

## Übung 3: Bootvorgang bei Linux

### Ziel der Übung:

Kennenlernen des Bootvorgangs unter Linux. Darüber hinaus werden weitere Linux-Kommandos vorgestellt und diverse Konzepte (*Daemon*, *rc-file*, ...) erläutert.

Pfad- und Dateiangaben in dieser Übung beziehen sich primär auf die CentOS-Distribution.

Linux Bootvorgang	
<b>POST (Power On Self Test)</b>	Zunächst Test und Initialisierung zentraler Hardware (CPU, BIOS-ROM, DMA-Controller, Tastatur). Als nächstes erfolgen der Test und die Initialisierung von System-Erweiterungen (RAM, Schnittstellen, Festplatten-Controller). Wenn diese Tests ok sind, dann sucht das BIOS auf den Bootgeräten nach einer gültigen Bootsequenz. <sup>1</sup>
<b>Bootloader</b>	BIOS lädt den MBR ( <i>Master Boot Record</i> ) des ersten eingetragenen Bootmediums. Der MBR enthält neben der Partitionstabelle ein Programm, das die Partitionstabelle auswertet. Ist eine dieser Partitionen mit einem <i>bootable Flag</i> markiert, wird deren Bootsektor angesprungen und der dort enthaltene Code ausgeführt. Ein <i>Bootmanager</i> wird benötigt, wenn beispielsweise zwei Betriebssysteme installiert sind. Der Benutzer kann beim Booten wählen, welcher Bootvorgang ausgeführt wird (Windows-Partition oder Linux).
<b>Initialisieren des Kernels</b>	Initialisierung aller Kernelteile (virtuelle Speicherverwaltung, Interruptroutinen, Zeitgeber, Scheduler, Dateisystem, Ressourcen der Interprozesskommunikation). Nach dem Hochfahren ruft der Init-Prozess (PID=1) abhängig von der Kernelkonfiguration erste Daemonen <sup>2</sup> ins Leben. Außerdem ist der Init-Prozess Vater aller Prozesse. Er greift auf die <i>/etc/inittab</i> , in der die systemspezifische Initialisierung definiert ist. Der Bootvorgang wird beispielsweise mit dem Starten eines weiteren Programms (z.B. Start der grafischen Oberfläche oder mit dem Start einer Shell) abgeschlossen.

(Hinweis: Hilfreich Informationen zu sämtlichen Linux-Fragen finden sich unter anderem unter LinWiki.org.)<sup>3</sup>

<sup>1</sup> Welche Bootmedien durchsucht werden und in welcher Reihenfolge dies geschieht, kann im CMOS-Setup eingestellt werden.

<sup>2</sup> Dienste, die von Serveranwendungen angeboten werden.

<sup>3</sup> <http://de.linwiki.org/wiki/Hauptseite> (abgerufen am 15.03.2014, 14:00 Uhr).

Runlevel:

Zum Startup bei Linux können verschiedene Runlevel definiert werden. Runlevels legen grob gesagt fest, welche Programme automatisch gestartet werden. Definiert sind die Runlevels in der Datei `/etc/inittab`. Ein Runlevel wird durch die Ausführung der entsprechenden rc-Datei initialisiert.

Übung:

Starten Sie ein Terminal und navigieren Sie zum Verzeichnis `/etc/`. Lassen Sie sich nun die Datei „inittab“ über den Befehl „cat“ anzeigen. (Alternativ können Sie die Datei auch über die grafische Oberfläche in einem Texteditor öffnen)

1. Wie ist der Runlevel 5 beschrieben?
2. Finden Sie heraus, wie das System auf die Tastenkombination CTRL+Alt+Delete reagiert und lassen Sie sich die dafür zuständige Datei anzeigen!
3. Lassen Sie sich über den Befehl `who -r` den aktuellen Runlevel anzeigen.
4. Versuchen Sie anhand der inittab-Datei herauszufinden, welche Skripte für Ihren aktuellen Runlevel ausgeführt werden und stellen Sie die Verkettung der Skripte grafisch dar!

Hinweis: Unter einem Hardlink versteht man einen Verweis auf eine andere Datei innerhalb des Dateisystems. Die Rechte bleiben in diesem Fall unverändert.

Ein Softlink hingegen hat ähnliche Funktionalität, allerdings können damit auch Rechte angepasst und Verzeichnisse adressiert werden.

Beide sind durch ein führendes „l“ in der Berechtigungskette (sichtbar über `ls -la`) zu erkennen

▪ **ps**

ps (process status)	
<b>Beschreibung</b>	Zeigt die Prozesse mit ihrem Status an.
<b>Syntax / Beispiele</b>	ps <Optionen> ps r: Nur laufende Prozesse

Hinweis: Mit der Verknüpfung eines Befehls über eine Pipe (= Umlenken der Ausgabe eines Kommandos als Eingabe für ein nachfolgendes; Der Operator für eine Pipe ist: „|“) mit dem Kommando *more* werden die Ausgaben an das more-Kommando übergeben (nützlich für lange unübersichtliche Ausgaben).

*Übung:*

1. Lassen Sie sich nach dem Startup mit `ps -x` alle laufenden Prozesse anzeigen. Finden Sie heraus welche Informationen mit diesem Befehl angezeigt werden. Was wurde initialisiert?
2. Was ist der Unterschied zwischen dem `ps`-Befehl und dem `top`-Befehl?

Hinweis: Kurze Erläuterung der wichtigsten Ausgaben:

PID	TTY	STAT	TIME	CMD	
Prozess-ID	Bezeichnung des aktuellen Terminals	Status	Verursachte CPU-Zeit	Programmname	



S = sleeping

R = running

SW = sleeping + File ausgelagert

▪ **kill**

kill	
<b>Beschreibung</b>	Beendet ein Programm.
<b>Syntax / Beispiele</b>	kill <PID> kill 1234: Beendet Prozess mit der PID 1234.

▪ **wc**

wc	
<b>Beschreibung</b>	Word count: Zählt die Anzahl der verwendeten Zeilen, Wörter und Zeichen.
<b>Syntax / Beispiele</b>	wc <dateiname> wc test1 test2 (Endezeichen= Strg + d) 2

*Übung:*

1. Öffnen Sie ein Terminal und geben Sie den Befehl „wc“ ein. Öffnen Sie ein zweites Terminal und versuchen Sie herauszubekommen, welche PID der wc-Prozess besitzt. Beenden Sie nun das wc-Programm des anderen Terminals mit Hilfe des `kill`-Kommandos. Was ist passiert?

**rc-File<sup>4</sup>:** Es handelt sich hierbei um eine Script-Datei in der Start-Anweisungen für eine Anwendung oder für das Starten von Systembestandteilen hinterlegt sind.

<sup>4</sup> <http://linux.about.com/cs/linux101/g/rcfile.htm> (abgerufen am 15.03.2014, 15:00 Uhr).

**Daemon<sup>5</sup>:** Unter einem Daemon versteht man ein Programm, das im Hintergrund abläuft und diverse Dienste zur Verfügung stellt. Beispielsweise startet der *cron-daemon* andere Programme bzw. Skripte zu festgelegten Zeiten.

Hinweis: Im Internet finden sich diverse Anleitungen zum Einrichten einer automatisch, zu festgelegten Zeiten ablaufenden Aufgabe mittels eines *cron-Batchdaemons*<sup>6</sup>.

*Übung:*

1. Informieren Sie sich im Internet über rc-Files<sup>7</sup>. Suchen Sie sich ein rc-File und sehen Sie sich den Inhalt an. Welche Programme / Daemonen werden gestartet?
2. Starten Sie unter System den System Monitor. Das Programm finden Sie je nach Linux z.B. unter / System Tools / System Monitor. Welche Daten können Sie diesem Programm entnehmen?

---

<sup>5</sup> <http://de.wikipedia.org/wiki/Daemon> (abgerufen am 15.03.2014, 15:10 Uhr).

<sup>6</sup> Ein möglicher Suchbegriff für Google wäre „Linux cron daemon“.

<sup>7</sup> Ein möglicher Suchbegriff für Google wäre „Linux rc files“.