



Consoliero

Mac OS X Consoliero – Referenz

Mac OS X Boot Sequenz

Von Moment wenn ein Benutzer seinen Mac einschaltet bis das Anmeldefenster erscheint, führt OS X ein Boot Sequenz durch welche den System auf seine Aufgabe vorbereitet. Diese Referenz beschreibt die einzelnen Vorgänge und zeigt die Konfigurationsdateien auf.

BootROM

Wenn ein Mac angeschaltet wird, die BootROM Firmware ist der erste Code der aktiviert wird. BootROM, welche Teil der Apple Hardware ist, hat zwei Aufgaben: die System Hardware initialisieren und ein Betriebssystem zum starten auszuwählen. BootROM hat zwei Komponenten welche helfen diese Aufgaben durchzuführen.

POST (Power-On Self Test): Initialisiert einige der Hardware Schnittstellen und überprüft das RAM.

Open Firmware: Initialisiert den Rest der Hardware, bildet die Abbildung der Hardware und sucht das zu startende System aus.

Die gleiche Version von BootROM kann OS 9 oder auch OS X starten.

BootX

Wenn BootROM OS X ausgewählt hat, geht die weitere Kontrolle an BootX Booter (`/System/Library/Core Services/BootX`). Die Aufgabe von BootX ist es die Kernel Umgebung zu laden. BootX zeichnet das „boot Image“ auf den Bildschirm.

Während die Kernelumgebung geladen wird, probiert BootX vorgehend gespeicherte Gerätetreiber zu laden (`mext cache`) für die Geräte welche während dem Bootprozess gebraucht werden. Sind diese Caches nicht da, schaut BootX in `/System/Library/Extensions` nach Treibern und anderen Kernel Erweiterungen.

Wenn der Kernel und alle Treiber geladen sind, BootX startet die Kernel Initialisierung Prozedur. Von diesem Punkt an kann nicht mehr auf die OpenFirmware zugegriffen werden.

Der Kernel initialisiert dann die Mach und die BSD Daten Struktur und dann das I/O Kit. Das I/O Kit verbindet die geladenen Treiber in den Kernel. Am Ende startet BootX den `mach_init` Prozess. Dies ist der Bootstrap Port Server welcher Mach Nachrichten verwaltet.

System Initialisierung

Nachdem das Root Filesystem durch BootX gemountet ist, beginnt die System Initialisierung die „System Startup Items“ zu laden und alle System Daemons. Wie bereits besprochen, Der `mach_init` Prozess startet den BSD `init` Prozess. Dieser Prozess hat die ID (PID) 1. Er „besitzt“ alle andern Prozesse auf dem System. Die Aufgabe von `init` sind folgende vier:

- 1: Er schaut ob der Benutzer den Singel User Mode aktivieren möchte oder ob ab einer CD-ROM gestartet wird. Und führt dann entsprechende Schritte aus.
- 2: Er startet und kontrolliert die System Initialisierungs Shell Scripte `/etc/rc.boot` und `/etc/rc` (auf die wir später noch im Detail kommen). Das `/etc/rc` Script startet das `systemStarter` Programm, welches die komplexere Prozeduren (Startup Items) ausführen kann (auch auf diese kommen wir noch im Detail).
- 3: Via dem `getty` Befehl, startet `init` das `loginwindow` Programm und führt die Benutzererkennung durch.
- 4: Als „Mutter“ aller Prozesse, führt `init` alle administrativen Aufgaben im Zusammenhang mit Prozessen und deren Beendigung durch.

rc.boot und rc Scripts

Die beiden Bourne Shell Scripts in `/etc` führen beide grundlegende Initialisierungs-Aufgaben durch. Zuerst für das Script `rc.boot` eine Konsistenzprüfung (`fsck`) durch, danach synchronisiert es den Speicher mit dem File System (`sync`). Danach führt das `rc` Script folgende Aktionen durch.

- 1: Es startet den Device Driver Loader (`kextd`).
- 2: Es startet den ATS Server (Apple Type Solutions Server).
- 3: Es startet den Windows Server.
- 4: Es startet den `update` Hintergrund Prozess, welche periodisch den File System Cache leert.
- 5: Es generiert das Swap File für das Virtual Memory und startet den Dynamic Pager.
- 6: Als letzter Schritt, startet das `rc` Script das `systemStarter` Programm um die Lokalen und System Startup Items abzuarbeiten.

Startup Items

Startup Items sind die Prozeduren welche als letzte in der Phase des Bootens ausgeführt werden. Die System Startup Items welche von Apple sind unter: `/system/Library/StartupItems`. Dieses sollten nicht verändert werden. Wenn man selber Startup Items erstellen möchte hat Apple für diesen das Verzeichnis `/Library/StartupItems` bereitgestellt.

Version	Ersteller	Datum	Prüfung	Druckdatum	Freigabe
1.0	Christoph Müller	1/5/03 14:52	Dave Uhlmann	1/5/03 15:11	Christoph Müller

Das Programm `systemStarter`, welches das Letzte ist welches `rc` ausführt, koordiniert die Ausführung aller Startup Items. Um zu verstehen was `systemStarter` macht, ist es hilfreich zu wissen was ein Startup Item ist. Ein Startup Item ist ein Ordner welcher im Minimum aus zwei Files besteht.

- Einem Programm (Normalerweise ein Shell Script) welcher den selben Namen wie der Ordner hat.
- Ein Konfigurations-File (`property-list`).

Das Konfigurations-File für jedes Startup Item heisst: `startupParameter.plist`. Dieses File spezifiziert den Namen des Startup Items und noch wichtiger, die Abhängigkeit. Also, es spezifiziert was gestartet werden muss und was schon laufen muss damit das Programm dann auch richtig funktioniert. `systemStarter` benutzt diese Informationen der Abhängigkeit um die Startreihenfolge festzulegen. `systemStarter` startet diese in Gruppen organisiert nach Abhängigkeit. Er probiert so viele wie möglich parallel zu starten. Wenn die eine Gruppe gestartet ist, wird die Nächste gestartet.

Die folgende Tabelle listet die „Core Startup Items“ von OS X auf. Die Reihenfolge ist die allgemeine Reihenfolge. Sie wird aber je nach Konfiguration bei Systemstart durch `systemStarter` festgelegt.

Weitere ausführliche Beschreibungen der Daemons sind im Consoliero Referenz „Mac OS X Prozesse“ festgehalten.

SystemTuning	Setzt die Performance Werte für das System, basierend auf diversen Faktoren, inkl. der Anzahl verfügbarem RAM.
SecurityServer	Startet den Security server, welcher die System Authorisierung und den Schlüsselbund observiert .
ConfigServer	Bildet eine Art Platzhalter für die Startup Items welche auf dem <code>configd</code> Daemon basieren.
Disks	Startet den <code>autodiskmount</code> Daemon. Überprüft und mountet lokale Disks.
Accounting	Startet das Accounting System <code>accton</code> , welches Accounting Informationen für jeden gestartet Prozess in das Verzeichnis <code>/var/account/acct</code> schreibt.
Network	Konfiguriert die lokalen Netzwerk Interface basierend auf den Daten in <code>/etc/iftab</code> . Setzt den Hostnamen, konfiguriert Netzwerk Routing, sowie IP Routing (wenn spezifiziert), Setzt den Host Id in den Kernel, und lädt die „shared IP kernel extension“ um eine IP Adresse über alle Programme, inkl. Classic, zu ermöglichen.
NetworkExtension	Lädt diverse Netzwerk Kernel Erweiterungen.
BIND	Startet den lokalen DNS Server.
mDNSResponder	Startet den Multicast-DNS Responder, welcher für Rendezvous gebraucht wird.
Portmap	Startet den <code>portmap</code> Daemon. (Siehe: Consoliero Referenz „Mac OS X Prozesse“)
SystemLog	Startet den <code>syslogd</code> Daemon. (Siehe: Consoliero Referenz „Mac OS X Prozesse“)
AppleTalk	Startet das <code>AppleTalk</code> Startprogramm in einem der folgenden Modes auf: Router Mode, multihoming (non routermode), Single Port. Die Einstellungen werden definiert unter: <code>/etc/hostconfig</code> .
CrashReporter	Wenn Crash Reporting eingeschaltet ist, wird das Programm <code>CrashReporter</code> gestartet.
LDAP	Startet den LDAP Server
NetInfo	Startet die Netinfo Servers, <code>netinfod</code> und <code>nibind</code> .
NIS	Startet Network Information Services.
DirectoryServices	Startet den name-resolver Daemon (<code>lookupd</code>).
NetworkTime	Startet den Dienst um die Uhr mit einem Netzwerkserver, über das Protokoll NTP, abzugleichen.
LoginWindow	Signal um zu signalisieren das für das System alle Service bereit sind um das Programm LoginWindow zu starten. Der <code>init</code> Prozess startet jetzt das Programm LoginWindow, während <code>systemStarter</code> weiter StartupItems im Hintergrund lädt.
SSH	Wenn „Remote Acces“ eingeschaltet ist wird der <code>sshd</code> Daemon geladen.
NFS	Startet das Network File System, welches die asynchronen I/O ausführt (<code>nfsiod</code>). (Siehe: Consoliero Referenz „Mac OS X Prozesse“)
Samba	Startet den Samba Server (<code>smbd</code>), welcher Netzwerk Anschluss via dem Server Message Block Protokoll (SMB) realisiert.

Version	Ersteller	Datum	Prüfung	Druckdatum	Freigabe
1.0	Christoph Müller	1/5/03 14:52	Dave Uhlmann	1/5/03 15:11	Christoph Müller

AuthServer	Startet den Authorisierungs Server.
AppService	Startet den core service daemin (<code>coreservicesd</code>).
PrintingServices	Startet CUPS (<code>cupsd</code>).
IPServices	Startet TCP/IP (<code>inetd</code>), Host Configuration Service (OOTP) und den „netboot client management server“.
AppleShare	Wenn eine Netzwerkverbindung entdeckt wurde startet es den AppleShare Service.
SendMail	Startet den ausgehenden Mail Service Daemon (<code>sendmail</code>).
Apache	Wenn Websharing eingeschaltet ist, wird der Apache HTTP Server gestartet (<code>httpd</code>)
Cron	Startet den <code>cron</code> Daemon.

Die Existenz eines Startup Items unter `/System/Library/StartupItems` garantiert nicht in jedem Fall das der Service für den Benutzer verfügbar sein wird. Jedes Startup Item hat Zugriff auf die Konfigurations-Informationen in `/etc/hostconfig` und benutzt diese Informationen um zu entscheiden ob der Service nun gestartet wird oder nicht. Nicht alle Startup Items lesen diese Informationen. Die Startup Items welche auf diese Informationen zugreifen sind: AppleTalk, AppleShare, AuthServer, Disks, SSH, SendMail, IPServices und Apache.

Christoph Müller, chm@pts.ch / www.pts.ch

macosx**alliance**

Version	Ersteller	Datum	Prüfung	Druckdatum	Freigabe
1.0	Christoph Müller	1/5/03 14:52	Dave Uhlmann	1/5/03 15:11	Christoph Müller