diese Bilder ursprünglich herbekommen habe. An dieser Stelle noch einmal ein dickes Danke an e1904, nicht nur für seine (IMHO) hervorragenden Bilder, sondern auch für seine Hilfe ohne die dieses HowTo nicht möglich gewesen wäre! Wie dem auch sei, e1904 scheint keine kompletten Theme-Pakete für seine höheren Auflösungen anzubieten, so dass ich dringend dazu raten möchte erst einmal alles mit dem 1024x768 Paket ans Laufen zu bekommen, und danach erst eigene Experimente zu starten. Es ist so wesentlich einfacher.

## 2 Kernel-Einstellungen

### 2.1 Wie patche ich den Kernel richtig?

Da Sie ja ARCH LINUX' vorbildliches ABS benutzen können Sie recht leicht ihr eigenes Kernelpaket zur optimalen Integration in das Paketmanagement erstellen. Folgen Sie diesen einfachen Schritten:

1. Starten Sie *abs* als root um den ABS Baum zu aktualisieren
2. Kopieren Sie */usr/abs/kernels/kernel* (oder */kernel-scsi* falls Sie SCSI Unterstützung brauchen) nach */usr/abs/local/kernel* um eine gute Basis zu haben mit der Sie arbeiten können. Sie dürfen selbstverständlich ein evtl. vorhandenes eigenes Kernelverzeichnis nutzen.
3. Laden Sie den Patch herunter und kopieren Sie ihn in das lokale Kernelverzeichnis das sie soeben angelegt haben, direkt zu den anderen Patchdateien.
4. Editieren Sie die PKGBUILD Datei um den Namen des bootplash Patches zum "source" Array und eine korrekte "patch" Zeile direkt unter dem existierendem GCC Patch hinzuzufügen. Die PKGBUILD Datei sollte in etwa so aussehen:

```
[...]  
backup=('boot/kconfig')  
source=(ftp://ftp.kernel.org/pub/linux/kernel/v2.4/linux-$pkgver.tar.bz2  
        linux-gcc3-$pkgver.patch bootplash-3.0.7-$pkgver-vanilla-1.patch config  
        )  
  
build() {  
    rm -rf $startdir/pkg/*  
    cd $startdir/src/linux-$pkgver  
    patch -Np1 -i ../linux-gcc3-$pkgver.patch  
    patch -Np1 -i ../bootplash-3.0.7-$pkgver-vanilla-1.patch  
    cp ../config ./config  
    [...]
```

Sie sollten ferner den "pkgrel" Wert auf "custom1" oder etwas ähnliches setzen um Verwechslungen mit den offiziellen Kernelpaketen auszuschließen.

5. Erstellen Sie jetzt eine neue Kernelkonfigurationsdatei mit aktivierter bootsplash Option. Anfänger sollten einfach zur vorhandenen "config" Datei im ABS Kernel build Verzeichnis diese Zeilen am Ende hinzufügen:

```
CONFIG_EXPERIMENTAL=y
CONFIG_BLK_DEV_RAM=y
CONFIG_BLK_DEV_RAM_SIZE=4096
CONFIG_BLK_DEV_INITRD=y
CONFIG_BLK_DEV_LOOP=y
CONFIG_FB=y
CONFIG_FB_VESA=y
CONFIG_FBCON_SPLASHSCREEN=y
```

Alle bis auf die letzte Option sind im Arch-Standardkernel bereits aktiviert, so dass Sie nur die "FBCON" Zeile irgendwo in diese Datei schreiben müssen sofern Sie nicht bereits eine spezielle Konfiguration benutzen.

Diejenigen die bei dieser Gelegenheit gleich einen komplett eigenen Kernel basteln möchten sollten einen anderen Weg einschlagen und erst einmal die benötigten Kernelquellen herunterladen und in ein temporäres Verzeichnis entpacken. Danach müssen die Quellen gepatcht werden indem der bootsplash patch in das Kernelquellenverzeichnis kopiert wird und Sie daraufhin den Befehl *patch -Np1 -i ./bootsplash\** in diesem Verzeichnis ausführen. Der Patchvorgang sollte problemlos durchgeführt werden, so dass Sie im Anschluss mit *make menuconfig* den Kernel nach ihren Wünschen zu konfigurieren vermögen. Zuerst sollten Sie aber die Konfigurationsoptionen aus */boot/kconfig* laden und mit diesen als Basis weiterarbeiten. Diese Datei enthält die Optionen des momentan installierten Kernelpakets. Wenn Sie sich sicher sind, dass Sie diese Optionen nicht als Basis verwenden wollen wissen Sie vermutlich sowieso was Sie tun. Sobald sie bereit sind ihren Kernel zu konfigurieren stellen Sie sicher, dass die folgenden Optionen aktiviert sind (wählen Sie grundsätzlich "yes" anstatt "module" wo es möglich ist.);

```
Code maturity / Prompt
Block devices / Loopback device support
Block devices / RAM disk support
Block devices / (4096) Default RAM disk size
Block devices / Initial RAM disk (initrd) support
Console / Frame-buffer / Support for frame buffer
Console / Frame-buffer / VESA VGA graphics console
Console / Frame-buffer / Use splash screen instead of boot logo
```

Auch hier gilt, dass im Standardkernel alles bis auf die letzte Option normalerweise bereits aktiviert ist. Wenn sie die splash screen Funktion nirgends finden können, prüfen Sie, dass in den code maturity options die Options zur Anzeige von experimentellen Optionen aktiviert ist! Ferner sollten Sie prüfen ob der Patch anfangs fehlerfrei appliziert wurde. Wählen Sie keine Framebuffer Unterstützung ausser der für VESA, es wird sonst höchstwahrscheinlich nicht funktionieren! Als Modul können Sie sie behalten, aber ggf. möchten Sie sie komplett entfernen um Platz zu sparen. Vergessen Sie nicht ihre neue Konfiguration zu speichern und in das ABS Verzeichnis als neue "config" Datei zu kopieren.

6. Jetzt sollten Sie ihre PKGBUILD Datei dahingehend geändert haben, dass die Quellen automatisch gepatcht werden und eine modifizierte "config" Datei an Stelle des Originals die Bootsplash Option einbindet. Alles was Sie jetzt noch tun müssen ist "makepkg" zu starten und zu warten bis die Kernelcompilation abgeschlossen ist.
7. Installieren Sie das neue Kernelpaket und starten Sie ggf. LILO zur Aktualisierung

## 2.2 Eine Auflösung auswählen

Mit einkompilierter Framebuffer-Unterstützung brauchen Sie nur noch ihrem Kernel mitzuteilen welche Bildschirmauflösung er benutzen soll. Wie Sie vielleicht schon ahnen wird dies mit Hilfe eines Bootparameters erledigt den Sie zu ihrer *lilo.conf* oder der GRUB Kernelzeile hinzufügen müssen. Der Parameter lautet "vga=791" und aktiviert eine Auflösung von 1024x768x16. Beachten Sie, dass das boot splash Feature nur mit einer Farbtiefe von 16bit funktioniert!

## 3 Userspace

### 3.1 Erstellung eines Themes

Detaillierte Informationen über das Format der Konfigurationsdateien für bootsplash Themes finden Sie auf der offiziellen Homepage unter <http://www.bootsplash.org>. Die Erklärungen dort sind etwas knapp, aber immerhin komplett. Diese HowTo wird nur erläutern wie das arch-theme seine Magie wirkt; Wie weiterführende Spielereien wie Fortschrittsbalken und Animationen stehen auf der ToDo Liste für eine spätere Version dieser HowTo.

Nachdem das geklärt ist sollten Sie zuerst das Verzeichnis */etc/bootsplash/themes* erstellen und die heruntergeladene *arch-theme.tar.gz* dort entpacken um so ein neues Unterverzeichnis mit Namen "arch" zu erstellen welches die Theme-dateien dieses Beispiels enthält.

Ein paar Hintergrundinformationen damit alles etwas klarer wird; Das "splash" Programm des splashutils Pakets (dazu kommen wir später) erwartet eine Konfigurationsdatei zum Auswerten. Diese Konfigurationsdatei muss solch wichtige Informationen enthalten wie z.B. welche Bilder zu verwenden sind, wo die Textausgabe auftauchen soll, usw. Ein "Theme" ist also praktisch nur eine Sammlung von Konfigurationsdateien für das splash Programm und alle Bilder und zusätzlichen Dateien welche Sie benötigen.

Werfen Sie einfach einen Blick in die Konfigurationsdatei des Boot splash screens in *arch/config/arch-1024-boot.cfg*. Diese simple Datei enthält nicht wirklich viele Optionen; Tatsächlich sagt sie nur, dass es die Konfigurationsversion 3 nutzt, welche Bilder dargestellt werden sollen, welche Farbe für den Text und seinen Hintergrund genommen wird und in welches Areal jegliche Textausgabe zu finden sein soll. Nicht zu vergessen sind die Pfade zu den eigentlichen Dateien die angezeigt werden sollen, eine für die verbose Ausgabe (mit Text) und eines für silent. Der silent mode unterdrückt die Ausgabe der boot messages zu Gunsten eines optionalen Fortschrittsbalkens der allerdings ein wenig komplexeres Setup mit einem wrapper Skript voraussetzt. Deshalb lasse ich diesen erst einmal inaktiv. ;)

Basierend auf dieser Datei könnten Sie beispielsweise das referenzierte Bild nach ihren Vorstellungen modifizieren oder ein wenig mit den Textfarben herumspielen. Lassen Sie aber erst einmal alles in Frieden und führen Sie den Setup zuende.

### 3.2 Installation der userspace Programme

Dieses Beispiel nutzt nur das "splash" Programm. Um dieses mit ein paar anderen hilfreichen Werkzeugen zu installieren laden Sie das Binärpaket oder

die ABS PKGBUILD je nach Geschmack herunter und kompilieren/installieren Sie das splashutils Paket. Es wird letztlich eine "splash" binary mit ihren Freunden in */usr/sbin* installiert werden.

### 3.3 Installation eines Themes

Da der Kernel zu dem frühen Bootzeitpunkt an dem der splash screen gezeigt werden soll noch keinen Zugriff auf ihre Festplatten hat müssen Sie eine initrd anlegen. Glücklicherweise brauchen Sie das nicht selbst zu tun; Hier offenbart sich der ganze Nutzen des splash Programms, da dieses vollautomatisch eine initrd auf der Basis einer Konfigurationsdatei (=Theme) die Sie wählen anlegt. Alles was Sie tun müssen ist "splash -s -f /etc/bootsplash/themes/arch/config/arch-1024-boot.cfg *;* /boot/initrd.splash" auszuführen um eine neue initrd zu erstellen. Fügen Sie diese initrd zu ihrer lilo.conf oder GRUB Konfiguration hinzu, und vergessen Sie nicht ggf. LILO neu zu starten. Mal wieder.

An diesem Punkt möchten Sie sicherlich erst einmal neu starten um sich wie ein Schneekönig über den neuen, extrem coolen boot splash screen zu freuen. Wenn es nicht funktionieren sollte, durchsuchen Sie die dmesg Ausgabe nach Fehlern und beheben Sie sie.

## 4 Hintergründe für virtuelle Konsolen

### 4.1 Konfigurationsdateien

Eine andere schicke Sache die man mit dem "splash" Programm machen kann ist es ein Hintergrundbild für die virtuellen Konsolen zu setzen. Das arch-theme Paket enthält ein halbes Dutzend Konfigurationsdateien um genau dieses zu tun. Diese Dateien sind praktisch identische Kopien der boot Konfig, aber benutzen andere Bilder. Das ist es auch schon, alles andere blieb unangetastet. Ich habe mich entschieden für jede virtuelle Konsole ein eigenes Bild zu verwenden um sie leichter auseinander zu halten.

### 4.2 Aktivieren der Hintergründe

Ehrlich gesagt ist das eigentlich ein kruder Hack, aber es funktioniert prima. Um den Hintergrund einer VC zu setzen kann man die -u Option des splash Programms verwenden. Eine Handvoll dieser Kommandos in ihrer */etc/rc.local* wird die Hintergründe direkt nach erfolgreichem Boot setzen, und alles ist prächtig. Meine rc.local beinhaltet folgende Einträge:

```
# Set background images on consoles
/usr/sbin/splash -s -u 0 /etc/bootsplash/themes/arch/config/arch-1024-vt1.cfg
/usr/sbin/splash -s -u 1 /etc/bootsplash/themes/arch/config/arch-1024-vt2.cfg
/usr/sbin/splash -s -u 2 /etc/bootsplash/themes/arch/config/arch-1024-vt3.cfg
/usr/sbin/splash -s -u 3 /etc/bootsplash/themes/arch/config/arch-1024-vt4.cfg
/usr/sbin/splash -s -u 4 /etc/bootsplash/themes/arch/config/arch-1024-vt5.cfg
/usr/sbin/splash -s -u 5 /etc/bootsplash/themes/arch/config/arch-1024-vt6.cfg
```

Sie können natürlich mehr VCs so definieren, Sie müssen nur die vorhandenen Bilder und Konfigurationsdateien modifizieren und diese dann zur `rc.local` hinzufügen. Ich bin mir recht sicher, dass sie das zu Grunde liegende Prinzip verstanden haben.

## 5 Feedback

Feedback ist natürlich immer erwünscht, auch wenn es sich nur um eine kurze "Funktioniert bei mir!" Nachricht handelt. Wenn Sie die Zeit und Lust dazu haben die fortgeschrittenen Features der bootsplash Konfiguration wie z.B. Fortschrittsbalken, Boxes und Animationen zu erläutern sind sie herzlich dazu eingeladen! Bitte senden Sie jedwedem Feedback an <dennis@archlinux.org>.