

## **Erledigt** Boot-Parameter

**Beitrag von „svenmac“ vom 29. Oktober 2013, 22:30**

Hallo an Alle!

Es gibt zwar schon einen Wiki-Eintrag über die Kernel-Flags hier in diesem Forum, aber es gibt auch einige Parameter die nicht dabei sind und diese würde ich gerne an dieser Stelle ergänzen. Wenn das der falsche Ort sein sollte bitte verschieben, Danke!



Erst einmal vielen Dank an den Griven für seinen Wiki-Beitrag.

Also fangen Wir mal an:

**-v**

Der Verbose-Mode, der wichtigste Kernel-Parameter, man sieht was passiert und kann rausfinden wo es hängen bleibt um gezielter nach Hilfe zu Fragen, oder sich selber zu helfen

**-x**

Der Safe-Mode, es wird nur das nötigste an Kexten und Booteinstellungen geladen

**-f**

Der Kext-Cache wird ignoriert und neu generiert. Wenn man neue Kexte dem System hinzugefügt hat und nach dem Reboot Kernel-Panics bekommt, sollte das helfen. Der Kext-Cache wird neu aufgebaut

**-s**

Der Single-User-Mode. Es wird keine Grafische Oberfläche geladen, man bekommt nur eine Eingabeaufforderung. Nur intressant für Leute die sich mit dem Terminal auskennen.

***PCIRootUID=1/0***

Wenn man eine ATI-Grafikkarte in seinem Hackintosh hat, sollte man PCIRootUID=0 setzen damit er startet oder der Bildschirm nicht schwarz bleibt. Ich für meinen Teil habe eine Nvidia-Karte und muss auch meine PCIRootUID auf 0 setzen. Ich glaube standart bei Chameloen ist PCIRootUID=1

### ***GraphicsEnabler=No/Yes***

Für viele Grafikkarten wichtig, es werden Treiber geladen. es kann sein das das System trotzdem startet und auch alles zu funktionieren scheint, aber QE oder Opengl Beschleunigung nicht.

### ***IGPEnabler=Yes/No***

Ist ähnlich wie GraphicsEnabler nur für CPU-Interner Grafikkern. Soll auch sehr nützlich sein wenn man eine Nvidia-Garfikkarte hat und mit der Internen gleichzeitig betreiben will.

### ***darwake=0/1***

OSX kann Teile des Systems schlafen schicken während andere Teile noch arbeiten. Es kann aber passieren das OSX die schlafen geschickten Teile des Systems nicht mehr aufwecken kann. Mit diesem Flag wird die Funktion Ein oder Ausgeschaltet.

### ***npci=0x2000/0x3000***

wenn man im Verbose-Mode startet und bleibt bei [PCI Configuration Begin] stehen sollte man mit diesem Parameter weiter kommen.

### ***dart=0***

Schaltet im Kernel den Support für IntelVT-d ab. Somit braucht man es im Bios/UEFI nicht abstellen und kann es im Dualboot für Windows nutzen. Ist für VmWare ganz witzig oder auch für Hyper-V. Ich konnt auch feststellen das OSX trotzdem VT-d nutzt, also scheint da nur die überprüfung abgestellt zu werden.

### ***cpus=x***

man kann die Anzahl der CPU-Kerne die OSX nutzen darf einschränken. Oft nützlich für AMD-Hackintoshes oder für Intel-User auf Sockel 2011 mainboards

### ***maxmem=x***

Damit kann man den Kernel zwingen nur eine bestimmte menge an Ram zu nutzen. Es war mal so das einige Installationen ab einer bestimmten Menge Ram Probleme machten. Dank an Leandros für diese Info

### ***arch=i386/x86\_64***

Mit diesem Parameter kann man OSX zwingen in einem 32 bit Betrieb zu gehen. Für manche AMD-Hackintoshes oder Atom-CPU's ganz nützlich. 64 Bit Programme funktionieren nach wie vor, allerdings kann jedes Programm nur 4GB Ram für sich beanspruchen. Was bei Videobearbeitung, Rendern, Sound- und Bildbearbeitung nicht unbedingt schön ist

### ***-legacy***

Ist für AMD-User interessant wenn sie den Kernel im 32 Bit modus starten müssen, müssen sie auch oft das Userland im 32 Bit Modus starten. inwiefern das noch unter Mountain Lion oder höher relevant ist kann ich nicht sagen. Da Mountain Lion und Höher Grundsätzlich 64 Bittig sind.

### ***-force64***

genau das Gegenteil von -legacy . Startet das Userland, aber nicht den Kernel im 64Bit Modus. Nach wie vor kann ich über den nutzen unter Mountain Lion oder Höher nichts sagen. ist im groben auch nur für User interessant die eine CPU haben die von OSX nicht unterstützt wird.

### ***-nosse3bit***

Schaltet die SSE3-Emulation des Kernels ein, wichtig für AMD Hackintoshes! Erst die FX-CPU's von AMD haben SSE3

### ***iog=0x0***

Für Macbooks interessant. Wenn man einen externen Monitor angeschlossen hat und den Deckel schliesst geht das Macbook nicht in den Ruhezustand. Auch interessant wenn man ein externes Display betreibt und Spiegelt kann man eine höhere Auflösung einstellen als das Interne Display unterstützt

### ***platform=X86PC/ACPI***

Powermanagement wird abgeschaltet/eingeschaltet

### ***idlehalt=0***

Soll Probleme auf Dual-Core Cpu's beheben. Wenn man kleine Hänger hat

***rd=disk0s1***

damit zwingt man den Kernel von einer bestimmten Partition zu Booten. in diesem Beispiel die erste Partition auf der ersten Festplatte

***Graphics Mode=1024x786x32***

Zwingt den Kernel mit der angegebenen Auflösung und Farbtiefe zu starten. In dem Beispiel in der Auflösund 1024 mal 786 in 32 Bit Farbtiefe

***?memory***

Zeigt Informationen zum verbauten Speicher an

***?video***

Zeigt Informationen zur verbauten Grafikkarte und die Unterstützten Auflösungen an

***Kernel=mach\_kernel***

Welchen Kernel man laden möchte. Nützlich für User die einen gepatchten Kernel nutzen. In diesem Beispiel wird der Kernel: mach\_kernel geladen (Standart). Die Kernel-Datei sollte im Wurzelverzeichnis der Festplatte liegen. Man kann Unterverzeichnisse, als Beispiel: "kernel=/atom/mach\_kernel" angeben

Das war es fürs erste. Wenn ich noch etwas finde, werde ich das hier gerne Ergänzen!

Gruß

---

**Beitrag von „Dr.Stein“ vom 29. Oktober 2013, 22:35**

TOP, Danke...

kann das Jemand in die WiKi packen ?

---

### **Beitrag von „Griven“ vom 29. Oktober 2013, 22:45**

Ich behalte den Thread hier im Auge und kombiniere es am bei Gelegenheit unter Erwähnung des TE mit dem bereits vorhandenen WIKI Inhalten. Ist denke ich sinniger als einfach den ganzen Post in die WIKI zu schieben 😊

---

### **Beitrag von „Leandros“ vom 30. Oktober 2013, 01:56**

maxmem wurde dafür verwendet um in den installer zu kommen, da dieser teilweise mit einigen RAM dimensionierungen nicht klar gekommen ist.

---

### **Beitrag von „HarryHirsch“ vom 28. November 2013, 01:23**

Hier wird aber einiges durcheinander geschmissen. Um das mal verständlich zu machen.

-v, -x, -s sind alles BootArgs und keine Kernelflags. -f gibt es auf einem Mac überhaupt nicht. Dieses Bootarg hat boot.efi noch nie unterstützt. GraphicsEnabler usw. betrifft nur Chameleon Derviate und/oder Kopien von dessen Code. CPUS=X ist dagegen ein Kernelflag welches vom mach\_kernel direkt unterstützt wird und nichts mit dem bootloader an sich zu tun hat.