

Khaki-Clover

Autor Slice



Original official document -

<https://github.com/CloverHackyColor/Clover>

[Bootloader/releases/download/5129/Clover_Khaki_Color_rus_5129.pdf](https://github.com/CloverHackyColor/Clover/releases/download/5129/Clover_Khaki_Color_rus_5129.pdf)

Übersetzt mit DeepL
Version 0.2.1
15.02.2021

Khaki Clover. Version 5.1 Revision 51291
Moskau, 2021

Inhaltsübersicht	
VORWORT.....	9
CHRONOLOGIE DER ENTWICKLUNG	10
TAKTISCHE UND TECHNISCHE MERKMALE.....	14
WAS IST WAS?	16
MBR SEKTOR.....	17
PBR SEKTOR.....	17
BOOT ODER CLOVEREFI.....	18
CLOVERIA32.EFI UND CLOVERX64.EFI ODER CLOVERGUI	19
ORDNERSTRUKTUR	19
EFI..TREIBER.....	21-
KEXTS BELASTUNG.....	23
ENTWICKLUNG	24
ANFORDERUNGEN.....	26
CODE..... SCHREIBEN	26
Kompilieren.....	27
ERSTELLEN DER DEBUG-VERSION VON CLOVER	30
ANLAGE.....	31
VERWENDEN DES INSTALLATIONSPROGRAMMS	31
DEN BOOTLOADER MANUELL..... INSTALLIEREN	34
OSX.....	34
Linux.....	35
Windows.....	36
EMPFOHLENE INSTALLATIONSOPTIONEN.....	36
Design	38
THEMA AUSWÄHLEN	38
Clover ... THEMEN.....	40
SCHNITTSTELLENKONFIGURATION IN CONFIG.PLIST	41
<KEY>GUI</KEY> 41	
<key>NurText</key> 41	
<key>KonsoleModus</key> 42	
<key>Thema</key> 42	
<key>EmbeddedThemeType</key> 43	
<key>Zeitzone</key> 43	
<key>PlayAsync</key> 43	
<key>BenutzerdefinierteSymbole</key> 43	
<key>Bildschirmauflösung</key> 44	
<key>Sprache</key> 44	
<key>Maus</key> 44	
<key>Ausblenden</key> 44	
<key>Scan</key> 45	
<key>Benutzerdefiniert</key> 45	
<key>Einträge</key> 45	
<key>Legacy</key> 45	
<key>Werkzeug</key> 45	
<key>Unterdrücken</key> 46	
<key>Scan</key> 46	
<key>CustomLogo</key> 46	
<key>ShowOptimus</key> 46	
ENTWURF: THEME.PLIST	47
<key>Komponenten</key> 47	

<key>BootCampStyle</key>	47
<key>Hintergrund</key>.....	48
<key>Banner</key>.....	48
<key>Auswahl</key>.....	49
<key>Schrift</key>.....	49
<key>Abzeichen</key>.....	50
<Key>Scroll</ Key>.....	51
<key>Anime</key>	51
<key>Ursprung</key>.....	53
<key>DesignWidth</key>.....	53
<key>DesignHöhe</key>.....	53
<key>Layout</key>	53
<key>Vertikal</key>.....	53
<key>BannerOffset</key>.....	54
<key>ButtonOffset</key>	54
<key>TextOffset</key>.....	54
<key>AnimAdjustForMenuX</key>	54
<key>HauptEinträgeGröße</key>.....	54
<key>TileXSpace</key>	54
<key>TileYSpace</key>	54
<key>AuswahlBigWidth</key>	55
VEKTOR-THEMEN	55
Warum brauchen Sie es	55
Wie man ein Vektordesign erstellt	56
Unterstützung für den SVG-Standard in Clover.....	58
Texte und Schriftarten.....	61
...Themen-Attribute	62
Fazit.....	63
KONFIGURIEREN DER HARDWARE	65
ERSTELLEN DER DATEI CONFIG.PLIST	65
BOOT.....	66
<key>Timeout</key> 66	
<key>Schnell</key> 66	
<key>DefaultVolume</key> 66	
<key>DefaultLoader</key> 66	
<key>Legacy</key> 66	
<key>Argumente</key> 67	
<Schlüssel>Debug</schlüssel> 67	
<key>NoEarlyProgress</key> 68	
<key>XMPDetection</key> 68	
<key>Sichern</key> 68	
<key>Policy</key> 69	
<key>WhiteList</key> 69	
<key>BlackList</key> 69	
<key>NeverHibernate</key> 69	
<key>SkipHibernateTimeout</key> 69	
<key>StrictHibernate</key> 69	
<key>RtcHibernateAware</key> 69	
<key>HibernationFixup</key> 70	
<key>SignaturFixup</key> 70	
<key>NeverDoRecovery</key> 70	
<key>DisableCloverHotkeys</key> 70	
SYSTEMPARAMETER.....	70
<key>CustomUUID</key> 70	
<key>InjectSystemID</key> 70	
<key>BacklightLevel</key> 70	

<key>InjectKexts</key>	71
<key>KeineCaches</key>	71
<key>NvidiaWeb</key>	71
SMBIOS	71
<key>Produktname</key>	71
<key>SmUUID</key>	72
<key>FirmwareFeatures</key>	72
<key>PlatformFeature</key>	72
<key>BoardSerialNumber</key>	72
<key>BoardType</key>	72
<key>BoardVersion</key>	73
<key>Mobil</key>	73
<key>ChassisTyp</key>	73
<key>ChassisAssetTag</key>	73
<key>SmbiosVersion</key>	73
<key>BiosVersion</key>	74
<key>EfiVersion</key>	74
<key>NoRomInfo</key>	74
<key>Vertrauen</key>	74
<key>Speicher</key>	74
<key>Schlitze</key>	76
CPU	78
<key>FrequenzMHz</key>	78
<key>BusSpeedkHz</key>	78
<key>UseARTFrequency</key>	79
<key>QPI</key>	79
<key>Typ</key>	79
<key>Speichermodus</key>	80
<key>QEMU</key>	80
<key>TurboDisable</key>	80
<key>HWPEnable</key>	80
<key>HWPValue</key>	80
<key>TDP</key>	80
GRAFIKEN	81
<key>GraphicsInjector</key>	81
<key>Inject</key>	81
<key>VRAM</key>	81
<key>LoadVBios</key>	81
<key>DualLink</key>	82
<key>BootDisplay</key>	82
<key>PatchVBios</key>	82
<key>PatchViosBytes</key>	82
<key>EDID</key>	83
<key>Inject</key>	83
<key>Benutzerdefiniert</key>	83
<key>ProduktID</key>	84
<key>VendorID</key>	84
<key>HorizontalSyncPulseWidth</key>	84
<key>VideoEingangSignal</key>	84
<key>VideoPorts</key>	84
<key>FBName</key>	84
<key>RadeonDelnit</key>	84
<key>NVCAP</key>	85
<Schlüssel>Display-cfg</schlüssel>	86
<key>NvidiaGeneric</key>	86
<key>NvidiaSingle</key>	86

<key>NvidiaNoEFI</key>	86
<key>ig-platform-id</key>	86
KERNEL AND KEXT PATCHES	86
<Schlüssel>Debug</Schlüssel>	86
<key>KernelCpu</key>	86
<key>FakeCpuID</key>	87
<key>AppleIntelCPUPM</key>	87
<key>AppleRTC</key>	87
<key>KernelLapic</key>	87
<key>KernelPM</key>	87
<key>KernelXCPM</key>	88
<key>DellSMBIOSPatch</key>	88
<key>KextsToPatch</key>	88
Patching mit Maske	90
Symbolic-Patching	91
<key>ForceKextsToLoad</key>	92
<key>ATIConnectorsController</key>	92
<key>ATIConnectorsData</key>	92
<key>ATIConnectorsPatch</key>	92
<key>KernelToPatch</key>	95
<key>BootPatches</key>	95
DEVICES (GERÄTE)	95
<key>Einspritzen</key>	95
<key>Eigenschaften</key>	95
<key>Audio</key>	96
<key>USB</key>	97
<key>FakeID</key>	97
<key>NoDefaultProperties</key>	98
<key>Eigenschaften hinzufügen</key>	98
<key>UseIntelHDMI</key>	99
<key>HDMIInjektion</key>	99
<key>Beliebig</key>	100
<key>ForceHPET</key>	101
<key>SetIntelBacklight</key>	101
<key>SetIntelMaxBacklight</key>	101
<key>IntelMaxValue</key>	101
<key>DisableFunctions</key>	101
<key>LANInjection</key>	101
RTVARIABLEN	101
<key>MLB</key>	101
<key>ROM</key>	101
<key>CsrActiveConfig</key>	102
<key>BooterConfig</key>	102
DISABLE DRIVERS	103
QUIRKS	103
<key>AvoidRuntimeDefrag</key>	104
<key>DevirtualiseMmio</key>	104
<key>MmioWhitelist</key>	104
<key>DisableSingleUser</key>	104
<key>DisableVariableWrite</key>	105
<key>DiscardHibernateMap</key>	105
<key>EnableSafeModeSlide</key>	105
<key>ProvideCustomSlide</key>	105
<key>ProvideMaxSlide</key>	105
<key>EnableWriteUnprotector</key>	105

<key>ForceExitBootServices</key>	105
<key>ProtectMemoryRegions</key>	105
<key>ProtectSecureBoot</key>	105
<key>ProtectUefiServices</key>	106
<key>ProvideConsoleGopEnable</key>	106
<key>RebuildAppleMemoryMap</key>	106
<key>SetupVirtualMap</key>	106
<key>SignalAppleOS</key>	106
<key>SyncRuntimePermissions</key>	106
<key>FuzzyMatch</key>	106
<key>KernelCache</key>	107
<key>AppleXcpmExtraMsr</key>	107
<key>AppleXcpmForceBoost</key>	107
<key>DisableIoMapper</key>	107
<key>LinkeditJettison deaktivieren</key>	107
<key>DisableRtcChecksum</key>	107
<key>DummyPowerManagement</key>	107
<key>ExternalDiskIcons</key>	107
<key>IncreasePciBarSize</key>	107
<key>PowerTimeoutKernelPanic</key>	107
<key>ThirdPartyDrives</key>	108
<key>XhciPortLimit</key>	108
ACPI	108
<key>ResetAddress</key>	108
<key>ResetValue</key>	108
<key>HaltEnabler</key>	108
<key>smartUPS</key>	108
<key>PatchAPIC</key>	109
<key>DropTables</key>	109
<key>FixMCFG</key>	109
<key>DisableASPM</key>	110
<key>SSDT</key>	110
<key>DropOem</key>	110
<Schlüssel>Generieren</Schlüssel>	110
<key>PLimitDict</key>	111
<key>UnderVoltStep</key>	111
<key>DoubleFirstState</key>	111
<key>MinMultiplikator</key>	111
<key>MaxMultipliiert</key>	111
<key>Generate</key>	112
<key>PluginTyp</key>	112
<key>DSDT</key>	112
<key>Debug</key>	112
<key>Name</key>	112
<key>FixMask</key>	113
<key>Fixes</key>	113
<key>ReuseFFFFFF</key>	114
<key>SuspendOverride</key>	115
<key>Patches</key>	115
Andere ACPI-Tabellen	116
<key>FixHeaders</key>	117
<key>RenameDevices</key>	117
DSDT-EINSTELLUNG	118
ADDDTGP BIT(0):.....	119
FIXDARWIN BIT(1):.....	119
FIXDARWIN7 BIT(16)	120
FIXSHUTDOWN BIT(2):.....	120
ADDMCHC BIT(3):.....	120

FIXHPET BIT(4):.....	120
FAKELPC BIT(5):.....	120
FIXIPIC BIT(6):.....	120
FIXSBUS BIT(7):.....	120
FIXDISPLAY BIT(8):.....	120
FIXIDE BIT(9):.....	121
FIXSATA BIT(10):.....	121
FIXFIREWIRE BIT(11):.....	121
FIXUSB BIT(12):.....	121
FIXLAN BIT(13):.....	121
FIXAIRPORT BIT(14):.....	121
FIXHDA BIT(15):.....	121
FIXRTC.....	121
FIXTMR.....	122
ADDIMEI.....	122
FIXINTELGFX.....	122
FIXWAK.....	122
UNBENUTZT..... LÖSCHEN	122
FIXADP1.....	122
ADDPNLF.....	122
FIXS3D.....	122
FIXACST.....	122
ADDHDMI.....	123
FIXREGIONEN.....	123
FIXHEADERS.....	123
AUSWAHL DER PATCHES.....	124
MANUELLES RICHTEN DSDT.....	124
Voraussetzungen	124
Erstellen eines leeren.....	125
Dekompilieren.....	125
Was zu reparieren.ist	127
Syntaxfehler.....	127
Bedeutung..der.. Irrtümer	129
NATIVE SPEEDSTEP.....	132
CONFIG ARRAY.....	133
CTRLLOOP ARRAY.....	133
CSTATEDICT.....	134
SCHLAFPROBLEM	134
HIBERNATE.....	135
ANWENDUNG	137
ERSTE BEGEGNUNG.....	137
WARUM STARTET DER Clover SO LANGSAM?	139
DEBUG.LOG/PREBOOT.LOG AUSWERTEN	141
BETRIEB VON OSX AUF NICHT UNTERSTÜTZTER HARDWARE	146
SCHRIFT SPERRE	148
STECKPLATZ-NAME (AAPL,STECKPLATZ-NAME).....	150
HDMI-TON.....	150
COMPUTER-STARTTON	150
NVRAM, IMESSAGE, MULTIBOOT.....	152
VERWENDUNG VON MEHRFACHKONFIGURATIONEN.....	154
WIE MACHE ICH BOOT.EFI NICHT ZU SPAMMIG AUF DEM BILDSCHIRM?	156
FAQ	157
F. Ich möchte Clover ausprobieren, wo soll ich anfangen?	157

F. Welche Version von Clover ist am besten für meine Hardware geeignet?	157
B. Nicht arbeiten	158
F. Ich habe Clover installiert, erhalte aber einen schwarzen Bildschirm	158
Frage Ich sehe 6_ auf dem Bildschirm und sonst passiert nichts	158
B. Es wird nur das textuelle Gegenstück des BIOS mit fünf Punkten geladen, der oberste - Weiter>	158
F. Ich habe Clover auf einem Flash-Laufwerk installiert, davon gebootet und kann meine HDD nicht sehen.....	159
F. Beim UEFI-Boot sehe ich keinen MacOS-Bereich, sondern nur Legacy	159
F. Beim UEFI-Boot sieht Windows wie ein Legacy-System aus, obwohl es EFI ist.....	159
Frage Stellen Sie die native Auflösung im Bootloader ein, aber der Bildschirm ist schwarz..... umrandet	159
B. Das Betriebssystem bleibt bei einem schwarzen Bildschirm hängen, wenn es versucht, zu starten	159
B. Der Kernel beginnt zu booten, gerät aber nach der zehnten Zeile Unable To find driver for this in Panik ACPI	160
C. Das System beginnt zu booten, friert aber beim Warten auf das Root-Gerät ein...	160
B. Das System bootet, bis die Meldung: Waiting for DSMOS.....	160
B. Das System passiert diese Meldung, aber es ändert sich nichts weiter, obwohl die Festplatte Brummen, als ob das System hochfährt	160
B. Das System fährt hoch und zeigt die Meldung an: [Bluetooth- Controller	160
B. Das System bootet, alles ist in Ordnung, aber der System Profiler hat Fehler...	160
SCHLUSSFOLGERUNG	160

Anmerkung zu den Farbmarkierungen:

Blau: Überschriften

Rot: Wichtig, unbedingt beachten

Grün: Veraltet, wird nicht mehr empfohlen

Gelb: Unklare Übersetzung (konnte auch nicht durch den Vergleich mit der englischen Übersetzung geklärt werden ev. mit Anmerkungen zur Übersetzung.)(s. S. 10)

Vorwort

Wovon reden wir hier? Sicherlich nicht eine Blume, die auf einer Wiese zur Freude der Kühe. Die Rede ist von der Software, dem Betriebssystem-Loader, der es Ihnen ermöglicht, ein ungewöhnliches System, Mac OSX, auf einem gewöhnlichen Computer laufen zu lassen, was Apple nicht zulässt, vor allem mit der Begründung, dass "wir keine Leistung für Computern, die nicht von Apple hergestellt wurden, erbringen." Nun, lassen Sie uns das System auf eigene Gefahr betreiben. Nun, es lohnt sich nicht, einen kommerziellen Vorteil daraus zu ziehen, um weitere rechtliche Komplikationen zu vermeiden.

Ein Nicht-Apple-Computer mit installierter Mac OSX-System wird Hackintosh genannt, die Herkunft des Wortes ist klar. Um MacOS auf einem Hackintosh zu starten, benötigen Sie einen speziellen Bootloader, und es gibt viele unterschiedlich, lassen sich aber in zwei Klassen einteilen: FakeEFI und RealEFI.

FakeEFI wurde vor vielen Jahren von David Elliot erfunden und funktioniert nach einem einfachen Prinzip: Tun wir so, als ob unser EFI bereits funktioniert hat, hinterlassen wir Spuren seiner Aktivität (Boot -args und den gesamten Tabellenbaum), lassen wir im Speicher EfiRuntime in der vereinfachten Form "Not unterstützt", und führen Sie mach_kernel aus. So funktioniert Chameleon, und es funktioniert erfolgreich, aber mit ein paar Ausnahmen wie dem Bedienfeld "Bootfähiger Datenträger". Es ist möglich, dass Apples Timing uns weitere Probleme beschert wird, die mit dem Fehlen der Runtime Services zusammenhängen. Januar 2013: Es ist passiert! iMessage funktioniert nicht mehr, weil es braucht SetVariable(), die in Chameleon "nicht unterstützt" wird. Irgendwie wurde sie überwunden, aber die Chameleon hat wieder Probleme. Legacy-Lader-Optionen: Chameleon, Enoch, Chimera, PC-EFI, Revoboot.

Können wir bis 2020 bereits den Tod dieser Methode feststellen?

RealEFI müsste anstelle des BIOS geflasht werden, aber für diejenigen mit einem Motherboard auf Das BIOS-basierte, bootfähige EFI wird erfunden. Dieses EFI-Boot-System auf einem Rechner mit BIOS wurde von Intel erfunden, ein Open-Source-Projekt unter tianocore.org. (Die Quelle das Projekt ist in Vergessenheit geraten! Es hat nur im Clover-Projekt überlebt). Eigentlich wird dieser Lader DUET genannt. Aber das Problem ist, dass EFI, das Betriebssystem geladen wird und gebootet, aber Mac OSX ist dort nicht vorgesehen. Der nächste erforderliche Schritt ist die Anpassung von DUET an Mac OSX-Anforderungen. Neuere Motherboards haben bereits EFI, aber es ist genauso unbrauchbar um den Hackintosh zu booten. Die EFI-Bootloader-Optionen fallen in zwei Kategorien: für PC-BIOS -bareboot, XPC, und für UEFI BIOS - Ozmosis,

OpenCore. Clover bedient beide Kategorien.

Der Name Clover ist einem der Gründer des Projekts zu verdanken kaby'l'a, der die Ähnlichkeit der "Command"-Taste, die es nur auf Macs gibt, mit vierblättrigem Kleeblatt.

Der vierblättrige Clover ist eine solitäre Kleepflanze, die eine starke mindestens ein Vierfachblatt, im Gegensatz zu herkömmlichen 3-Blatt. In der westlichen Tradition gibt es den Glauben, dass solche die Pflanze bringt dem Finder Glück, besonders wenn sie gefunden wird durch Zufall[1].

Die Legende besagt, dass jede der Platten des Vierfachtellers etwas repräsentiert die erste ist die Hoffnung, die zweite der Glaube, die dritte die Liebe und die vierte ist Glück[2].

*Das ursprüngliche grüne Logo sieht übrigens eher aus wie Hasen-Kohl als wie Klee. Bei uns **Vierblättriger Sauerklee** (*Oxalis tetraphylla*) auch Glücksklee, was ein Markenname für die Pflanze ist, genannt, und sie ist in Blumenläden sowohl dreiblättrig als auch vierblättrig erhältlich und hat botanisch nichts mit dem Klee zu tun :)*

In der russischen Version nennen wir das Programm "Clover". Das heißt, nicht **Clover(?)**.

Die Projektentwicklung findet in den Foren statt.



[1] Ich frage mich, wie man Klee zufällig finden kann? Durch regelmäßiges Zwickeln des Grases auf der Wiese?!

<http://www.projectosx.com/forum/index.php?showtopic=2562&st=0> RIP
<https://applelife.ru/threads/clover.42089/>
<http://sourceforge.net/projects/cloverefiboot/>
<https://www.insanelymac.com/forum/327-clover/>
<https://github.com/CloverHackyColor/CloverBootloader>

Es gibt viele andere Foren, die sich mit und über Clover beschäftigen, aber die eigentlichen Entwickler haben damit nichts zu tun.

Anmerkungen zu dem Text dieser Schrift:

Die rote Farbe hebt hervor, worauf Sie achten müssen. Die Farbe des Moderators.

Was veraltet ist und nicht mehr unterstützt wird, ist im Text in grüner Schrift gehalten, zur Referenz. Farbe der Form.

Überschriften sind blau hervorgehoben.

Schwarz dargestellte Schlüsselwörter sind **fett** hervorgehoben.

Zeitplan für die Entwicklung

Der Bedarf für einen neuen Lader ergab sich aus der Unfähigkeit von Chameleon, die das damals aufkommende System 10.7 (Lion) zu starten. Das Projekt wurde am 4. März 2011 auf Initiative von Kabyl ins Leben gerufen, der jedoch gegenüber allem, was er bis dahin zu verstehen vermochte, wich er der Entwicklung aus und verschwand bald ganz.

(Ich habe den starken Verdacht, dass er nicht mehr auf dieser Welt ist.)

Der erste MacOSX-Systemlauf mit einem modifizierten DUET fand am 6. April statt 2011.

<http://www.projectosx.com/forum/index.php?showtopic=2008&view=findpost&p=13810>

4. Mai werden gravierende Probleme des neuen Laders formuliert, ohne dass deren Lösung in einem neuen Projekt keinen Sinn machte. Die Pause zog sich bis August hin, denn allein mit diesen Problemen erschien mir unrealistisch.

In der Zwischenzeit erwachte Chameleon zum Leben und bewältigte den Lion-Boot und ich habe an meinem Brunch gearbeitet. Die Chameleon-Admins haben mich jedoch ignoriert, also habe ich es aufgegeben. Dann tauchte Ninja mit seinem iBoot auf, und ich schloss mich ihm an, um zu versuchen, einen EFI-Bootloader herzustellen und die Hängeprobleme zu lösen. Dieses Projekt begann im August 2011, und zwar Ich habe DUET (CloverEFI) mit der Summe aus CloverEFI+iBoot feinjustiert. Allerdings hat der chaotische Ursprung dieses iButtons nicht erlaubt, sich richtig zu entwickeln. 09. August 2011 hat dmdimon russische Schriftart für den Bootloader gemacht. Ich habe in der Zwischenzeit, SMBIOS und ACPI auf einer viel höheren Ebene zu betreiben gelernt, als es in Chameleon möglich war.

19. Oktober 2011 wurde endlich das Problem gelöst, Duet auf einem Laptop zu betreiben. Davor hat es einfach neu gestartet.

14. November 2011 Auftreten von Katzen in der Clover-Schnittstelle. Das heißt, für 10.4 zeichnen wir Tiger, für 10.5 Leoparden und so weiter. Tolle Innovation! Das Erscheinungsbild spielt eine Rolle.

14.12.2011 das Speicherpanik-Problem auf jüngeren OSX-Systemen gelöst, auf Lyon und älter hatten dieses Problem aus irgendeinem Grund nicht funktioniert.

05. Januar 2012 ist das Problem des Sleep gelöst worden. Von diesem Punkt an konnte man das Projekt als tragfähig betrachten. Zu diesem Zeitpunkt hatte Ninja die Szene bereits verlassen, und ich beschloss, die eigene grafische Menüprojekt des Bootloaders zu beginnen, basiert auf dem bereits bekannten rEFI. Es

lich sauber, und es war nun möglich, um die internationale Anerkennung des Projekts zu kämpfen. So ist Clover-v2 entstanden.

Die Erstellung der neuen Hülle dauerte zwei Monate, und die erste Veröffentlichung erfolgte am 29. Februar 2012. Eigentlich gab es rEFIt schon, es war nur nicht geeignet um in der EDK2-Umgebung kompiliert zu werden, und alle seine Bibliotheken mussten neu geschrieben und durch ihre eigenen ersetzt werden. Und alle hackintosh-spezifischen Dinge dem Projekt hinzu gefügt werden, die in der Tat, in Verbindung mit jadran. Projekt-Kompilierungstools entwickelt worden sind. Jetzt mit gcc-4.4, und jetzt 64 Bit.

09. März 2012 Dmazar, den ich seit August 2011 kenne, hat sich dem Projekt mit seine Idee, einen UEFI-Bootloader auf Basis von Clover zu erstellen angeschlossen.

31. März 2012 Geek baute Interaktivität für Eingabeparameter in der Shell Options Menu ist der Bootloader.

12. April 2012 crazybirdy hat ein Clover-Installationsprogramm erstellt.

21. April 2012 Dmazar hat den UEFI-Upload nicht geschafft, aber weiter an dem Projekt gearbeitet, um es zu verbessern und zu korrigieren. Das heißt, OsxAptioFixDrv wird erstellt.

05. Juni 2012 erschien pcj und bot seine Quellen mit neuen Technologien an: DSDT-Patch, Kexts Inject, Kernel-Patch, die unseren Bootloader auf ein ganz neues Niveau gebracht haben, ein für Wettbewerber unerreichbares Niveau.

18. September 2012. Pene rief mich und Dmazar an den runden Tisch, um darüber nachzudenken die iCloud-Probleme in den Griff zu bekommen. Das wurde bis 21. September 2012 gelöst.

30. September 2012 wurde das Aussehen der Maus in der Bootloader-Oberfläche erarbeitet.

19. Oktober 2012 wurden Animationen in der Bootloader-Shell eingeführt.

Oktober - Dezember 2012 Schritt für Schritt wurde die native Auflösung im Bootloader für Nvidia, ATI, Intel, für CloverEFI und für UEFI hinzugefügt. Diese Fortschritte haben die Messlatte für die Anforderungen an eine gute GUI höher gelegt

Ich möchte auch den Beitrag von Blackosx zur Verbesserung der Themenunterstützung erwähnen.

09. Januar 2013 bezahltes iMessage-Problem gelöst, das ging nicht in Chamäleon. Es war ein fundamentaler Sieg für die Idee eines EFI-Bootloaders. Nichts ist unmöglich, und in Chameleon einen Monat später wiederholten sie diese Methode, aber die Benutzer meinten, jetzt lass uns nach Clover benutzen.

Frühjahr 2013. Mit Jrcs Bemühungen wurde Clover mit zusätzlichen Dienstprogrammen und Internationalisierung in 20 Sprachen im Installateur, Bedienfeld und Service automatische Updates. Deutlich erhöhte Funktionalität der Skripte kompilieren, starten und beenden und automatischen Up-Dateservice kam hinzu. Der Bootloader hat sich zu einem Hackintosh-Wartungskomplex entwickelt

27. Juli 2013 endlich wurde das UEFI-Boot-Schlafproblem behoben, das seit Herbst 2012 bestand. Es scheint, dass es sich mit dem OS-Update angekündigt hat.

29. September 2013 wurde auch der UEFI Ruhezustand, Herunterfahren und Neustart korrigiert
Von diesem Punkt an können Sie UEFI-Booting als primäre Methode auf diesen Computer, wo immer möglich benutzt werden.

20. Januar 2014 Tiefschlaffähigkeit - Hibernation ist geschafft. Nicht in allen Fällen, aber im Moment ist Clover der einzige Downloader, der das kann.

Februar 2014. Auszeichnung zum Projekt des Monats bei sorceforge.net.

05. April 2014. Gibt den Abschluss der Entwicklung der Revision 2652 bekannt. Diese sind, Schließt mögliche Verbesserungen in der Zukunft natürlich nicht aus, nur dass sich nichts ändern wird. Das Wesentliche ist bereits erledigt.

Juni 2014. Apple veröffentlicht das System 10.10 DP1 Yosemite, die ersten Erfolge begannen. Installationen und neue Patches in Clover für das neue System. Und an dieser Stelle stellt sich heraus, dass Chameleon ist nicht in der Lage ist, dieses System zu booten. Es gibt einzelne erfolgreiche Berichte von bareBoot und Ozmosis Bootloader, die ebenfalls EFI-Bootloader sind, die einige von Clovers Codes verwenden. Clover hat sich zum wichtigsten Hackintosh-Lader entwickelt. Wieder ein Chamäleon korrigiert, es wird Yosemite geladen, aber noch keine iMessage.

21. August 2014. Dmazar behebt NVRAM-Betriebsfähigkeit des UEFI-Boot für einige Leute, bei denen es vorher nicht funktioniert hat.

In der Zwischenzeit hat die Version 2k die 3000er Revision passiert. Ein neuer Compiler mit LTO-Optimierung, neue Codegröße, Behebung alter Bugs, Verbesserungen in Algorithmen. Fast nichts Neues, aber Clover wurde besser.

Januar 2015. Da das von Apianti angekündigte Clover 3 noch aussteht, ist Clover 2 die Version 2.3k geworden, die der Revision >3000 entspricht.

Alle Entwickler verstreuten sich, projectosx.com hörte auf zu existieren. Ich bin zurückgeblieben allein, aber ich arbeite weiter, wenn auch ohne großen Eifer. Ich habe ein neues Spielzeug - Wein.

Juni 2015. Das System 10.11 El Capitan ist erschienen. Clover hat es heruntergeladen, ohne Fragen zu stellen. Chameleon steckt wieder fest. Das haben sie dann gelöst, aber so wie ich es sehe, ist das Interesse des durchschnittlichen Benutzers an Chameleon nicht mehr groß. Nur die gleichen sechs Personen, die interessiert sind.

Unterstützung. Chameleon/Chimera wird immer noch von Neueinsteigern verwendet, die mit Multibista und die noch nichts von Clover gehört haben, oder AMD-CPU-Anwender, die keine Entwickler sind, und die wissen einfach nicht, was sie mit Clover machen sollen. Ein neues Problem ist, Inject Die Lösung wurde von einem zufällig vorbeikommenden Benutzer geliefert.

solstice (Revision 3258). Ozmosis-Anwender standen lange Zeit ohne ein neues System da, bevor seine Entwickler eine neue Version mit diesem Patch bereitgestellt haben.

Frühling 2016. Zenith432 hat eine Kompilierung mit Xcode vorgenommen, wir brauchen nicht mehr gcc.

Sommer 2016. Mit den neuen Kodierern gibt es auch größere Änderungen in der Clover-Oberfläche. Scrollen, wo immer Sie es brauchen, Checkboxen und Radio-Buttons, neue Designstile.

Herbst 2016. Zusammen mit vit9696 wird die lang erwartete Technologieunterstützung gemacht: FileVault2.

Winter 2017. Wir sind mit der Revision 3999 unterwegs. Als nächstes sind 4000 und die Version 2.4k an der Reihe. Auf dem Clover-Weg. muss noch weiter entwickelt werden, weil neue Systemunterstützung benötigt wird, hier für den Sierra 10.12.4, zum Beispiel, das einen neuen Patch benötigte.

Sommer 2017. Es erschien das High Sierra-System mit dem neuen apfs-Dateisystem. Für das der Chamäleon-treiber noch nicht verfügbar ist. Für EFI-Lader gibt es einen nativen von Apple. Übrigens: Multibeast ist längst nach Clover umgezogen, ebenso wie andere kommerzielle Projekte.

Herbst 2017. Beseitigung des schwarzen Bildschirms für Radeons Grafikkarten. Dies wurde von vit9696 in seinem Kext gemacht. WhateverGreen, Mieze hat mir erklärt, wie das geht, und ich habe als Checkbox in der Config RadeonDeInit=true. Die Lösung ist unvollständig, im Sinne der Grafikfabrik, aber ausreichend gut für schlafende Radeons. Und wieder hat vit9696 eine Lösung gefunden, wie man AptioFix repariert, damit die native Version des NVRAM funktioniert, fast jeder hat es jetzt!

Frühjahr 2018. vit9696 hat herausgefunden, wie die RTC in macOS verwendet wird, und damit sind das Problem der Aufbewahrung des Ruhezustandsschlüssels und andere RTC-Probleme gelöst.

Juni-Dezember 2018. Unterstützung für SVG-Vektorgrafiken, und dementsprechend, skalierbare Themen.

Januar 2019. Das Auftreten von Ton in der Clover-Oberfläche, dank Goldfish64.

Juli 2019. Eine Reihe von neuen Patches zum Booten von 10.15. Und Revision 5000! Jetzt hat Clover die Version 2.5k.

Mai 2019. vit9696 hat einen neuen OpenCore-Bootloader basierend auf mit seinen neue Ideen. Die Ideologie ist vielversprechender, aber auch Clover lebt weiter und entwickelt sich weiter. Seien wir koexistieren und kooperieren. Clover sollte vom Unnötigen, vom Krumpfen befreit werden, das Beste von OpenCore nehmen, aber es ist eine unrealistisch große Aufgabe, die nur langsam verbessert wird.

Ich möchte auch erwähnen, dass Vector-sigma dem Clover-Team beigetreten ist den gesamten Benutzerbereich. Steuert Kompilierskripte, Laufzeit, Installer und erstellt eine spezielle Clover.app. Das heißt, sie steuert alles, was früher unter JrCs Verantwortung war. So lebt und entwickelt sich auch der Loader-Wrapper.

September 2019. Wir haben das gesamte Projekt einer Generalüberholung unterzogen, sind zu GitHub umgezogen, das stabil und schnell wurde, nachdem Microsoft die Seite gekauft hatte, und damit auch die folgenden Änderungen:

- Die Clover-Quellen werden jetzt nicht mehr über das SVN-Protokoll, sondern über das GIT-Protokoll gesteuert. Hier hat es seine Nachteile, nämlich das Problem mit der Revisionsnummerierung, aber es hat auch seine Vorteile, zum Beispiel die Möglichkeit, eine Bisektion durchzuführen, um einen kürzlich gemachten Fehler zu finden.

- Ich bin es leid, seltsamen "Verbesserungen" des EDK2 zu folgen. Davor verwendeten wir das stabile UDK2018 Version, aber es ist eindeutig veraltet, und mit dem neuen EDK2 war Clover gerade nicht kompatibel. Die Lösung ist die Einbindung aller notwendigen Bibliotheken aus dem neuen EDK2 einfach in das Clover-Projekt mit unseren Patches ein, und fügen Sie die Bibliotheken, die bereits wurden davon ausgeschlossen, indem sie aus dem alten UDK2018 übernommen wurden. Wir sprechen über Legacy-Computer-Support auf Core2Duo und dergleichen. Jetzt benötigt Clover keine externen Bibliotheken, alle notwendig ist, ist in seinem Repository enthalten, und all dies wird von den Entwicklern aktualisiert von Clover selbst, basierend auf Änderungen im EDK2. Übrigens, die Halbierung wurde möglich genau mit dem Verzicht auf jegliche externe Quellen, alles was benötigt wird, ist in einem Ordner.

- Der Hauptcompiler des Projekts ist nun gcc-10, im Gegensatz zu Klang kann er LinkTimeOptimisation zu implementieren, d. h. unbenutzte Bibliotheken zu verwerfen bei Verknüpfungsstufe. Die Codegröße verringerte sich von 960kb auf 870kb! Zugleich erhalten wir zusätzliche Diagnose für mögliche Fehler. Eine davon ist, dass EDK2 erlaubt die Verwendung von Zeigern auf ein leeres Array, und dies widersprach einigen Algorithmen in Clover. Nach einem kleinen Tweaking hat sich das Design anders entwickelt! Und könnte auch sein, daß die anderen Fehler verschwinden.

- die Struktur der Treiberordner geändert, da wir die Idee des Kompilierens komplett aufgegeben haben in 32 Bit, und die Angabe von 64 Bit ist überflüssig.

- kexts Ordnerstruktur geändert, jetzt können Sie in der Oberfläche verbinden und trennen verschiedene Versionen der CloverKonfiguration, um zu prüfen, welche am besten funktioniert.

Januar 2020. Die nächste große Transformation von Clover. Es gibt einen neueEntwickler Jief Machak, der anbot, das Projekt in C++ zu übersetzen. Dies ist nicht eine triviale

Aktion, da sie die Änderung einiger Bibliotheken und Kompiliermethoden erfordert. Aber es er kam nicht mit leeren Händen, sondern brachte die notwendigen Akten mit, die bis ins Jahr 1997 zurückreichen, und hatte offensichtlich schon damals ein solches Problem mit C++ gelöst. Und jetzt passt es sich an die neue Bedingungen an.

Ich nahm die Herausforderung an, und hier sind wir nun, als Team, und haben die Transformation in Clover bereits begonnen. Auch Pene hilft uns. Natürlich impliziert C++ auch das Kompilieren von Dateien, die in C geschrieben wurden, also dass in der ersten Phase des Übergangs zu C++ nichts passiert ist. Aber es macht keinen Sinn, zu sagen über die neue Sprache, sondern um in der alten zu programmieren. So nutzen Sie die Vorteile von C++, wir müssen sie benutzen.

Lassen wir Linus Torvalds in Ruhe mit seinen Flüchen an die Adresse von C++, seine Beweggründe sind klar, Linux ist bereits in C geschrieben, und es kann nicht neu gemacht werden, Alles, was wir tun können, ist nach Ausreden zu suchen. Und wir können unsere Quellen umschreiben, zumindest teilweise, aber Clover ist viel kleiner und man kann nicht alles neu schreiben, unter Berücksichtigung der Kompatibilität der Sprache. Und doch, was tun wir? Neue Paradigmen einführen programmieren und die Codes neu schreiben, um die neuen Anforderungen zu erfüllen. Und dieses Element führt zu sein Umdenken, die Entwicklung neuer Algorithmen und die Beseitigung alter Bugs, Nudeln aufdröseln. Die GUI ist auf neue Algorithmen, auf Klassen und damit auf alte die Themen haben gelitten, aber sie können geheilt werden, und Clover ist noch nicht ganz fertig, work in progress. Und das Ergebnis der Transformation ist ein schnellerer Betrieb und die Beseitigung von Fehlern, die sie früher nicht entwirren konnten.

Sommer 2020. Das Auftauchen des 11.0 Big Sur Systems hat gezeigt, dass der Mythos der Nicht-Letalität von Zeichenpatches unhaltbar ist. Es ist kaputt, sowohl in Clover als auch in OpenCore. Nun, wir machen weiter.

Herbst 2020. Wieder hat Jief Machak die Aufgabe übernommen, die Clover-Lader zu verbinden und OpenCore lädt bereits BigSur, da vit9696 die Kexts direkt in Cache lädt, und Clovers Methode der Injektion in den BigSur-Gerätebaum mit seiner neuen Sicherheitsanforderungen abgelehnt. Was ist einfacher, das Problem herauszufinden oder die Methode von der Konkurrenz? Jif beschloss, den zweiten Weg zu gehen. Nun, so sei es. Wie kombinieren? Vitalys Idee, Clover GUI als einen Satz von Bibliotheken zu nehmen und sie an OpenCore anzupassen, ist leider unrealistisch, Clover ist bereits in C++ geschrieben, und zurück zu C-Bibliotheken wird es niemand mehr zerschneiden. Aber um OpenCore-Bibliotheken zu nehmen und sie von Clover aus auszuführen, kein Probleme, nur mühsame Arbeit mit Delegation. Also hat er es getan. Jetzt gibt es ein zusammengeführter Bootloader, aus zwei Repositories,

<https://github.com/CloverHackyColor/CloverBootloader>

<https://github.com/CloverHackyColor/OpenCorePkg>

kompiliert als Einheit, läuft als Clover und sieht aus wie Clover mit seiner GUI und Einstellungen und delegiert bei der Auswahl des zu ladenden Systems die Autorität an OpenCore und gibt ihm die Kontrolle. Ja, OpenCore ist nicht das Original, denn sein Team wird nicht kooperieren, da haben wir keine andere Wahl, als unsere eigene Version zu betreiben und unsere eigenen Änderungen vorzunehmen.

Taktische und technische Merkmale

EFI - Extensible Firmware Interface - Erweiterbare Hardware-Zugriffsschnittstelle abhängige Funktionen. Im Gegensatz zum BIOS, das 64kb belegt und in 16-Bit Codes geschrieben ist, EFI belegt ab 4MB, wird in 32- oder 64-Bit-Codes geschrieben, und ist Hardware-unabhängig, obwohl... natürlich geschehen keine Wunder, und 100%ige Kompatibilität mit jedem Plattform ist nicht zu erreichen. Das eigentliche UEFI-BIOS ist natürlich hardwareabhängig.

Clover ist ein EFI-Bootloader für Betriebssysteme, für Computer, die bereits über UEFI-BIOS (Unified EFI...), und für Computer, die kein solches haben. Die Betriebssysteme können EFI-Booten unterstützen (OSX, Windows 7-64EFI, Linux), oder nicht (Windows XP), in letzterem Fall gibt es einen Legacy-Boot - eine Rückkehr zum alten BIOS-Bootstrap-Schema über Bootsektoren. Streng genommen ist Clover kein Bootloader, sondern Boot Manager, der die Vorbereitung und den Start von nativen Bootloadern verschiedener Betriebssysteme durchführt (boot.efi, grub.efi, bootmgfw.efi).

EFI ist nicht nur die erste Stufe des Bootens eines Betriebssystems, es erstellt auch Tabellen und Dienste, die für die Verwendung im Betriebssystem verfügbar sind, und seine Leistung hängt von der Korrektheit dieses Schrittes. Sie können OSX nicht über das eingebaute UEFI booten, genau wie Sie nicht OSX von einem sauberen Duet booten können. CloverEFI und CloverGUI bereiten eine ganze Menge zur Anpassung der eingebetteten Tabellen, um OSX ausführen zu können:

- Die SMBIOS (DMI)-Tabelle wird mit Daten bestückt, die reale Computer emulieren. Apple Macintosh ist eine Voraussetzung für den Betrieb von OSX. Die Seriennummern sind fiktiv, aber angemessen. Es ist jedoch wünschenswert, dass der Benutzer eindeutige Nummern verwendet. Warum lädt GRUB z. B. HackOS nicht? Weil sie nicht das Recht haben, um die Seriennummern der Mohnblumen aufzunehmen...

- ACPI-Tabellen, die im ROM eines Computers enthalten sind, enthalten normalerweise Fehler und Unzulänglichkeiten, die meist auf die Faulheit der Hersteller zurückzuführen sind: In der APIC-Tabelle ist die falsche Anzahl der CPU-Kerne, NMI-Daten fehlen, und das Reset-Register fehlt in der FACP-Tabelle, falsches Leistungsprofil, SSDT-Tabellen haben keine Daten für EIST, und über DSDT im Allgemeinen ist ein langer Vortrag. Das heißt, MacOS hat seine eigenen Anforderungen für ADCP, die mit den ADCs herkömmlicher Computer nicht mithalten können. Clover versucht all dies zu korrigieren;

- OSX versucht auch, zusätzliche Gerätedaten vom Bootloader zu erhalten, wie z. B. Grafikkarte, Netzwerk, Sound usw. über den String-Mechanismus der Geräteeigenschaften. Clover generiert solche Informationen;

- BIOS-basierte Computer zeichnen sich durch die Verwendung von USB im Anfangsstadium im Legacy-Modus aus, die bei der Übergabe der Steuerung an OS unakzeptabel ist. Der Bootloader führt die USB-Betriebsartenumschaltung durch;

- OSX tauscht auch Informationen mit der EFI über ein spezielles NVRAM aus, auf die über RuntimeServices zugegriffen wird, die nicht im Legacy Lader gibt. Clover bietet einen solchen Informationsaustausch, und zwar in beide Richtungen, die einen korrekten Firewire-Betrieb ermöglicht, die Möglichkeit, Nvidias Web-Treiber herunterzuladen, die Möglichkeit der Verwendung des Bedienfelds "Bootable Disk" für automatischer Neustart auf ein anderes System. NVRAM wird benötigt, um iCloud- und iMessage-Registrierung durchführen zu können, für den Ruhezustand, für FileVault2 und et al;

- ConsoleControl-Protokoll, nicht in Duet verfügbar, andere ähnliche Kleinigkeiten;

- ist es notwendig, einige Daten in EFI/Platform über das DataHub-Protokoll einzutragen, das ist in Duet abwesend und in UEFI nicht immer vorhanden. Die schwerwiegendsten FSBFrequency-Wert, dessen Definition Aufgabe des Bootloaders ist, die Daten in DMIs sind ungenau oder nicht vorhanden sind, und wir verwenden dieses Protokoll auch für Sensoren, aber nicht alle Hackintosh-Nutzer verwenden es, und einige gehen ihren eigenen Weg, und Ich frage mich, warum es nicht funktioniert;

- Die CPU muss ordnungsgemäß initialisiert werden, bevor sie startet, aber weil Motherboards universell gefertigt werden, für eine ganze Reihe von unterschiedlichen CPUs, sind keine CPU-Daten in den internen Tabellen vorhanden, oder eine Art Allgemeingültigkeit, die in bestimmten Fällen nicht stimmt. Clover korrigiert das.

Es übernimmt die vollständige Erkennung der installierten CPU und nimmt die notwendigen Korrekturen in den Tabellen vor, sowie in der CPU selbst. Als ein Ergebnis wird der Turbo-Modus aktiviert;

- Das BIOS nutzt irgendwie den Computerspeicher, teilt ihn in Regionen mit einem bestimmten Zwecke, aber oft ist dies nicht in der Art und Weise getan, die für den Betrieb von macOS erforderlich ist; Die Anpassung der Speicherabbildung und andere damit verbundene Kleinigkeiten werden vom Treiber erledigt OsxAptioFixDrv und seine Abkömmlinge, das aus verschiedenen Gründen von Clover getrennt ist, in ein separates Modul in erster Linie, um das Laden von Legacy-Dateien nicht zu beeinträchtigen;

- noch eine Kleinigkeit. Die Sourcen von DUET und dem gesamten EDK2 sind durchgängig geschrieben für unterschiedlichem Hardware....., aber die Abhängigkeit von der Hardware wird durch Konstanten hergestellt. Das heißt, es wird davon ausgegangen, dass der Benutzer Duet für eine bestimmte Konfiguration betreibt. Das Ziel von Clover ist es, universell zu sein, mit automatischer Erkennung der Plattform.

All dies geschieht automatisch, und ein Anfänger kann Clover sogar ohne Verständnis für die genannten Probleme nutzen. Nun, für den fortgeschrittenen Benutzer von Clover bietet sich die Möglichkeit, viele Parameter manuell zu ändern. Ihre Berücksichtigung ist der Zweck dieses Buches. Übrigens, manche Menschen beginnen ihre Bekanntschaft mit Hackintosh vom Studieren der Konfiguration. Das ist der falsche Ansatz. Sie reparieren die Konfiguration nachdem Sie versucht haben, etwas herunterzuladen. Dazu später mehr.

Was ist was?

Noch einmal in aller Kürze: Clover erfüllt vier grundlegende Funktionen:

1. Fängt die Steuerung ab, damit nicht das BIOS das Betriebssystem lädt, sondern damit es über Clover durchgeführt wird. Das BIOS ruft Clover auf und es lädt das Betriebssystem, übrigens immer noch und fragt es den Benutzer, welches von mehreren Betriebssystemen er laden soll.

2. Ändert die Daten, die das BIOS an das Betriebssystem sendet, dies ist grundlegend eine Voraussetzung für das Starten von MacOS. Zum Beispiel die Seriennummer.

3. Modifiziert das MacOS-System selbst, um auf dieser Hardware zu laufen.

4. In einigen Fällen korrigiert Clover den Zustand der Geräte, die nicht über das BIOS laufen.

Beispiele: ResetHDA, RadeonDelnit, HaltEnable.

Windows und Linux benötigen Punkt 3 nicht, und den zweiten nur selten. Nun aber zum ersten Punkt da hier Klärungsbedarf besteht, da Neueinsteiger mit der Frage "Warum lädt es nicht" beginnen. Wenn Sie den Computer einschalten oder neu starten, bootet das Betriebssystem mit Hilfe von Clover. Das erfolgt auf folgende Weise:

Option A. BIOS-basierter Computer (Legacy-Boot)

BIOS → MBR → PBR → boot → CLOVERX64.efi → OSloader (boot.efi im Falle von MacOSX, bootmgfw.efi für Windows, grub.efi für Linux).

Du verstehst es nicht, oder? In Worten:

1. Wenn der Computer eingeschaltet wird, startet das BIOS. Er wählt entsprechend seiner Einstellungen aus, von welcher Disk gestartet wird, wenn Legacy-Boot ausgeführt wird.

2. das BIOS liest einen Nullsektor von diesem Laufwerk. Der Sektor wird als MBR bezeichnet. Codes werden in den Speicher geschrieben und das BIOS überträgt die Kontrolle an den miniloader .

3. Ein in den MBR eingebettetes Programm sucht auf dieser Platte nach einer Boot-Partition, liest von dort den ersten Sektor in den Speicher und übergibt ihn an die Steuerodes. Dies wird als PKR-Sektor bezeichnet.

4. Ein in den PBR eingebettetes Programm sucht in seinem Dateisystem nach einer Datei namens boot System, lädt es in den Speicher und überträgt die Steuerung an es. In unserem Fall in die Datei ist es boot DUET, bzw. CloverEFI, die von Clover bereit gestellt wird.

5. CloverEFI sucht nach der Datei CloverX64.efi, lädt sie und überträgt die Steuerung an sie.

Dann geht es weiter wie in Option B (nächste Seite):

Option B. Computer basierend auf UEFI-BIOS (neues Schema, UEFI-Booter)
UEFI_BIOS → CLOVERX64.efi → OSloader (oder BootX64.efi)

1-5. Wenn das BIOS UEFI ist, kann es das EFI-Startmodul in der EFI-Partition finden. Der Standardname einer solchen Datei sollte EFI/BOOT/BOOTX64.EFI sein. In einigen BIOSen wird der Name EFI/microsoft/boot/bootmgfw.efi geflasht. Und wir wissen, wie man dem BIOS beibringt, sofort EFI/CLOVER/CLOVERX64.EFI-Datei zu laden. Aber Sie müssen trotzdem lernen, wie man es selbst macht. Für diejenigen, die kann, wird die Datei EFI/BOOT/BOOTX64.EFI ebenfalls von Clover gefälscht.
6. Endlich wird CLOVERX64.EFI geladen, und wir sehen die Auswahl der Betriebssysteme und können eine von ihnen zu laden.

Damit dies alles funktioniert, müssen folgende Dateien an den vorgeschriebenen Orten sein:

MBR-Bereich

Nullblock des externen Mediums, von dem gebootet wird (HDD, SSD, USB-Stick, USB HDD, DVD). In diesen Block sollen die ersten 440 Bytes einer der Optionen geschrieben werden:

boot0af (Active First) - seine Aufgabe ist es, die aktive Partition in der MBR-Festplattenpartitionierung zu finden und das Management zu seinem PBR-Bereich übertragen. Ein hybrides MBR/GPT-Partitionsschema ist möglich.

Wenn die Partitionierung rein GPT ist, d. h. es gibt keine aktiven Partitionen, wird die Verwaltung auf eine EFI-Partition, die durch GUID=C12A7328-F81F-11D2-BA4B-00A0C93EC93B erkannt wird. Wenn Sie diese Partition plötzlich neu formatieren, prüfen Sie, was die jetzt GUID. Es gibt eine Variante, bei der die erste Partition mit FAT32 richtig ist, aber das Booten nicht funktioniert, weil seine Signatur 0C00 ist und EF00 lauten sollte. Dies wird mit dem Dienstprogramm gdisk behoben.

boot0ss (Scan Signature) - Suche nach der ersten Partition mit 0xAF-Signatur, d. h. HFS+-Partition mit installierten OSX, und übertragen Sie die Kontrolle an dessen PBR. Auf diese Weise ist es möglich, die System von einer HFS+-Partition auf einer GPT-partitionierten Festplatte, aber nur von der ersten solchen Partition. Wenn eine solche Partition nicht gefunden wird, wird weiter nach FAT32- oder ExFAT-Partitionen gesucht, in dieser Reihenfolge.

Das System ist nicht in der Lage, die Unterschrift zu prüfen, und übergibt die Kontrolle an sie.

boot0ab - Suche nach Partition mit 0xAB-Signatur - Apple Boot Partition. Das ist Recovery. Für Perverse. Realistisch betrachtet, hat Recovery nie dazu gedient, ein Vermächtnis zu hinterlassen.

boot0md ist ein Kombinationsfeld, das nach einer HFS+-Partition über mehrere Festplatten hinweg sucht, anstatt nur durch den Hauptschalter. Seltsame Option, nicht in den Installationsoptionen enthalten.

PBR-Bereich die ersten Blöcke jeder Partition auf dem ausgewählten Medium. Hier wird die Bootloader-Phase geschrieben. Dieser Bootloader kennt das Dateisystem seiner Partition und ist in der Lage, dort eine Datei mit Namen Boot zu finden, um ihn zu laden und die Steuerung an ihn zu übertragen. Entsprechend, gibt es Varianten für verschiedene Dateisysteme.

boot1h2 - für HFS+-Dateisystem und unterstützt Bootdateien mit einer Länge von bis zu 472kb. Alt die *boot1h*-Variante, die mit dem Bootloader "Chameleon" verteilt wird, unterstützt nur 440kb Datei (Sie benötigen 472). **Für diejenigen, die Chameleon zuerst installiert haben, erinnere ich Sie noch einmal: Sektor PBR muss mit der boot1h-Datei aus dem Clover-Bausatz aktualisiert werden, sonst wird es nicht starten.**

Die Variante *boot1h2* hat eine Pause von 2 Sekunden zum Umschalten des Bootloaders. (B Chameleone, ich glaube, sie haben es endlich repariert.)

boot1h - auch, aber ohne Pause.

boot1f32alt - für FAT32-Dateisystem. Dieses Dateisystem hat Schreibunterstützung, Es ist also sehr praktisch, um einen Bootloader darauf zu installieren. Sie können eine EFI-Partition verwenden, sie können das Flash-Laufwerk so verwenden, wie es ist, da Flash-Laufwerke bereits mit FAT32 formatiert verkauft werden. *boot1f32alt* macht eine Pause von 2 sec.

(2): Aus historischen Gründen wird empfohlen, jedes Flash-Laufwerk neu zu formatieren, und zwar unbedingt in Windows-System. Diese Empfehlung ist inzwischen veraltet und entbehrt jeglicher Grundlage.

boot1f32 - Variante ohne Pause.

Warnung! Das Dateisystem muss FAT32 sein, nicht MSDOS, da FAT16 ist inakzeptabel. Der Bootloader funktioniert darauf nicht!

boot1x - Zenith432 hat einen Bootsektor erstellt, der für das exFAT-Dateisystem ausgelegt ist. Leider ist der Treiber im Clover nur ReadOnly, trotzdem ist dieses Dateisystem sehr interessant für externe Medien, da es Dateien größer als 4 GB unterstützt, und gleichzeitig sowohl von Mac als auch von Windows unterstützt wird. Diese Sektoren (genauer gesagt Phase-2-Lader) haben eine weitere nützliche Funktion. Sie sind haben eine anfängliche Startverzögerung von zwei Sekunden und warten auf Tastatureingaben.

Die eingegebene Ziffer wird an den Dateinamen angehängt, z. B. durch Drücken der Taste 1 bei schwarzem Bildschirm ganz am Anfang des Bootens (nach der BIOS-Meldung "booting from...") laden wir die Datei boot1, Drücken Sie 3, um boot3 zu laden, drücken Sie 6, um boot6 zu laden. Die Bedeutung ist, verschiedene Varianten von Ladern an einem Ort zu halten, oder sogar verschiedene Lader, indem Sie ihnen einfach unterschiedliche Nummern geben. Zum Beispiel:

boot - Clover, aktuelle Version, oder Testversion

boot1 - Chamäleon

boot3 - Clover-32bit, getestete, funktionierende Version (zeigt Nummer 3 an)

boot4 - Clover-32bit, mit BiosBlockIO-Treiber

boot5 - Clover-64bit, gekürzte Größe, um EBDA nicht zu überlappen (siehe unten)

boot6 - Clover-64bit, getestete, funktionierende Version (zeigt Nummer 6 an)

boot7 - Clover-64bit mit BiosBlockIO-Treiber, der mit beliebigen Controllern funktioniert, die vom BIOS unterstützt werden. (zeigt den Buchstaben B oder L oder 7 an)

Zusätzlich zu diesen Optionen kann der PBR-Sektor einen Windows-Bootmanager enthalten, der das NTFS-Dateisystem kennt; GRUB, der das EXT4-Dateisystem kennt; und andere, die nicht mit Clover zu tun haben. Zumindest an dieser Stelle.

Boot oder CloverEFI

Im Fall von Chameleon ist die "Boot"-Datei der gesamte Bootloader. Im Fall von Clover ist in dieser Datei das gesamte EFI-System sowie den Boot-Dienst zur Übergabe der Kontrolle an den nächsten Bühne enthalten. All dies sollte bei Option B als bereits im ROM des Computers vorhanden betrachtet werden. Es stellt sich tatsächlich heraus, dass nicht alles vorhanden ist, und einige Teile (sogenannte Treiber) zusätzlich heruntergeladen werden sollten. Bereits montierte Teile zur Boot-Datei Variante A. Im Gegensatz zu den vorherigen Stufen ist die Boot-Datei bereits in der Bitrate unterschiedlich, d. h. Einzelne Optionen für 32- und 64-Bit-Boot. In den meisten Fällen sollten Sie die 64-Bit-Variante wählen

Option, wenn der Prozessor diesen Befehlssatz unterstützt (Achtung! Pentium 4 und Yonah - Nur 32 Bit). Wenn man jedoch aus irgendeinem Grund nur mit dem 32-Bit-Betriebssystem arbeiten möchte, dann macht es Sinn, EFI32 zu laden. Er ist 20 % kleiner und dementsprechend schneller. Leider nicht kompatibel mit Windows 7 EFI, das immer 64bit ist.

November 2016:

Auf Wiedersehen, Clover-32! Ich bin nicht mehr daran interessiert, es zu unterstützen.

Im Wesentlichen handelt es sich bei der Boot-Datei um einen modifizierten DUET Datei. Der Anteil der inhaltlichen Korrekturen liegt wahrscheinlich bei weniger als 1 %, aber dieser Prozentsatz führt zum Erfolg. Der kardinale Unterschied zu Duett ist, dass Clover für den Zweck arbeitet, für den es bestimmt ist. Wenn jemand denkt, dass ich seit drei Jahren Nnia mache, DUET editiere und das ist genug nehmen Sie Vanille und fügen Sie AppleSim hinzu, dann bon voyage! Das ist es noch lange nicht, aber es ist nicht mein Ziel, alle Feinheiten dieser Entwicklung zu beschreiben. Es ist bereits geschehen. In der Zukunft nennen wir dieses Programm generisch CloverEFI. Mit jeder seiner binären Revisionen sind unterschiedlich, da sich EDK2 und manchmal auch die Quellen ändern, z.B. die letzten Änderungen waren vom Typ StrCpy => StrCpyS, sozusagen eine sicherere Version. Allerdings ist die Funktionalität CloverEFI und hat sich seit langer Zeit nicht mehr verändert, wenn Sie also ein Problem mit dem neuen

Version haben, setzen Sie die Version ein, die bei Ihnen funktioniert hat. Also in dem Bausatz, der offiziell auf SF.NET verteilt wurde, enthielt die alte Datei boot7-MCP79, für einige Der Grund, warum das neue boot7 nicht mehr mit diesem Chipsatz funktioniert.

Ein weiteres kleines Problem. Der PBR sucht nach einer Boot-Datei. Aber das Windows-Installationsprogramm hat einen Ordner mit mit diesem Namen, auch in der Wurzel. Ich muss meine Boot-Datei in boot5 umbenennen, um einen alternativen Bootvorgang zu verwenden. Irgendwo habe ich eine Version von boot1f32 erstellt, die durch default lädt boot5 sofort, ein Byte fix. Diese Version ist jedoch nicht vorhanden in der Standard-Installation.

CLOVERIA32.efi und CLOVERX64.efi oder CloverGUI

Diese Datei, in zwei Versionen mit unterschiedlichen Bitraten, ist eine grafische Bootloader-Shell zur Auswahl des Betriebssystems, zur Installation zusätzlicher Optionen, zum Laden zusätzlicher Treiber und zum tatsächlichen Starten des Betriebssystems. Grafiken und die Menüs basieren ursprünglich auf dem rEFIt3-Projekt, was sich im Verzeichnisnamen und im Menü Über zeigt. Im Moment macht der Originalteil kaum 5% des gesamten Programms aus, und sogar das in einer korrigierten Form. Und jetzt sind sowohl die Grafiken als auch die Menüs auf Klassen umgeschrieben C++, so dass es Erinnerungen an die ursprünglichen Codes gibt.

Nachfolgend wird dieses Programm allgemein als CloverGUI bezeichnet.

Daddy's work

Struktur des Ordners: Außerdem benötigt der Downloader zusätzliche Dateien, die Ordnerstruktur sieht ungefähr folgendermaßen aus: (Clover 5218):

Name	Änderungsdatum	Größe	Art
BOOT	04.10.2020, 06:32	--	Ordner
BOOTX64.efi	07.01.2021, 08:57	1,3 MB	Dokument
CLOVER	09.01.2021, 13:29	--	Ordner
ACPI	27.01.2021, 12:53	--	Ordner
CLOVERX64.efi	07.01.2021, 08:57	1,3 MB	Dokument
config-sample.plist	13.12.2020, 09:16	27 KB	Clover...cumen
doc	04.10.2020, 06:28	--	Ordner
drivers	27.01.2021, 12:54	--	Ordner
BIOS	07.01.2021, 08:57	--	Ordner
AudioDxe.efi	07.01.2021, 08:57	39 KB	Dokument
FSInject.efi	07.01.2021, 08:57	21 KB	Dokument
SMCHelper.efi	07.01.2021, 08:57	3 KB	Dokument
XhciDxe.efi	07.01.2021, 08:57	27 KB	Dokument
off	27.01.2021, 12:54	--	Ordner
BIOS	27.01.2021, 12:54	--	Ordner
UEFI	07.01.2021, 08:57	--	Ordner
UEFI	07.01.2021, 08:57	--	Ordner
AudioDxe.efi	07.01.2021, 08:57	39 KB	Dokument
DataHubDxe.efi	07.01.2021, 08:57	3 KB	Dokument
FSInject.efi	07.01.2021, 08:57	21 KB	Dokument
SMCHelper.efi	07.01.2021, 08:57	3 KB	Dokument
kexts	09.01.2021, 13:29	--	Ordner
10.11	04.10.2020, 06:28	--	Ordner
10.12	04.10.2020, 06:28	--	Ordner
10.13	04.10.2020, 06:28	--	Ordner
10.14	04.10.2020, 06:28	--	Ordner
10.15	04.10.2020, 06:28	--	Ordner
Off	04.10.2020, 06:28	--	Ordner
Other	04.10.2020, 06:28	--	Ordner
misc	04.10.2020, 06:28	--	Ordner
OEM	27.01.2021, 12:54	--	Ordner
ROM	04.10.2020, 06:28	--	Ordner
themes	Heute, 12:24	--	Ordner
embedded	04.10.2020, 06:28	--	Ordner
logo_main.png	04.10.2020, 06:28	13 KB	PNG-Bild
pointer-metal.png	04.10.2020, 06:28	4 KB	PNG-Bild
random	04.10.2020, 06:28	--	Ordner
tools	04.10.2020, 06:32	--	Ordner

(3) Oft wird gefragt: "Warum nicht rEFInd? Ich antworte: "Weil es später kam als Clover erschien".

(Neue Seite:)

Das heißt, die Datei CLOVERX64.efi muss sich unter /EFI/CLOVER/ befinden, und die Schriftart BoG_LucidaConsole_10W_NA.png im Ordner /EFI/CLOVER/themes/black_green/. Realistischerweise sind diese und auch andere Ordner stärker mit Inhalten bestückt. Im Verlauf der Beschreibungen werden wir detaillierter aufzeigen, was wofür verwendet wird.

Ein paar Worte zum Ordner /EFI/CLOVER/OEM/

Der Ordner dient dazu, Boot-Optionen für verschiedene Konfigurationen zu speichern. Typisch ist die Situation, wenn wir ein bootfähiges Flash-Laufwerk erstellen und darauf gibt es zusätzlich zur allgemeinen Config /EFI/CLOVER/config.plist, auch die gut abgestimmte /EFI/CLOVER/OEM/Inspiron 1525/ config.plist und /EFI/CLOVER/OEM/H61M-S1/UEFI/config.plist, sowie deren ausgearbeitete DSDT.aml, unterschiedlich für verschiedene Computer, und verschiedene Sätze von Kexte.

Der Ordnername wird vom SMBIOS ermittelt und Sie können im boot.log genau sehen, wie Ihren Computer heißt:

```
0:100 0:000 Running on: 'Z170X-UD5 TH' mit Board 'Z170X-UD5 TH-CF'
```

Die erste Zeile ist der Name des gebooteten Systems, typisch für Laptops, auf Desktops gibt aber etwas Abstraktes. In der zweiten Reihe befindet sich das Motherboard-Modell, das für Desktops geeignet ist, aber nicht für Laptops. Beide Namen sind für den Namen Ihres Ordners in Ordnung, wählen Sie den mehr verständlich.

Sie können auch einen EFI-Ordner in Ihrem Ordner haben, um verschiedene Konfigurationen für EFI zu haben (Option B) und für Legacy-Downloads (Option A) auf demselben Computer (obwohl ich persönlich Ich bezweifle, dass es jemand braucht).

EFI-Treiber

Separat erwähne ich die Ordner drivers32, drivers64, drivers64UEFI, jeweils für 32, für 64-Bit-Boot mit BIOS-Boot-Option A und für EFI-Boot mit Option B. Die Zusammensetzung dieser Ordner haben für verschiedene Versionen des BIOS sowie für verschiedene Konfigurationen unterschiedlich Abschnitte.

Warnung. Die Ordnerstruktur in Revision 5000 hat sich geändert, sie hat jetzt einen Treiberordner, mit den Unterordnern BIOS und EFI. 32Bit ist für immer verschwunden.

Es ist zu beachten, dass diese Treiber nur für den Zeitraum des Bootloaders gültig sind. Sie beeinflussen nicht das geladene Betriebssystem, außer indirekt, durch die Art und Weise wie Geräte initialisiert werden (vierte Clover-Funktion).

Was soll in diesen Ordnern abgelegt werden? Auswahl durch den Benutzer.

- **NTFS.efi** - NTFS-Dateisystemtreiber, um Windows EFI booten zu können, Allerdings.es scheint das der nicht besonders notwendig ist. Der ESP ist auch der Ort, an dem sich der Windows-Bootloader befindet, auf FAT32.

- **HFSPlus.efi** - HFS+-Dateisystemtreiber, erforderlich zum Betrieb von MacOSX.

Erforderlich für Variante B, aber in A ist es bereits in der Boot-Datei vorhanden. Leider, Diese Datei ist proprietär und kann in einem OpenSource-Projekt nicht vorhanden sein, daher ->

- **VboxHFS.efi** ist eine legale Alternative für HFSPlus.efi und unterscheidet sich durch etwas geringere Ausführungsgeschwindigkeit. Die neue Version unterstützt Links, sogar mehr als die native Version Apples HFSPlus.efi. HardLink, SymLink werden unterstützt ! Während HFSPlus.efi nur Hardlinks. In der modernen Version von Clover ersetzt die heruntergeladene Datei die eingebettete. Das heißt, wenn eine eingebettete VboxHFS.efi in der Boot-Datei vorhanden ist, und drivers/BIOS/ die Datei HFSPlus.efi hat, dann wird letztere funktionieren.

- **VBoxExt2.efi** - EXT2/3-Dateisystemtreiber, wird benötigt, um Linux EFI auszuführen. Ähnlich VboxExt4.efi. Wiederum, wenn Grub nicht in der EFI-Partition liegt.

- **Fat.efi** ist ein FAT32-Dateisystemtreiber. Natürlich ist sie bereits vorhanden. Allerdings, AMI EFI BIOS enthält einen sehr fehlerhaften Treiber, daher habe ich mich entschieden, meine Option zu verwenden.

- **FSInject.efi** - Dateisystem-Hook-Treiber zum Ermöglichen die Injektion einiger Kexte in das System zu erzwingen, es sei denn, das System selbst hält es für notwendig. Schwierig zu verstehen? Wir werden auf diese Frage später zurückkommen, wenn werden wir den ForceKextsToLoad-Schlüssel berücksichtigen

- **PartitionDxe.efi** - tatsächlich gibt es einen solchen Treiber in CloverEFI, und in EFI ist es, aber nur ist es weder für Apple-Partition noch für MBR/GPT-Hybrid ausgelegt. Fazit: In Option B benötigt manchmal einen Driver. Wenn Sie ein normales GPT haben, können Sie damit durchkommen.

- **OsxFatBinaryDrv.efi** - notwendiger Treiber für Variante B, sorgt für den Start der Fat-Module, wie z. B. boot.efi auf Systemen vor 10.9. Auf neueren Systemen Es geht es auch ohne.

- **OsxAptioFixDrv.efi** ist ein spezieller Treiber für die Korrektur von Ram-Speicherkarten, die das AMI AptioEFI- erstellt, sonst kann das Betriebssystem nicht gestartet werden.

- **OsxAptioFix2Drv.efi** ist eine leicht modifizierte Version. Es stellte sich heraus, dass damit Folgendes möglich ist Hibernaten im System 10.9.1 bei EFI-Boot! Aber leider ist diese Option nicht in der Lage, 10.7.5. zu booten

- **OsxAptioFix3Drv.efi** - neue Version korrigiert von vit9696 zur Unterstützung nativen NVRAM, mit der können Sie nun auf dessen Emulation verzichten!

- **AptioMemoryFix.efi** ist ein komplett neu entwickelter Treiber aus dieser Serie, Autor vit9696, mit zusätzlicher automatischer Suche nach dem besten Wert für slide=xxx. Im Moment wird empfohlen, es an erster Stelle zu verwenden

- **OsxLowMemFix.efi** - vereinfachte Version von AptioFix, geeignet für einige fremde UEFI-BIOS-Varianten (Insyde H2O).

- Diese fünf Aptio-Optionen werden nicht gleichzeitig verwendet, auch wenn alle vorhanden sind, wählt Clover einen von ihnen nach Priorität aus: **AptioMemoryFix**, **OsxAptioFix3Drv.efi**, **OsxAptioFix2Drv.efi**, **OsxAptioFixDrv.efi** und zu letzt **OsxLowMemFix.efi**.

- Für die neue Revision des Clover 5120 wird empfohlen, alle diese Treiber zu vergessen *Fix*, und die Menge der OcQuirks und OpenRuntime zu verwenden, die jetzt im Paket enthalten sind Clover, und ab 5122 wird OcQuirks nicht mehr benötigt, es ist in Clover enthalten;

- **NvmExpressDxe** - NvmExpress-Controller-Treiber, der als SATA-Ersatz für SSD-Laufwerke dient. Sie sollten bedenken, dass, wenn der Controller auf Motherboard ist, wird dieser Treiber höchstwahrscheinlich bereits im UEFI-BIOS vorhanden sein. Sie benötigen es für zusätzlicher Controller.

- **Usb*.efi**, **UHCI.efi**, **EHCI.efi**, **XHCI.efi**, **OHCI.efi** - USB-Treibersatz, für die Option B Fälle, in denen die eingebauten Treiber aus irgendeinem Grund nicht gut funktionieren. Warum sollte Ich dennoch auf einmal benutzen? Vielleicht gibt es eine Art Verbindung zu anderen Funktionen, die zu deaktivieren sind.

- **PS2Mouse...**, **PS2MouseAbsolute...**, **UsbMouse...** - Satz von Treibern zur Unterstützung der Maus/Trackpad/Touchpad-Zeiger in CloverGUI. Bei der Bedienung des Systems, haben diese Treiber keine Wirkung. Das UEFI-BIOS sollt ebenfalls einen solchen Treiber haben, Sie sollten ihm nicht in die Quere kommen.

- **AptioInputFix** - auch für Maus und Tastatur, wenn die UEFI-BIOS-Option richtig funktioniert . Autor vit9696.

- **DataHubDxe.efi** - dieser Treiber ist bereits in Option A vorhanden, und zwar möglicherweise, ist auch sie in UEFI verfügbar. Aber falls es nicht da ist, lohnt es sich, sie extern zu booten. Es wird keinen Konflikt entstehen, aber es wird sicher sein, dass es einen gibt. Mehr und mehr sehe ich Beschwerden von Benutzern, dass etwas nicht geladen wird oder für sie nicht funktioniert. Und sie haben keine DataHubDxe! Woher haben Sie all diese Gerissenheit? Hat man Sie auf tonimac davon überzeugt, dass dieser Treiber extra ist? Dann kommen Sie auch nicht damit zu mir.

- **CsmVideoDxe.efi** - Videotreiber, der mehr Reichweite bietet als die eingebaute UEFI-Bildschirmgröße wird für Option B benötigt, wenn die Grafikkarte kein UEFI VideoBIOS hat. Es gibt Varianten, bei denen dieser Treiber nicht kompatibel damit ist mit. Wenn Sie eine Videokarte, z. B. eine eingebettete Intel-Grafikkarte, verwenden, seien Sie vorsichtig. Der Treiber benötigt die Aktivierung des Kontrollkästchens CSM Enable im BIOS.

- **GrubNTFS**, **GrubEXFAT**, **GrubHFSPLUS**, **GrubUDF...** - Satz von Treibern für die meisten verschiedene Dateisysteme, die von GRUB-Quellen angepasst wurden, um in als Teil von Clover zu arbeiten. Es ist sehr ermutigend, dass wir jetzt Unterstützung für all diese dateibasierten Systeme, und lizenztechnisch sauber, haben. Dank an AnV, der die folgenden Informationen beschafft und angepasst hat Quellen. Sie haben einige Vorteile gegenüber unseren Drivern, zum Beispiel GrubHFS unterstützt komprimierte Volumes, aber der Status des Ganzen ist eher "Beta". Sie sind langsam und fehlerhaft. Und der Funktionsumfang ist nicht beeindruckend. HFS - keine Unterstützung, UDF - nicht alle Hider sind lesbar, EXFAT - kein Schreiben (ich wünschte, es wäre so!).

- **AppleImageCodec.efi** - Decoder für PNG- und BMP-Dateien, benötigt für FileVault2.

- **AppleKeyAggregator.efi** - erstellt ein spezielles Protokoll für die Eingabe eines Passworts in der FileVault2-Schnittstelle.

- **AppleUITheme.efi** - erstellt das Protokoll, mit dem FileVault2 den Hintergrund zeichnet auf dem Bildschirmbild (in Sierra sind es z. B. unscharfe Berge).

- **AppleKeyFeeder.efi** - erforderlich für die Eingabe eines Kennworts in der FileVault2-Schnittstelle, wenn eine PS2-Tastatur verfügbar ist. Es gibt eine Alternative dazu **AptioInputFix.efi** von vit9696.

- **FirmwareVolume.efi** - erzeugt das FirmwareVolume-Protokoll, das jedoch nur in FileVault2 verwendet wird, das erwartet, dort ein Bild zu finden den Cursor. Ja. Es ist da.

- **SMCHelper.efi** - erzeugt das AppleSMCProtocol, das folgende Eigenschaften haben sollte Der SMC-Chip kommuniziert mit dem SMC-Chip und erhält von dort Peripherie-Statusinformationen. Wir haben einen Emulator erstellt, der die Werte zurückgibt, die von ihm erwartet werden. Gefüllt mit echten Inhalten ist sowohl Clover selbst als auch FakeSMC, das speichert Tasten im NVRAM, wenn FakeSMC mit dieser Funktion installiert ist. Alle diese wird vor allem für FileVault2 benötigt, hat aber auch Auswirkungen auf das System und ohne sie ist keine Tastenrückgabe an das System vorgesehen, da es sich um Zweifel an der Durchführbarkeit dieser Maßnahme. Es gibt eine externe Alternative zu VirtualSMC von Mit Ihren Gesetzen.

- **HashServiceFix.efi** - Erzeugt HashService-Gruppenprotokolle, wenn UEFI sie nicht hat. FileVault2 interagiert mit ihnen, wie Sie in den Protokollen sehen können. Aber den visuellen Effekt habe ich nicht sägen.

- **APFS.efi** ist ein neuer Dateisystemtreiber, der in 10.13 eingeführt wurde. hat kein Open-Source-Analogon, daher verwenden wir das, was in der installierten Apple-System unter /usr/standalone/i386/apfs.efi, oder → - ApfsDriverLoader ist eine Open-Source-Alternative zum APFS-Treiber. Dank an savvas und andere zu Mitgliedern der Acidanthera-Gruppe. Seine Essenz ist, den richtigen APFS-Treiber zu laden einfach aus dem Container und hat somit immer die richtige Version. Wenn jedoch wir booten das alte System, wir bekommen den alten Treiber und können nicht lesen APFS-Partition mit dem neuen System. Hier geht es mehr um apfs.kext, aber das Problem ist ähnlich.

- **AudioDxe** ist ein von Goldfish64 entwickelter Soundtreiber. Es kann verwendet werden für um den Ton beim Hochfahren des Computers abzuspielen. Es ist jedoch eine Verschönerung, wenn ein solcher Ton nicht benötigt wird, können Sie ohne den Treiber leben. Es wird ein Fehler bemerkt, dass danach in Wenn der Ton auf dem System selbst fehlt, dann deinstallieren Sie einfach diesen Treiber.

Laden der Kexte

In der Ordnerstruktur sehen wir einen gemeinsamen Ordner /EFI/CLOVER/kexts/ mit den Ordnern 10.7/, 10.11/ und so weiter, sowie Ordner mit den Namen OFF/ und OTHER/. Schauen wir uns das Thema einmal genauer an. "Kernel Extensions" für das Betriebssystem MacOSX, die meisten davon sind Treiber für irgendeine Art von Gerät. Oder ein Pseudo-Treiber eines nicht existierenden Geräte wie FakeSMC. Wir laden sie herunter, weil ein sauberes System von Apple einfach nicht unsere Geräte unterstützt, oder wir wollen neue Funktionen einführen. Wir bezieht uns hier speziell auf MacOSX, für Linux und Windows werden keine Kexte geladen.

Ab der Revision 3281 werden zuerst alle Kexte aus dem Ordner Others/ geladen und dann aus Ordner mit einer bestimmten Betriebssystemnummer. Dies ist sinnvoll, weil die meisten nicht an die Systemversion gebunden sind und sie funktionieren mit jeder Version gleichermaßen, FakeSMC im Besonderen. Wenn der Treiber geladen ist, werden möglicherweise bestimmte Ordner benötigt nur für ein bestimmtes System zu unterstützen. AppleHDA, zum Beispiel, ist auf jedem System anders (Ich weiß es nicht, tatsächlich, laden sie es mit Clover). Für mich macht also nur FakeSMC Sinn, den Rest kopieren Sie in den System-Ordner, und dann auch nur zum Zeitpunkt der Installation, denn im Installationsprogramm werden alle Caches gepackt sein, und es ist unmöglich, etwas Eigenes hineinzuschieben (obwohl... es scheint, dass OpenCore kann es). Aber da die Leute alle Kext-Sets mit Clover laden wollen, soll es eine solche Möglichkeit geben.

Optionen:

Erstens, wenn Sie das Menü manuell schreiben (Benutzerdefiniert → Einträge), können Sie für die einzelnen Elemente InjectKexts=Yes/No/Detect beschreiben.

Zweitens: Es gibt eine gemeinsame Einrichtung für alle:

```
<key>SystemParameter</key>
<dict>
    <key>InjectKexts</key>
    <string>yes</string>.
```

Detect ist veraltet und wird nicht mehr verwendet, und wir wollen immer das Kexte injizieren werden.

Ein weiterer Punkt. Im Captain gibt es eine zusätzliche Prüfung für unsignierte Kexte. Um sie zu überwinden, muss der SIP (System Integrity Protection) ausgeschaltet werden. Das erfolgt in der Config im Schlüssel:

```
CsrActiveConfig
<key>RtVariablen</key>
<dict>
    <key>CsrActiveConfig</key>
    <string>0x3F7</string>
```

Nur Bit 0x1 (ALLOW_UNSIGNED_KEXTS) spielt eine Rolle, wenn Sie unsigned die Kexte haben. Ich habe alle Kexte signiert und kann sie mit Null laden.

Ab Revision 4233 ist es im Clover-Menü nun möglich, das Laden einzelne Kexts aus den Ordnern /EFI/CLOVER/kexts/* zu blockieren. Auf diese Weise können Sie dort neue, fragwürdige Kexte ablegen, und versuchen Sie zu booten. Wenn es nicht klappt, können sie ihn beim Neustart mit diesem Kontrollkästchen einfach sperren. Und ab der Revision 5052 können Sie den fraglichen Kext in den Ordner /EFI/CLOVER/kexts/Off/ legen, z. B. VoodooPS2 von Vasya Pupkin, der verspricht, dass sein Kext besser ist als Ihrer. Und Ihr Kext befindet sich im Ordner OTHER. Nachdem Sie das Clover-Menü geladen haben, drücken Sie die Leertaste, und Sie sehen das -Menü um den Kext zu deaktivieren.

Der Standardwert, der für den Ordner "OFF" nicht zulässig ist, und der Standardwert, der für den Ordner "OTHER" zulässig ist, lautet standardmäßig: Sie aktivieren dieses neue VoodooPS2 testweise und deaktivieren das alte. Wenn das nichts funktioniert, dann funktioniert der alte wieder, wenn Sie neu starten. Wenn alles in Ordnung ist, und Vasya richtig liegt, dann ändern Sie manuell, wo der Cache liegen soll.

Entwicklung

Das Projekt hat aus lizenzrechtlichen Gründen keine kommerzielle Relevanz, und auch ist es zu groß, um es alleine zu machen, daher ist die sinnvollste Lösung es mit open source zu machen, und lassen Sie jeden, der etwas beitragen möchte, mitmachen. Bis 2019 Jahr basierte das Projekt auf dem sf.net-Server im Repository <http://cloverefiboot.sourceforge.net/>

Mit Version 2.5k sind wir in das neue Repository unter <https://github.com/CloverHackyColor/CloverBootloader> Es gibt auch kompilierte Versionen, die sofort einsatzbereit sind:

<https://github.com/CloverHackyColor/CloverBootloader/releases>

Eine lyrische Abschweifung. Die Erstellung eines Projekts, und sogar eines so großen, erfordert harte Arbeit zu den folgenden Punkten:

- Sammlung von Dokumentationen, Datenblättern, Spezifikationen, Softwaremustern für das gelieferte die Aufgabe, und weitere Informationen über die Hacker finden. Ich hatte eine guten Start Punkt - **Chameleon**, bei dem "alles" funktioniert (allerdings in Anführungszeichen!). Stimmt, die Zeit geht weiter, und neue Prozessoren, neue Grafikkarten, neue Anforderungen für neue Versionen von OSX, und wir müssen erneut nach Beschreibungen suchen oder neue Tests durchführen.

Der Ausgangspunkt wurde von Kabyl systematisiert, verbeugen Sie sich zunächst vor ihm.

4 Open-Source-Projekte haben dennoch einige Einschränkungen in der Nutzung, um es so auszudrücken "Lizenzierung". Die GPL-Lizenz ist lästig, Intel empfiehlt ihre Verwendung nicht. Clover wird als BSD geliefert.

- Wenn es eine Problemstellung gibt, Eingabedaten und was man am Ausgang erhalten möchte, ist es erforderlich, einen Algorithmus zu schreiben, möglichst kompakt und schnell, vorzugsweise fehlerfrei und sicher. Dies ist die Art von Aufgabe, die Programmierer lieben, und die meisten Bögen in diesem Projekt werden an solche Assistenten vergeben. Ich möchte vor allem um Dmazar und Geek hervorzuheben, sie haben in den frühen Tagen einige wirklich harte Sachen für die Entwicklung des Bootloaders gemacht.
- Es ist Zeit für die mühsame und anstrengende, aber **unkomplizierte** (?) Arbeit des Schreibens von tausenden von Codezeilen, bei denen man nicht viel nachdenken muss, Copy-Paste mit Korrekturen. Und hier ist die Sache, Es tut mir leid, ich habe seit drei Jahren fast keine Hilfe mehr. Ihr Projekt sind Sie und Sie arbeiten sich den Arsch ab. Nur, dass Apianti es irgendwie mit einem Unterschied gemacht hat, ein besonderer Dank an ihn. Solche rhetorische Frage: wenn ich die Quellen von jemand anderem für mein Projekt adaptiere, Wie hoch ist der Beitrag, den ich leiste? Ich meine, wir schreiben sowieso von Proben...
- Dann kommt die Arbeit des Testens und der Fehlererkennung. Ich habe eine Menge von Assistenten, ich kann mich gar nicht an alle erinnern. 20.000 Beiträge allein auf applelife.ru. Testen kommt in vielen Formen vor, vom einfachen Jammern, dass es nicht funktioniert, bis hin zu spezifische Anweisungen, was im Code geändert werden muss. All dies ist nützlich, sogar wenn jemand jammernd, bringt es Sie dazu, darüber nachzudenken, wie Sie das Problem beseitigen können (ob Sie das Problem lösen oder nicht, das Gejammer muss aufhören!)
- Die schlimmste Art von Fehler, ist ein prinzipielles Problem. Die ersten drei Probleme, die ich über ein halbes Jahr lang allein gelöst: 1. KP in Junior-Systemen, 2. Schlafmangel, 3. Kein Start am Laptop. Die folgenden Probleme wurden gemeinsam mit Dmazar und Pene gelöst: 4. iCloud, 5. iMessage. Und dann gibt es noch das Problem, dass Legacyboot herumhängt. Ich sehe keinen Freiwilligen, der daran arbeiten will, obwohl es diejenigen gibt, die das Ergebnis sehen wollen. 2018: über die Legacy-Boot-Systeme sind zu vergessen, sie werden nicht mehr gebraucht.
- Weitere erwähnenswerte Dinge: Kompilierungsskripte, Installer, System-Skripte und Anwendungen, die direkt mit dem Projekt zusammenhängen, obwohl sie nicht Teil des Laders sind. All diese riesige Arbeit liegt praktisch außerhalb meiner Kompetenz, auch wenn ich damit angefangen habe. Die Hauptakteure hier sind Jadran, Crasybirdy, JrCs, yeah. Und Apianti hat sich darauf eingelassen. Jetzt liegt es in der Kompetenz von vector-sigma.
- Jief Machak hat eine großartige Arbeit geleistet, indem er es in C++ übersetzt und ich habe mit und von ihm gelernt, wie man C++ richtig einsetzt.
- Nun und das Programm muss dokumentiert werden. Wiederum akribisch und nicht ein zu interessantes Werk. An dieser Stelle besonderen Dank an xsmile für die Übersetzung dieses Buches in Englisch, und als solches gelangte es in das WIKI, wo verschiedene Leute von Zeit zu Zeit mitwirken.

Ein separates Wort zu Dmazar. Er ist nicht nur Beiträge gemacht, er ist ein zweiter Autor für Clover. So wie es jetzt steht, ist es nicht meine Autorenschaft. Ohne Dmazar wäre es nur ein Legacy Bootloader.

Dmazar hat einen UEFI-Bootloader erstellt. Er verfügt über die folgenden Technologien:

- OsxAptioFixDrv mit allen Optionen, sonst lädt UEFI macOS nicht. Beachten Sie, dass die Arbeit dieses Treibers ist für jeden Programmierer noch unverständlich, nur der Autor verstand es; dann kam vit9696, und er fand es heraus und machte eine neue Version von AptioMemoryFix, mit NVRAM-Unterstützung. Und jetzt ist es noch einmal umbenannt in OpenRuntime;
- DmpUefiCalls, ohne die wir nicht in der Lage wären, herauszufinden, was das MacOS braucht;
- EmuVariableUefi - für diejenigen, die kein echtes NVRAM haben, und war die Mehrheit. Ohne diesen Treiber auf einigen Konfigurationen wäre das Booten unmöglich.
- sowie das Arbeiten mit NVRAM innerhalb Clovers;
- ccj und Kernel-Patches. Ursprüngliche Quellen von pcj, der sie platziert hat und weggelaufen ist, und Dmazar hat es zu Ende gebracht, ohne sein Copyright darauf zu setzen;

- FSInject wird zum Patchen der Kexts benötigt. Auch hier weiß niemand mehr, wie es funktioniert;
- OsxFatBinaryDxe, wird benötigt für Systeme vor 10.8, da auf diesen Systemen boot.efi FatBinary gibt, was UEFI nicht versteht;
- Tiefschlaf. Wir arbeiteten alle, um es zu ermöglichen, aber es war Dmazar, der darauf hinwies, wie erreicht werden kann;
- und er hat mir tatsächlich geholfen, semantische Fehler in meinem Code zu korrigieren. Das ist richtig, Ohne seine Unterstützung wäre aus Clover nichts geworden

Die anderen Entwickler und Mitwirkenden werden auf die eine oder andere Weise in den Quellen und im Installer erwähnt.

Ich schreibe dieses Kapitel in der Erwartung, dass es andere gibt, die bereit sind, an dem Projekt zu arbeiten, und müssen Sie zumindest lernen, wie man kompiliert.

Requisites

Um das Projekt zu kompilieren, benötigen Sie außerdem einen Compiler und Bibliotheken, das-ist eine Binsenweisheit. Was braucht man in unserem Fall? Die Bibliotheken sind die Quellmodule von EDK2. Wie lautet der korrekte Name dafür? Rahmen? Umwelt? Auf Russisch sollte ich wahrscheinlich sagen MITTEL.

Von demselben Server heruntergeladen. <http://sourceforge.net/projects/edk2/>

Da das gesamte Projekt für Hackintosh und für Hackintosh erstellt wurde, wird an erster Stelle die Kompilierung des Projekts in der MacOSX-Umgebung betrachtet. Dies ist jedoch nicht die einzige Fähigkeit. EDK2 selbst sieht die Kompilierung auch in Windows, Linux und in einige andere Systeme und Umgebungen vor. Für Windows benötigen Sie das Visual Studio 20xx, für Unter MacOSX müssen die Xcode Command Line Tools installiert sein und unter Linux ist der gcc-Compiler ist standardmäßig enthalten. Die eingebauten MacOSX-Werkzeuge reichen jedoch nicht aus, um Kompilierung des Projekts durch zu führen, daher wird, wie bei Linux, vorgeschlagen, eine neue Version von gcc, unter Verwendung des Skripts builgcc.sh im Ordner Clover. Warum nicht Klan? Denn Klan igt immer noch keine brauchbaren Codes aus. Wir warten.

Wartet! Dank Zenith432 er hat endlich eine funktionsfähige Zusammenstellung mit Klan gemacht, in Verbindung mit der dieser Text entsteht Änderungen. Weitere Änderungen aufgrund der Tatsache, dass ich vorschlage, nicht permanent EDK2 ändern, und stabiler Brunch UDK2018. Ja, es ist genug. Allerdings ist die Option mit EDK2 ist ebenfalls vorhanden.

Ab der modernen 2.5k-Version hat sich auch hier eine dramatische Veränderung ergeben, daher sollte der vorherige Absatz als überholt betrachtet werden. Alle notwendigen Umgebungsbibliotheken heutzutage sind in Clover enthalten, und das Repository ist vollständig für die Kompilierung, keine externe Links werden benötigt. Und auch der gcc-Compiler wird wieder als die beste Option erkannt, jetzt Version 10. Die EDK2-Quellen existieren jedoch noch und werden verbessert, wir sollten sie ab und zu anzuschauen, sie mit den eigenen zu vergleichen.

Code schreiben

Wenn Sie Interesse haben, können Sie dem Programmiererteam beitreten und Ihren Teil beitragen. Wir gehen immer einen Schritt weiter, obwohl wir bestimmte Anforderungen an die Codes haben. Ursprünglich wurde das Projekt in der Sprache C erstellt und nach speziellen Regeln formatiert Tiano Code Style.

<https://github.com/tianocore/tianocore.github.io/wiki/Code-Style-C>

Mit dem Übergang zu C++ wollen wir jedoch nicht so viele Makros verwenden, und lieber einen Standard-Programmierstil verwenden. Es sollte verstanden werden, dass Tianos Anforderung ist die Fähigkeit, Quellen auf jeder Plattform zu kompilieren, um mit den schwächsten C-Compilern arbeiten zu können. Und in der Tat ist EDK2 kein funktionierendes Projekt, es ist nur

einen Satz von Quellen, um Ihr Projekt zu erstellen. Aber wir machen ein echtes Projekt, und wir setzen unsere eigenen Regeln, für leistungsfähige Compiler und moderne Hardware.

Über C++.

1. Indem wir Strukturen und Funktionen in logisch vereinheitlichten Klassen zusammenfassen, vereinfachen wir den Code, und dadurch wird sie besser sichtbar. Außerdem stellt sich heraus, dass viele der Techniken, die verwendet werden, wiederholen, so dass es eine unkomplizierte Möglichkeit ist, Codes zu reduzieren. Code wiederverwenden.

2. Es ist nicht notwendig, Variablen im Voraus zu deklarieren, wie es die Sprache C verlangt. Eine Variable muss angekündigt werden, wo es verwendet wird. Erstens ist es visueller, es ist nicht nötig, durch den Text zu scrollen zurück, um die Deklaration zu sehen, zweitens kann der Compiler zusätzliche Optimierung, drittens werden Programmierfehler bei wiederholten Anzeigen reduziert, die nicht durch die Sprache verboten sind. Ein Beispiel ist die Schleife `for (int i=0; i<0; ++i)`. Im alten C war es unmöglich und daher in C verboten, aber für uns ist es besser, den Schleifenindex auf diese Weise zu deklarieren.

3. Allocate/Free-Alpträume können durch die Verwendung von Funktionen beseitigt werden Compiler auf automatische Konstruktor/Destruktor-Speicherverwaltung.

4. Verzicht auf die Verwendung von Arrays zu Gunsten von Vektoren. Arrays steuern nicht außerhalb der Grenzen, während Vektoren einfach sind.

5. Verweigern Sie das Kopieren von Zeigern - die größte Quelle für Fehler im Projekt. Obwohl ermöglicht es Ihnen, etwas zu speichern. Verwenden Sie Links oder kopieren Sie Strukturen als Ganzes. Es scheint teuer zu sein, aber es zahlt sich durch sündhafte Arbeit aus.

6. Vermeiden Sie Übertreibungen. Das Schreiben von `(INTN)N` bedeutet einen logischen Fehler. Wenn die Variable `N` in einem vorzeichenbehafteten Ausdruck verwendet wird, d. h. der kleiner als Null werden kann, warum es als vorzeichenlos deklariert wurde? Und ähnlich verhält es sich mit Zeigertypen. C++ fördert nicht solche Typ-Substitution, und wir fordern, dass Typ-Substitution so selten wie möglich verwendet wird.

7. Verwenden Sie mehr konstante Variablen und Funktionen, d. h. geben Sie dem Compiler mitteilen, dass sich ihr Wert nicht ändern wird, der Compiler dann und der Code entsprechend das andere gilt. Und es wird eine Warnung geben, wenn ein Programmierer gegen das Verbot verstößt. Dies ist als Sicherheit bezeichnet und wird vom Compiler unterstützt.

8. Wir haben keine STL-Bibliothek, aber wir erstellen unsere eigenen Klassen und übersetzen alle Quellen in ihre verwenden.

9. C-Code Style impliziert die Verwendung von `CONST`, `VOID`, `STATIC` geschrieben in Großbuchstaben. Warum? Um die Sprache mehr wie Fortran oder so zu machen?! Wir sind neigen zu mehr Standard-Schreibweise `const`, `void`, `static`, und so weiter.

Compilation

Nun zur Sache. Der Leser, der in dieses Kapitel schaut, kann per Definition nicht ein Benutzer sein, dem man erklären muss, wie man das Terminal verwenden soll.

1. Laden Sie die Quellen herunter.

```
cd ~
mkdir src
git clone --recurse-submodule https://github.com/CloverHackyColor/CloverBootloader.git cd
CloverBootloader
```

2. Kompilierwerkzeuge zusammenstellen

```
./buildgettext.sh
```

```
./build_gcc10.sh
./buildnasm.sh
```

Es ist jedoch möglich, NASM nicht selbst zu bauen, sondern einfach die neueste Version von der offiziellen Website <http://www.nasm.us/pub/nasm/releasebuilds/>, derzeit die Version 2.15.05, und legen Sie es im Ordner `~/src/opt/local/bin/` ab. Wenn Sie etwas davon vergessen, erledigt das Clover-Kompilierungsskript dies für Sie.

3. Das Compilieren von Clover wird entweder teilweise oder vollständig durch skripte erledigt:

```
./buildme
```

Diese werden Sie genau fragen, was Sie wollen. Vector-sigma hat dort ein Menü mit Auswahlmöglichkeiten aus dem Test bereitgestellt

Builds zu einer Vollversion.

Obsolete!

1. Laden Sie die Quellen herunter und bereiten Sie die Umgebung vor:

```
cd ~
mkdir src
cd src
git clone https://github.com/tianocore/edk2 -b UDK2018 --depth 1 UDK2018 cd UDK2018 svn co
svn://svn.code.sf.net/p/cloverefiboot/code/ Clover make -C BaseTools/Source/C cd Clover
```

Oder verwenden Sie den traditionellen EDK2. Sie entwickelt sich auf ihre eigene Weise weiter, während Clover verwendet Funktionen, die schon seit langem entwickelt und abgestimmt wurden, also ist es ausreichend, UDK2018 zu verwenden, aber wenn Sie möchten, können Sie dies auch mit einer bestimmten Revision tun, oder mit letzterem, wenn bekannt ist, dass es mit Clover funktioniert:

```
svn co -r 18198 https://svn.code.sf.net/p/edk2/code/trunk/edk2 edk2 svn co
https://svn.code.sf.net/p/edk2/code/trunk/edk2 edk2
```

2. Kompilierwerkzeuge zusammenstellen

```
./buildgettext.sh
./buildgcc-4.9.sh
./buildnasm.sh
```

Jetzt wird gcc nicht besonders benötigt, Clover kompiliert gut mit eingebautem Klang, und Sie müssen Xcode ohnehin installiert haben, daher ist der Compile-Schalter `-t XCODE8` ist jetzt die Standardtaste. NASM benötigt mindestens die Version 2.13.03, am besten die neueste, sie scheint bereits im System vorhanden zu sein, aber eine alte Version, so dass es entweder mit dem obigen Skript gebaut werden muss, oder einfach Laden Sie die fertige Version hier herunter: <http://www.nasm.us/pub/nasm/releasebuilds/>

Die korrekte Version sollte hier stehen

```
~/src/opt/local/bin/nasm
oder einfach /opt/local/bin/
mkdir -p /opt/local/bin/
sudo cp ~/Downloads/nasm-2.13.03/nasm /opt/local/bin
```

Um mit dem Klang zu kompilieren, benötigen Sie auch das MTOC-Dienstprogramm, das Sie selbst aus dem aktuellen den Quellcode,

```
./buildmtoc.sh
oder nehmen Sie einfach die vorgefertigte Datei, die mit den Clover-Quellen im
Clover/BuildTools/usr/local/bin/mtoc.NEW.zip entpacken, und in /usr/local/bin/mtoc.NEW einfügen
```

Wird nicht mehr geliefert, wenn Sie es noch nicht haben, bauen Sie eine Version mit dem Skript, oder eine vorgefertigte Version aus dem Internet ausleihen. Sie muss nicht aktualisiert werden. Niemand hat jemals beobachtet, dass die eine Version anders funktioniert als die andere.

3. die Anpassung der EDK2-Umgebung an unsere Bedürfnisse durch Auswahl der entsprechenden Patches

```
cd ...  
./edksetup.sh  
cd Clover  
./buildgettext.sh  
cp -R /Patches_für_UDK2018/* ../
```

4. jetzt können Sie CloverEFI selbst bauen. Zum Beispiel so:

```
./ebuild.sh  
./ebuild.sh -mc --no-usb  
./ebuild.sh -ia32  
Standardmäßig ist es -x64 und --xcode8, wenn Sie mit gcc kompilieren wollen, dann  
./ebuild.sh -gcc49  
oder auch -gcc53, oder -t XCODE5
```

Es wurden weitere Kompilierungsskripte erstellt, die in der Regel selbstdokumentiert sind. Siehe, wählen, verwenden.

5. Eine Spitzfindigkeit. Das Repository enthält nicht die HFSPlus.efi-Dateien für 32 und 64 Bit, da sie Privatsphäre. Die beiden Möglichkeiten sind, diese Dateien aus anderen Quellen zu beziehen, oder mit frei VboxHfs. Dieser Treiber ist funktionsfähig, aber etwas langsamer. Es war Unterstützung fremder Dateisysteme

```
#INF Clover/VBoxFsDxe/VBoxHfs.inf  
INF RuleOverride=BINARY Clover/HFSPlus/HFSPlus.inf  
Make (dies ist die Voreinstellung)  
Unterstützung fremder Dateisysteme  
INF Clover/VBoxFsDxe/VBoxHfs.inf  
#INF RuleOverride=BINARY Clover/HFSPlus/HFSPlus.inf
```

6. Ebenso fehlt apfs.efi, aber unser Gegenstück existiert nicht, Sie müssen vom System 10.13+ von Ordner `usr/stanadlone/i386/apfs.efi`, oder `ApfsDriverLoader.efi` verwenden

7. Das Projekt steht nicht still, daher kann dieses Handbuch irgendwann wegen einer kleinen Änderung unbrauchbar. Dieses Projekt ist für diejenigen, die denken, die kann herausfinden, was falsch ist und was zu tun ist.

8. Jetzt den Installer zusammenbauen

```
cd ~/src/edk2/Clover/CloverPackage/  
./makepkg  
./makeiso
```

Alles erledigt! Einige Schritte können weggelassen werden, wenn Sie es für sich selbst und nicht zum ersten Mal machen Zeiten. Anstatt das gesamte Projekt herunterzuladen, aktualisieren Sie zum Beispiel nur `svn up` oder besser `./update.sh` (einschließlich EDK2-Update, aber nicht benötigt für UDK2018), ausschließen 32-Bit-Kompilierung und nicht zum Erstellen eines Pakets und eines Disk-Images.

Für Dummies sind vollautomatische Skripte vorbereitet CloverGrower und seine Pro-Version. Das ist lächerlich. Ein Skript, mit dem ein Dummkopf das Paket selbst kompilieren kann. Dummie ist es besser, das fertige Installationsprogramm herunterzuladen.

NB: Mit den Revisionen 3000+ wurde eine neue Kompiliermethode LinkTimeOptimization eingeführt, die wesentlich Reduzierung der Codegröße. Compiler gcc-4.8 ohne es, 4.9 mit es.

Die Größe ohne ist 675520, mit Optimierung ist 633536, und es funktioniert auf die gleiche Weise, nur schneller.

Das ist für Klang nicht notwendig.

Ab Revision 3059 war es möglich, die Größe der Boot-(CloverEFI)-Datei durch Verschieben von Translation Page Table von Adresse 0x90000 auf Adresse 0x88000, Dateigröße war 483Kb, wurde 450Kb, und dennoch arbeitet es auf einigen Chipsätzen sicherer. In der Tat, es hat Zenith432 hatte das Problem, dass Clover auf seiner Buche nicht funktioniert, er fand eine Überschneidung Adressen. Und da mit LTO die Codegröße erheblich gesunken ist, ist es möglich geworden, TRT zu bewegen. Jetzt lässt er auch Clover arbeiten!

Compilation mit dem Befehl:

```
./ebuild.sh -low-ebda
```

Dieser Schlüssel `undo` ist jetzt der Standard, der Schlüssel `-std-ebda` ist erforderlich, um ihn zu überschreiben. Für Entwickler ist es auch interessant zu wissen, wo der Buchstabe "B" oder "6" gebildet wird, der wird angezeigt, wenn CloverEFI startet. Verwenden Sie einen Hexadezimal-Editor, um den Offset zu suchen 0x02a9, dort sehen wir Byte 0x42 bzw. 0x36. Übrigens, um diesen Brief zu stornieren kann nicht, da es ein Trick ist, Clover überhaupt zum Starten zu bringen. Das BIOS erzeugt Initialisierung des Videoadapters an dieser Stelle. Sie können es auf Leerzeichen 0x20 ändern.

Making a DEBUG version of Clover

Wenn sich Leute darüber beschweren, dass Clover nicht startet und daher nicht funktioniert, kann ich nicht helfen, aber ein paar Standardtipps zu geben. Ich bin ein Mann, der kann einfach den Code bearbeiten und ein wenig kompilieren, kann zu Punkt X kommen, und dann werden wir zusammen festzustellen, was los ist.

Option 1: Wenn der Legacy-Boot bei Meldung 6_ hängen bleibt. Dies bedeutet, dass Sie im Inneren von Duet selbst hängen. Es ist nicht möglich, ein Protokoll zu erstellen. Es ist nur möglich, die Anzeige von Meldungen auf dem Bildschirm zu beobachten. Der Ablauf des Duetbetriebs ist wie folgt:

1. Startsektor `st32_64.s`. Genau an dieser Stelle wird 6 ausgegeben, lesen BIOS-ROM, A20-Umschaltung und CPU-Umschaltung von 16 Bit auf 32 und 64.

Meine Version des Sektors unterscheidet sich von der Vanilla-Version darin, dass sie auf einem Laptop läuft. Das ist nicht ausgeschlossen, dass es auch hier für jemanden ein Problem geben wird. Es ist notwendig, den Ausgang anderer Nummern an anderen Stellen einzufügen, und beobachten.

2. Beginn der C-Codes. Datei `efi32.s` oder `efi64.s`. Es ist unwahrscheinlich, dass sich hier etwas verlangsamen kann.

3. Datei `Efiloader.c`. Hier können Sie die Ausgabe auf den Bildschirm einfügen mit der Prozedur `PrintHeader('A')`; die derzeit auskommentiert ist.

4. Datei `Dxelp/Dxelinit.c`. Sie können auch `PrintHeader` einfügen, aber trotzdem `Debug.c` in um dieses Modul zu kompilieren.

5. `DxeCore`. Die Ausführung ist hier schon ausufernd und es ist schwieriger nach zu vollziehen, was und wo passiert.

Fazit: **Zum Bildschirm kann man in irgend einer Weise gelangen.(?)**

6. `CLOVERX64.EFI` selbst wird in der Prozedur `BdsBoot.c/BdsLibBootViaBootOption()` geladen.

An dieser Stelle im Programm können Sie bereits die Standard mit der Prozedur `AsciiPrint("\nasty!\n");`

Option 2. Duet selbst funktioniert, was Sie überprüfen können, indem Sie die Leertaste unmittelbar nachdem die 6 auf Bildschirm angezeigt wird. Entweder haben wir einen UEFI-Boot, und es gibt kein Duet, dann sollte die CloverGUI laufen, oder es gibt keine, oder es gibt eine, aber sie bleibt hängen.

Die Standard-Methode `Boot->Debug=true` funktioniert nicht, da Sie den Speicherort für mehr Details brauchen.

Entfernen Sie in diesem Fall in der Datei `Platform.h` den Kommentar in Zeile 11

```
##define DEBUG_ALL 2
```

oder in den Dateien, die Sie interessieren, in den oberen Zeilen `DEBUG_xxx 2` ein. In diesem Fall wird der gesamte Ausgang von `DBG("Nightmare #3\n");` wird auf den Bildschirm übertragen. Und auf diese Weise wird es möglich sein interaktiv zu beobachten, wohin die Programmausführung gelangt, bevor sie hängen wird.

ZS. Verwenden Sie keine russischen Buchstaben, wie ich hier illustriert habe! Es funktioniert nicht. C durch Umschalten auf C++ können russische Buchstaben im Debug verwendet werden

Option 3. Kompilieren der Debug-Version mit Haltepunkten (stop points), Ausführen unter Kontrolle von QEMU mit spezieller Version von gdb installiert. Dmazar hat diesen Weg einmal ausprobiert. Doch meiner Meinung nach sind diese Bemühungen nicht den Aufwand wert. Ein einfacher Trace ist immer ausreichend.

Option 4. Geben Sie beim Kompilieren `./ebuild.sh -D DEBUG_ON_SERIAL_PORT` und schließen Sie an die serielle Schnittstelle einen anderen Computer an, der in der Lage ist, Briefe in serielle Schnittstelle zu empfangen (Terminal in Windows). Die Variante funktioniert unter QEMU, wenn Sie beim Starten angeben

"-serial stdio"-Flag

Installation

Verwendung des Installationsprogramms

Wofür ist das Installationsprogramm?

So installieren Sie das Programm! Warum sollten Sie es manuell tun? Der Installer wird alles genauer machen als Sie! Die einzige Voraussetzung ist, dass Sie auf diesem Computer bereits MacOSX haben. Eine Möglichkeit ist, dass Sie die Installations DVD mit einem anderen Bootloader starten und von dem Installationsprogramm über die MacOSX-Installationsoberfläche die Installation von Clover starten (?)

Abhängig von der Sprache des Betriebssystems läuft das Installationsprogramm in Russisch, in Englisch oder sogar in Chinesisch. Hier sind die Anweisungen für die englische Version, auf Russisch können sie selber herausfinden wo was ist. und ich kann auch kein Chinesisch. Die aktuelle Version umfasst 20 Sprachen, darunter Indonesisch, falls jemand es braucht.



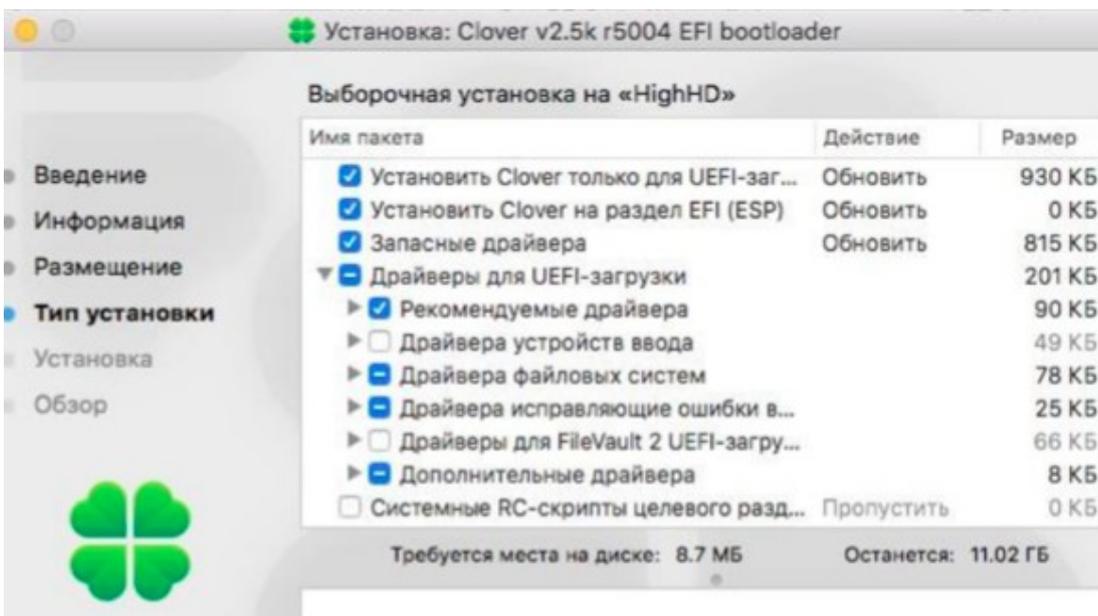
Ok.,

Folgen Sie dem Key Weiter (Continue) und OK und lesen und akzeptieren Sie die Lizenzvereinbarungen (hmm, gibt es die?), und kommen sie zu der Entscheidung, was wir wo und warum installieren.



Neueren Screenshot einfügen

Hier haben sie die Möglichkeit, den Installationsort zu ändern - wählen Sie aus, auf welchem Laufwerk der Bootloader installiert werden soll. Wenn Sie beabsichtigen auf einer EFI-Partition zu installieren und wählen Sie dann nur die Partition mit dem aktuellen System aus. MacHDD in diesem Beispiel. Kontrollkästchen auf "Clover im ESP installieren". Das Installationsprogramm findet die EFI-Partition auf demselben Laufwerk.



Englische Version und neueren Screenshot einfügen

Аangepasste oder Standardinstallation - Wählen Sie die Boot-Optionen. Wenn Sie den Cursor auf einer der Zeilen platzieren, erscheint im unteren Feld eine kurze Beschreibung dieser Option.

Clover nur für UEFI-Boot installieren - diese Option beendet die Installation der Boot-Dateien. Die englische Version bedeutet "Install for UEFI motherboard", und natürlich werden die Benutzer, das Kästchen ankreuzen. Das Ergebnis ist, wie immer, dass Clover nicht geladen wird. Ich bedauere sehr, dass die Entwickler des Installers diesen Punkt für ihre eigenen Zwecke erfunden haben.

Leute, um Himmels willen, aktivieren Sie dieses Kästchen nicht, auch wenn Sie einen Computer mit UEFI-BIOS haben. Der gewünschte UEFI-Boot wird trotzdem funktionieren!

Die Installation von Clover im ESP ist die beste Option, wenn es eine solche Partition gibt (GPT-Partitionsschema). Das Installationsprogramm kann diese Partition nicht sehen, Zeigen Sie also im Festplattenauswahlmenü auf die Partition, die sich auf derselben Festplatte befindet, auf ESP auf den wir den Bootloader legen wollen. Wir nehmen an, dass sich auf dieser Partition MacOSX befindet, wo die Skripte, das Control Panel und der Updater installiert werden (zu lang auf Russisch) "Automatisches Update-Programm").

Die ESP-Partition sollte vor der Installation ausgehängt werden (you should unmount the ESP partition). Und für Catalina müssen sie den Befehl, bevor Sie das Installationsprogramm ausführen, im Terminal eingeben:

```
sudo mount -uw /
```

die es Skripten ermöglicht, Dateien in Systemordner zu schreiben. Wenn das System schreibt, dass der Installer nicht geöffnet werden kann, weil sein Autor ein nicht autorisierter Entwickler (unidentified developer), können Sie in den Systemeinstellungen unter, Sicherheit gehen, und dort drücken Sie die Schaltfläche "Trotzdem öffnen". Oder Sie können den Befehl im Terminal ausführen

```
sudo spctl --master-disable
```

Es ist jedoch nicht meine Aufgabe, Ihnen die Tricks der macOS-Nutzung beizubringen.

Bootloader ist eine Variante mit dem BIOS (Variante A), die CloverEFI, oder mit UEFI (Option B).

- **aktualisieren sie MBR- und PBR-Sektoren nicht.** Aktualisieren sie diese Sektoren nicht, da sie bereits gibt, oder nur für die Option B;
- **boot0af im MBR installieren** - Booten mit boot0af, d. h. Suchen der aktiven section (Teilung?). Das Installationsprogramm macht die ausgewählte Partition aktiv. Die Ausnahme ist die Installation von auf eine EFI-Partition, wird diese nicht aktiv gemacht, und boot0af findet keine aktive Partition die Boot-Datei von der EFI-Partition laden, was wir für das Legacy benötigen Booten Sie von der GPT-Platte, von der ESP-Partition.
- **Installieren Sie boot0ss im MBR** - booten Sie mit boot0ss, d.h. suchen Sie nach einer HFS+-Partition, auch wenn sie inaktiv ist. Das Installationsprogramm ändert die aktuell aktive Partition nicht. Dies ist ist für eine Konfiguration mit einer aktiven Windows-Partition gemacht - Windows braucht es.

Alternatives Booten PBR verwenden - wie in dem Kapitel "Was ist was", kann der PBR-Sektor für 2 Sekunden angehalten werden, um die Tasten 1-9 zu drücken, oder ohne sie. Mit dieser Option wird der Sektor mit Pause eingestellt. Diese Option ist nicht mehr verfügbar in der Installateur ;)

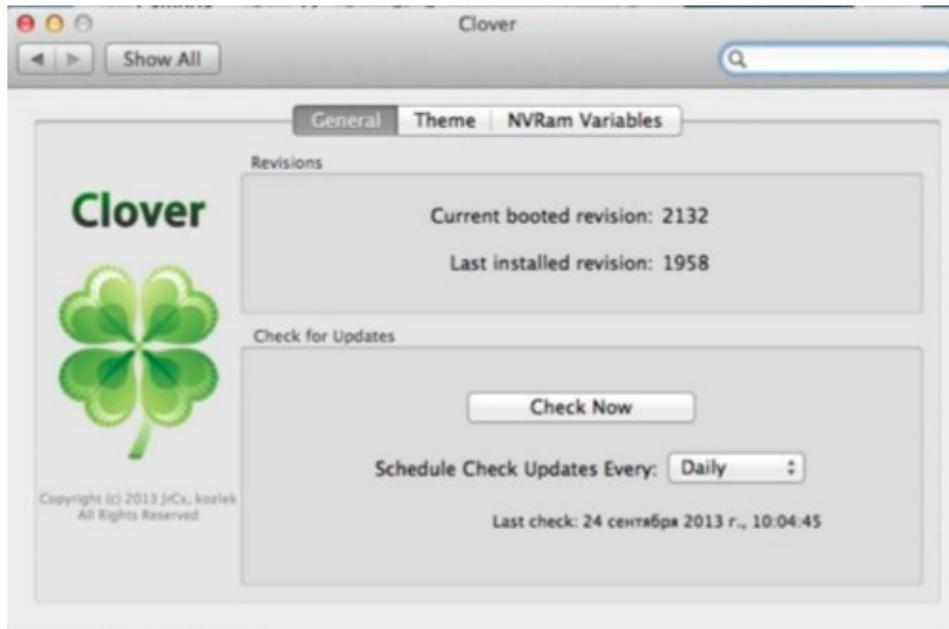
CloverEFI ist, wie Sie aus der Liste ersehen können, die Bitratenwahl des Bootloaders. Entweder 32 Bit oder 64 Bit. Auch hier gibt es eine spezielle Variante von BiosBlockIO. Dies ist eine Variante von CloverEFI-64, die einen speziellen Namen hat, nämlich boot7, und für Computer gedacht ist, die einen nicht standardisierten SATA-Controller haben. Dieser Treiber läuft über das BIOS, und normalerweise, funktioniert das mit jedem Controller (das BIOS muss ja mit Controllern arbeiten!). Aber es gibt auch Fehler, wie zum Beispiel das Dell Inspiron 1525. Eine weitere spezielle Variante ist boot7-MCP79. Es ist so ähnlich wie die einzige praktikable Option für den MCP79-Chipsatz, obwohl es dort nichts Spezifisches gibt für diesen Chipsatz, kann aber auch in anderen Fällen funktionieren. Diese Option wurde von Oscar09 gefunden, der auch einen USB-Patch für diesen Chipsatz vorgeschlagen hat.

Drivers. Die Auswahl der Treiber wird im obigen Kapitel "Was ist was" erläutert.

Installieren Sie die RC-Systemskripte auf der Hauptpartition - dies sind die rc.local und rc.shutdown.local, die vom OSX-System bei der Anmeldung und Abmeldung ausgeführt werden - eine notwendige Teil des gesamten Legacy-Clover-Konzepts. Für diejenigen, die ein funktionierendes NVRAM für UEFI-Boot haben, sind diese Skripte nicht notwendig.

Installieren Sie die Skripte auf allen anderen Partitionen - wenn es mehr als eine MacOSX-Partition auf dem Computer gibt. Das Installationsprogramm ist schlau genug, sie nicht auf Windows-Partitionen oder Linux Partitionen abzulegen.

Installieren Sie das Clover-Bedienfeld - dieses Bedienfeld hilft bei der Aktualisierung von Clover, der Themenauswahl und das Setzen von NVRAM-Variablen:



Manuelles Installieren des Bootloaders

Wird in zwei Fällen benötigt: bei Flohbefall und bei Durchfall. **Erstens**, wenn eine Person gut bescheid weiß, was er tut und er jeden Schritt kontrollieren will, dem Installer nicht vertraut (und das ist vergebens!), und, **zweitens**, wenn die Installation von einem anderen Betriebssystem aus erfolgt, auf dem das Installationsprogramm nicht ausgeführt werden kann.

OSX

Es wird dringend davon abgeraten, dies bei jemandem zu tun, der nicht weiß, was ein Terminal ist.

Installation auf einer HFS+-Partition in MBR oder hybrider Partitionierung. Warum MBR? Es ist eine oft zu findende Standardsituation, wenn ein Computer bereits vorhanden ist, und bereits mit Informationen darauf sind, die sie nicht zu verlieren wollen, dann können Sie nur einen neuen Bootloader installieren.

MBR-Sektor einrichten:

```
cd BootSectors
sudo fdisk440 -f boot0 -u -y / dev / rdisk0
```

Was bedeutet das?

fdisk440 ist eine spezielle Version des Dienstprogramms fdisk, die so korrigiert wurde, dass sie nur 440 Bytes des Null-Sektors korrigiert, damit es für die Windows-Kompatibilität funktioniert (Aufwachproblem), um das sich Apple nicht gekümmert hat.

boot0 - die oben im Kapitel "Was ist was" beschriebene Datei

rdisk0 ist das physikalische Gerät, auf dem Sie den Bootloader ablegen werden. Vergewissern Sie sich, dass es tatsächlich die Zahl 0 hat.

Diese Dateien werden mit Clover mitgeliefert. **In der neuen Version ist fdisk440 nicht mehr dabei, in erster Linie, weil sie in der neuen Umgebung nicht kompiliert werden kann.** Verwenden Sie nun dd stattdessen, wie für Linux gezeigt. Und Sie können die Partition mit dem Standard fdisk-Programm aktivieren. Bei der GPT-Partitionierung wird sie jedoch nicht benötigt.

Installation des PBR-Bereichs:

```
sudo dd if=boot1h2 of=/dev/rdisk0s9
```

boot1h2 - PBR-Sektordatei für HFS+-Dateisystem, anders als bei ähnlichen Unterstützung für große Boot-Dateien und die Möglichkeit, boot1,3,6 per Hotkey auszuwählen. Einzelheiten finden Sie im Kapitel "Was ist was".

rdisk0s9 ist die neunte Partition auf dem ausgewählten Gerät... Warum die neunte? So dass Narren nichts durch stupides Wiederholen von geschriebenen Befehlen vermasset können. (Es gibt es wahrscheinlich keinen solchen Abschnitt.) Aber Sie sollten die reale Zahl angeben, z. B. die erste Partition. Nun, nachdem die MBR- und PBR-Sektoren erfolgreich in den ausgewählten Gerät/ausgewählte Partition sind, sollten Sie diese Partition aktiv machen

```
fdisk440
-e /dev/rdisk0
>f 9
>w
>q
```

Die Neun in der zweiten Zeile ist wiederum die section number (Abschnittsnummer) (es gibt vier davon!) - leiten Sie sie ab. Sie können nun die Boot-Datei und den EFI-Ordner in das Stammverzeichnis der Partition kopieren.

Installieren Sie auf einer FAT32-Partition.

Im Gegensatz zur vorherigen Methode gibt es hier eine Feinabstimmung. Der PBR-Bereich muss enthalten Trenngeometrie. Diese Informationen werden dort während der Partitionierung eingetragen, so dass der Verlust solcher Informationen ist mit Konsequenzen behaftet ist. Die Methode der Installation des Sektors selbst ist Kompliziert

```
dd if=/dev/rdisk1s9 count=1 bs=512
of=origbs cp boot1f32alt newbs
dd if=origbs of=newbs skip=3 seek=3 bs=1 count=87
conv=notrunc dd if=newbs of=/dev/rdisk1s9 count=1 bs
=512
```

boot1f32alt - bereits im Kapitel "Was ist was" erwähnt - ist ein Sektor zur Installation auf einer FAT32-Partition. Aber das ist nicht FAT16! Seien Sie vorsichtig!

rdisk1s9 - wiederum die neunte Partition auf dem ersten Gerät. Geben Sie Ihre konkrete Zahlen ein. Der Rest sind die Buchstaben und Zahlen, die in dieser Vorschrift sind nicht verändert werden dürfen. Der Rest der Schritte ist ähnlich wie bei der Installation auf HFS+.

Für Besitzer von Festplatten mit 4k-Sektorgröße gibt es eine Warnung!

Im ersten und vierten Befehl müssen Sie anstelle von bs=512 bs=4096 schreiben.

ZS. Allerdings... das ist keine Tatsache! Bei mir hat eine 512er-Installation mit einem 4k-Laufwerk problemlos funktioniert. (Zum Hintergrund: <https://www.heise.de/ct/hotline/FAQ-Festplatten-mit-4K-Sektoren-1378642.html> (gilt das noch für die heute üblichen SSDs?))

Installieren auf einer exFAT-Partition

Ab Revision 3040 ist es nun möglich, Clover auf einer exFAT Format-Partition zu installieren. Ich habe zum Beispiel eine externe Festplatte mit diesem Format. Da ich sie nicht mehr neu partitionieren kann, und auf dem alten Weg gibt es keinen Platz für die Installation von Clover gibt. Die Installation eines PBR-Sektors erfordert ein spezielles boot1-Installationsprogramm, geschrieben von Zenith432. Der Befehl ist einfach:

```
./boot1-install -u -y /dev/disk2s1
```

Grundsätzlich hat Zenith mit diesem Dienstprogramm sowohl HFS+ als auch FAT32 installierbar gemacht. Wichtig ist, dass die entsprechenden Sektoren (boot1h, boot1f32, boot1x) im gleichen Ordner liegen. Bei exFAT heißt der Sektor boot1x. Die Option boot1xalt bootet mit einer Pause.

Es gibt jedoch noch eine weitere Überlegung bei der Installation dieser Sektoren. **Das Volume, auf dem Sie den Sektor installieren möchten, müssen Sie ihn zuerst aushängen!** Und dies ist nur möglich, wenn die Platte nicht die System-Platte ist. Andernfalls booten Sie von einem anderen Medium, um einen solchen Vorgang durchzuführen.

[Linux](#)

Unter Linux gibt es auch ein Terminal, und fast die gleichen Befehle, aber die Installation ist nur auf FAT32 möglich. Die Unterschiede sind wie folgt:

- anstelle von rdisk1 wird sdb verwendet - siehe Ihre Linux-Version für Details.
- Verwenden Sie das gleiche dd anstelle von fdisk440, um den MBR zu schreiben

```
dd if=dev/sdb count=1 bs=512 of=origMBR
cp origMBR newMBR
dd if=boot0 of=newMBR bs=1 count=440 conv=notrunc
dd if=newMBR of=/dev/sdb count=1 bs=512
```

Windows

Unter Windows ist es auch sinnvoll, den Bootloader nur auf das FAT32-Flash-Laufwerk zu legen, denn das führen Sie einfach mit dem Skript aus:

makeusb.bat E:

wobei E: der Buchstabe Ihres Flash-Laufwerks ist. Es sollte nicht mehrere Partitionen auf ihm haben. Das ist ja Windows!

Alle für das Skript benötigten Dateien sind im Lieferumfang von Clover enthalten. Allerdings, das Installationsprogramm muss zuvor entpackt werden, oder Sie können diese Dateien direkt von der svn herunter laden. Sie befinden sich im Ordner edk2/Clover/CloverPackage/CloverV2/BootSectors Ordner

Nach dem Ausführen des Skripts muss das Flash-Laufwerk ausgeworfen und wieder eingesteckt werden, dann kopieren sie noch die Boot-Datei und den EFI-Ordner darauf. Noch besser ist es, BootDiskUtility.exe von Cvad, für die Erzeugung eines Windows-Flash-Laufwerk zu verwenden.

<http://www.applelife.ru/threads/delaem-zagruzochnuju-clover-fleshku-s-macosx-iz-windows.37189/>

Empfohlene Installationsoptionen

Dies sind die Optionen, die ich persönlich für die Installation empfehlen würde:

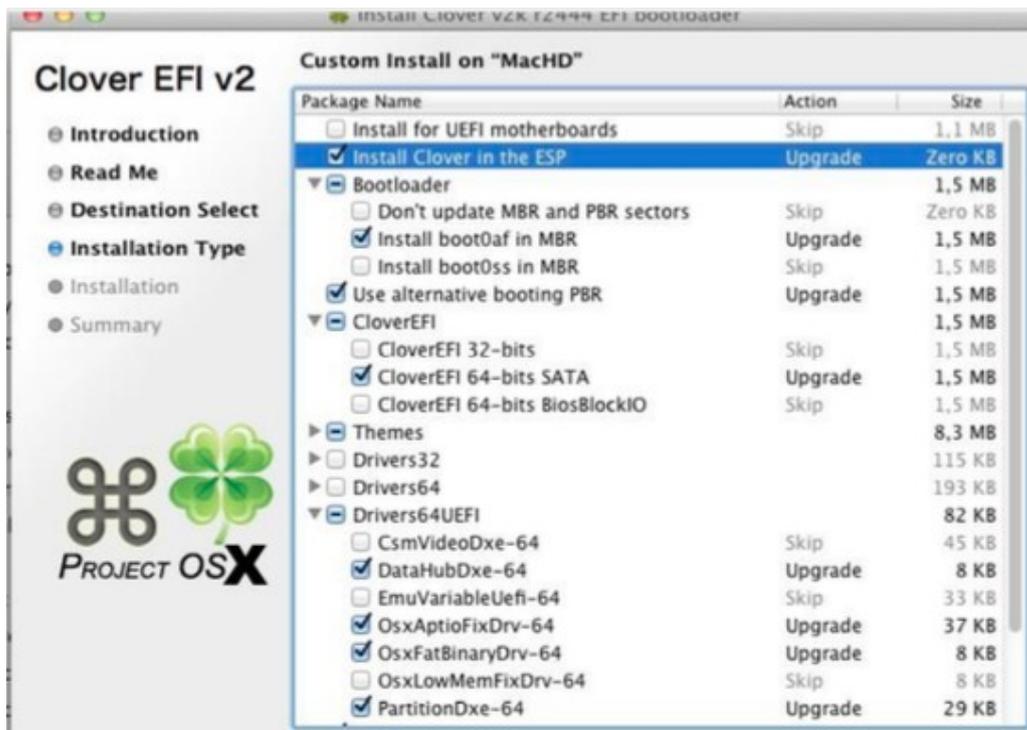
1. die primäre Festplatte ist in GPT (GUID Partition Table) partitioniert. Dies ist ein modernes Schema, das für MacOS korrekt ist, und es wird von Windows seit Version 7 unterstützt. In diesem Schema gibt es eine unsichtbare ESP-Partition (EFI System Partition) von 200 MB. Standardmäßig ist es bereits in FAT32 formatiert und muss nicht neu formatiert werden, ist aber auch nicht schädlich. Das ist der Ort, wo Clover sitzen sollte, sowohl für Legacy- als auch für UEFI-booten.

Für den UEFI-Boot müssen Sie nur den Ordner kopieren (oder mit dem Installationsprogramm installieren),

die EFI mit allen notwendigen Inhalten. Aber damit das BIOS diese Boot-Option sieht müssen Sie die Datei /EFI/CLOVER/CLOVERX64.EFI nach /EFI/BOOT/BOOTX64.EFI kopieren, wenn sie nicht vorhanden ist. Für diejenigen, die es nicht verstehen, machen Sie es in zwei Schritten: Kopieren in den Ordner /EFI/BOOT/, und benennen Sie ihn dann in BOOTX64.EFI um. Um Legacy-Boot durchzuführen, müssen Sie die Datei boot0af in den MBR-Sektor dieser Festplatte und in den PBR dieser EFI-Partition die Datei boot1f32 (oder boot1f32alt, die mit einer mit Pause startet) schreiben.

Machen Sie niemals eine Partition aktiv!

Dell-Laptop hat American Megatrend EFI BIOS, und es erlaubt nicht, das Legacy Booten auf eine GPT-Platte. Leider! Für andere ist das möglich, und im Installationsprogramm tun Sie dies: ([der Screenshot ist nicht mehr aktuell](#), aber die Bedeutung bleibt)



2. Die Festplatte ist bereits als FDISK-Partitionierungsschema partitioniert, das allgemein als MBR bezeichnet wird. Altes Schema, das schon immer für Windows XP verwendet wurde. Auf diesem Laufwerk haben wir eine Partition zugewiesen, auf der wir OSX installieren, um es auszuprobieren. UEFI-Bootung kommt hier nicht in Frage, aber Legacy, Wir könnten hier auch Clover verwenden. Dazu schreiben Sie die Datei boot0ss in den MBR-Sektor. In diesem Sektor wird nach einer HFS+-Partition gesucht, die wir für den Mac gemacht haben, und die ihm die Kontrolle gibt. Die aktive Partition kann für Windows bleiben, es braucht es mehr. Sie müssen die Datei boot1h in den PBR-Sektor dieser Partition schreiben aus dem Clover-Bausatz. **Ich erinnere Sie noch einmal daran, dass Sie, wenn Sie zuvor mit Chameleon gearbeitet haben, ist dort die gleiche Datei aus dem Chameleon-Bausatz vorgeschrieben, und es wird nicht mit Clover arbeiten. Wir müssen es ersetzen!** Die folgenden Kontrollkästchen sind im Installationsprogramm vorhanden



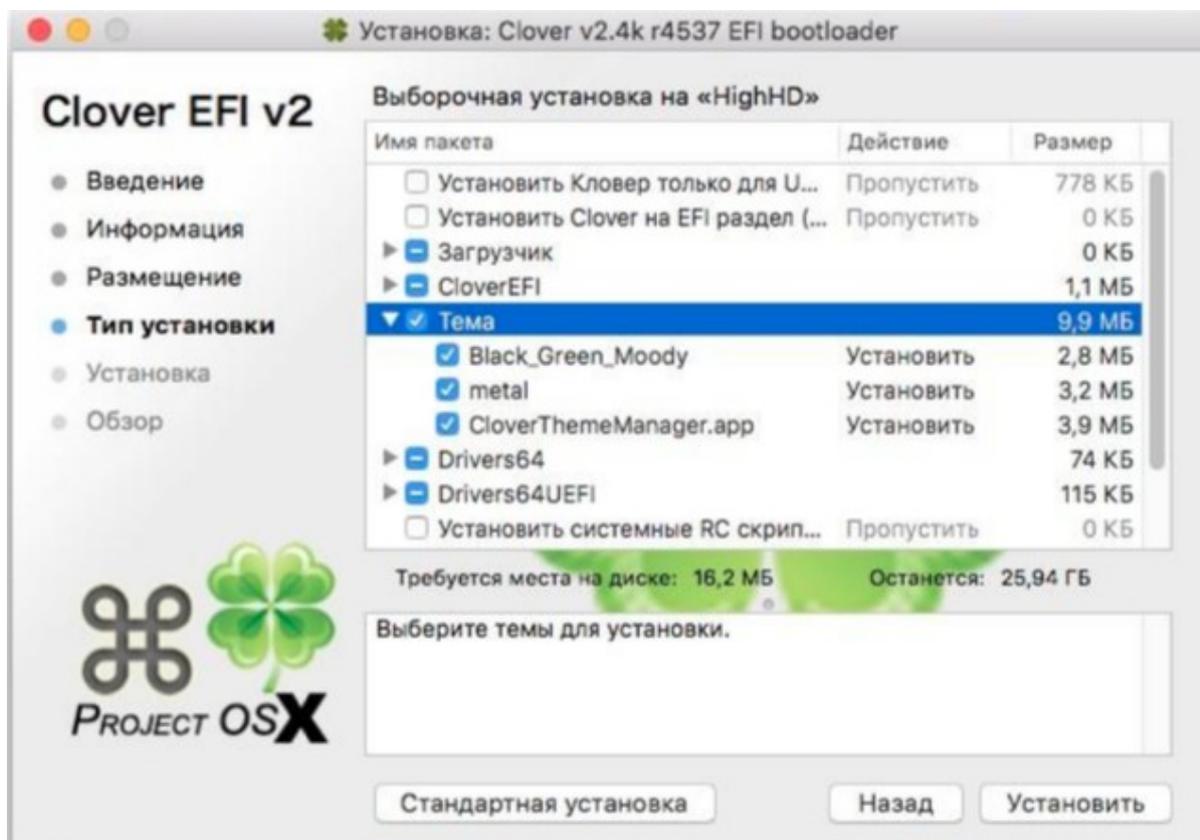
3. Sie haben sich entschieden, dem Mac eine separate Festplatte zu widmen. Hier gibt es eine einfache Möglichkeit per BIOS zu wählen, was geladen werden soll, und die Festplatte wird entsprechend dem ersten Szenario vorbereitet. Entweder Sie legen es auf die primäre die Datei boot0md in den MBR-Sektor. Seine Eigenschaft ist, dass er nach HFS+-Partition über alle Festplatten hinweg sucht, und es würde in diesem Fall als zweite Option funktionieren. Lesen Sie nun erneut das Kapitel zur manuellen Installation. Und zu guter Letzt. Ist es nicht an der Zeit, Windows XP ganz abzuschaffen, und Windows UEFI, z. B. Sieben zu nutzen? Die MBR-Platte kann ohne Datenverlust in HTP mit dem Dienstprogramm gdisk umgewandelt werden. MacOS würde davon profitieren, und Windows sowieso, das alle sechs Monate neu installieren werden muß. Falls es jemand falsch versteht: Windows UEFI wird von Clover perfekt installiert auf Maschinen, die kein UEFI-BIOS haben.

Registration

Design

Wahl des Themas

Lassen Sie uns nun ein Thema auswählen. Was ist ein Thema? Dies sind die Gestaltungselemente: das Banner, der Hintergrund Bild, Icon- und Button-Zeichnungen, Schriftart, vereint durch eine einzige künstlerische Idee.



(Screenshot ersetzen ru Version!)

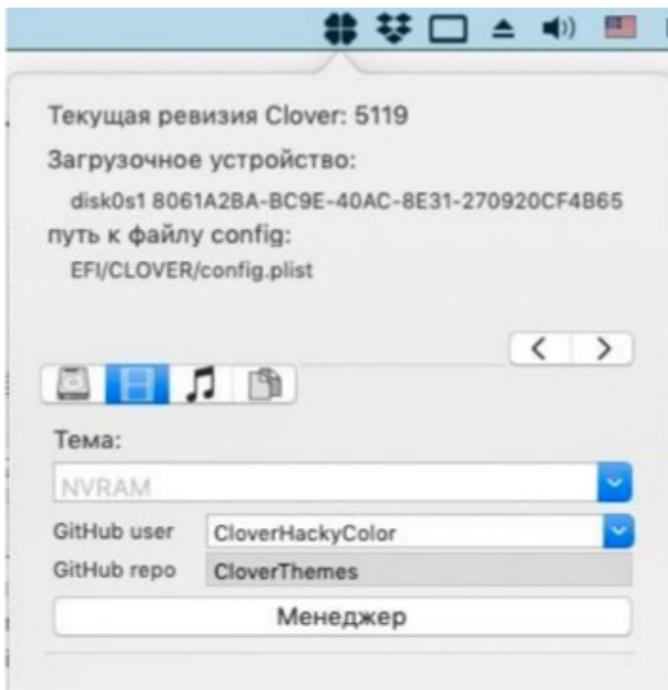
Es gibt drei Themen im Installationsprogramm, mein Cäsium-Thema, ein Thema von Clovy, das sind Vektorthemen, und ein Rasterthema von Blackosx, aus irgendeinem Grund hat er sein SVG-Thema nicht für den Installer angeboten. Außer ihnen gibt es "Ostereier", wer sie gesehen hat, weiß Bescheid, für den Rest wird es eine Überraschung sein. Und dann ist da noch Clover. Clovy's eingebautes Thema, genannt embedded. Wenn Sie kein anderes Thema installieren, wird dieses immer noch zur Verfügung stehen. Und es gibt eine App, mit der Sie jedes Thema aus den 40 verfügbar über die Gita. Ich werde Ihnen die Details nicht verraten, alles ist dort transparent.

<http://www.insanelymac.com/forum/topic/302674-clover-theme-manager/>

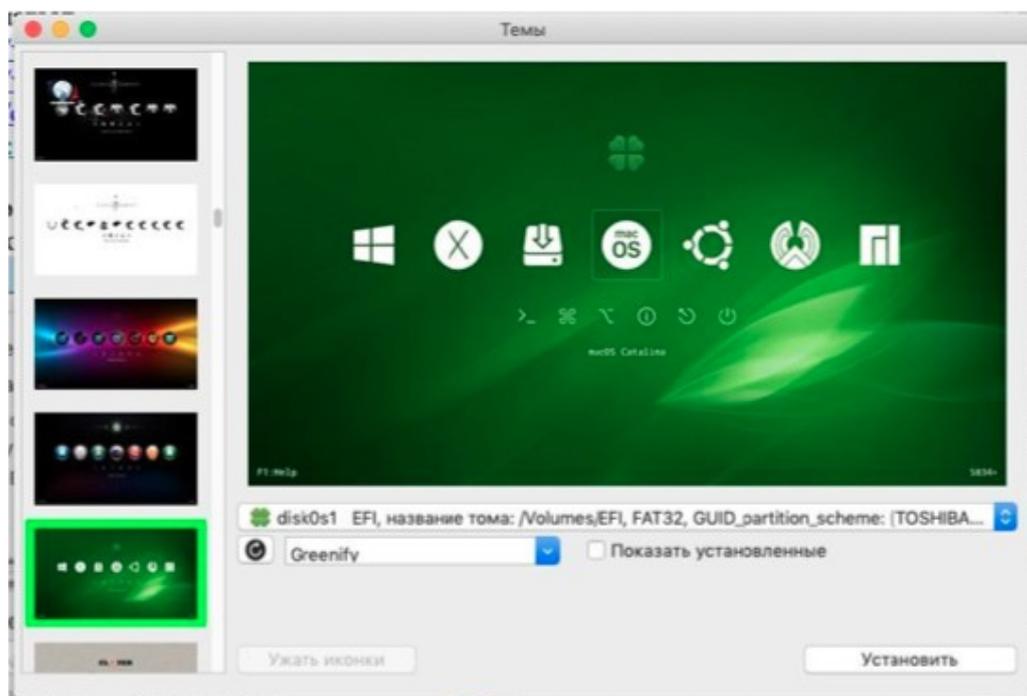
In der modernen Version können Sie ein "zufälliges" Thema angeben und jedes Mal, wenn der nächste Start machen, können sie die unterschiedlichen Designs beobachten.
Lassen Sie uns einen kurzen Überblick über die Themen geben. Der eigentliche Themenkomplex ist viel breiter, schauen Sie sich die Foren an, wer macht, was wer schlägt vor:

<http://www.applelife.ru/threads/themes-temy-dlja-zagruzchika-clover.36074/>
<http://www.insanelymac.com/forum/topic/288685-clover-themes/>
<http://clover-wiki.zetam.org/Theme-database>
<https://sourceforge.net/p/cloverefiboot/themes/>

Jetzt hat vector-sigma seinen Clover Themen-Manager innerhalb der Clover.app
Die Taste sieht aus wie Film.

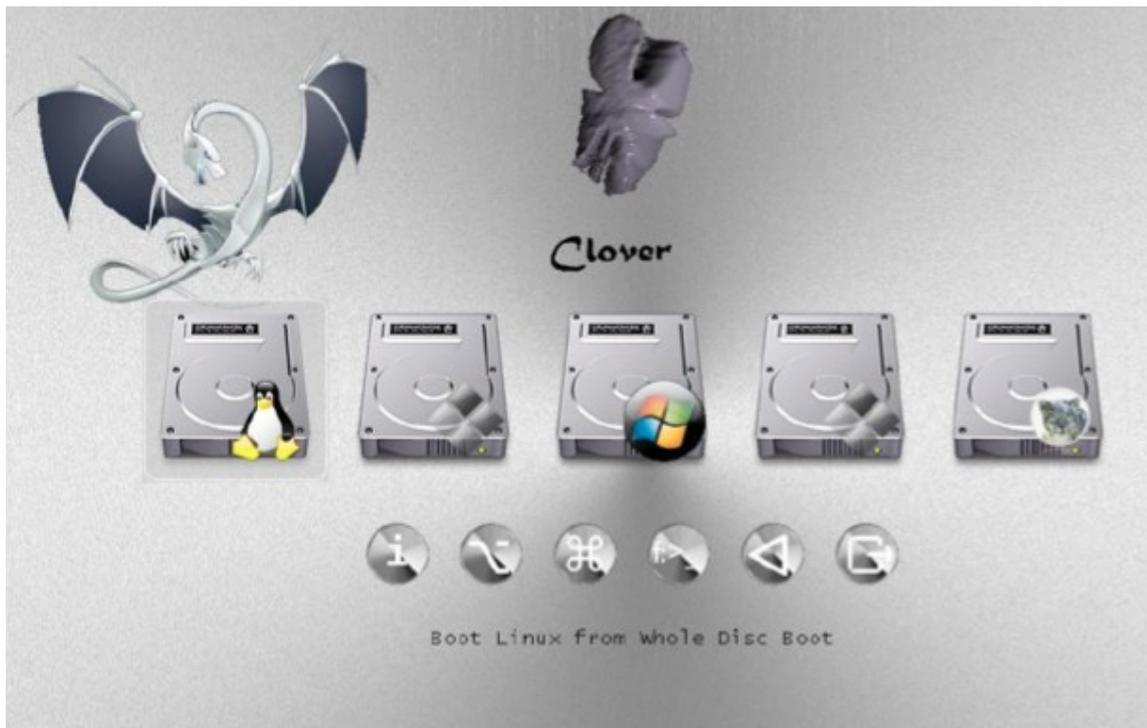


Auswahl eines Themas



Clover-Bootloader-Themen

Metall. Autor Slice. Thema 1.



Black-green (Schwarz-Grün). Autor blackosx. Das Theme, das standardmäßig mit diesem Installationsprogramm geliefert wird.



Steampunk. Autor medik.



Und Dutzende andere... Das Erstellen eines eigenen Themas ist eine besondere Geschichte mit eigenen Regeln.

Konfigurieren der interface (Schnittstellen) in der config.plist

Die Themen umfassen auch eine Reihe von Parametern, die in die Datei config.plist geschrieben werden. Für ältere Versionen, siehe die alten Versionen der Anleitungen.

Die Schnittstelleneinstellungen werden in EFI/CLOVER/config.plist unter GUI vorgenommen

```
<key>GUI</key>
```

```
<dict>
```

Die Oberfläche kann grafisch oder textuell sein (beginnend mit der Revision 1764). Zu diesem Zweck dient der folgende Key

```
<key>TextOnly</key>
```

```
<true/>
```

Wahrscheinlich gibt es nur in Russland Liebhaber textueller Oberflächen, Total Commander, Volkov Commander, DOS, usw., usw. Zu Ihren Diensten!

Clover - Main Menu

```
Boot Legacy from Test_Clover
Boot Windows from TEST_UISTA
Boot UEFI internal from EFI
Boot UEFI internal from Test_Clover
Boot Mac OS X from Test_Mavericks
Boot Mac OS X from Test_Lion
Boot Mac OS X from Test_Mountain_Lion
About rEFIt
Options
Clover Boot Options
Start UEFI Shell 64
Restart Computer
Exit Clover
```

Wenn der Key false gesetzt ist, läuft Clover im grafischen Modus.

Die Auflösung des Textbildschirms, hier wie auch in der Shell und dem boot.efi-Bildschirm, kann eingestellt werden mit Hilfe von

`<key>KonsoleModus</key>`

`<string>0/Min / Max / some number</string>`

Die Zahl finden Sie in Ihrem boot.log, oder geben Sie Max ein - es wird das Maximum der möglichen Auflösung sein .

Das Aussehen der grafischen Shell hängt vom gewählten Thema ab. Das Standardthema wird in der Variablen festgelegt:

`<key>Thema</key>`.

`<string>Metal</string>`

Das Thema kann jedoch auch in der Systemsteuerung ausgewählt werden, und diese Auswahl wird bestimmen. Wenn dort jedoch das falsche Thema angegeben ist (es gibt keine solche theme.plist-Datei von dem Thema) dann wird auf dem Bildschirm ein eingebettetes Thema (embedded) angezeigt, die neueste Version von Clovy-Thema kann auch im Bootloader-Menü geändert werden, dort wird eine Liste der installierten Themen angezeigt (rev. 1955), und Sie können angeben, welche Sie wünschen. Die Oberfläche wird neu gezeichnet, nachdem Sie zum Hauptmenü. (Revision 1936) zurück kehren.

random - das Thema wird bei jedem Laden zufällig aus der Liste der installierten Themen ausgewählt. Eine weitere Sache, die Sie bei der Auswahl eines Themas berücksichtigen sollten. Wenn Ihr Monitor eine Auflösung von z. B. 2560x1600 hat, dann sind die Symbole und Schriften auf dem Bildschirm zu klein dargestellt. Sie sollten ein großes Thema wählen. Beginnend mit Revision 4438 Clover wird dies automatisch tun. Auf einem kleinen Monitor verwendet es das Thema, sagen Sie metal, und auf dem großen Monitor wird versucht, das metal@2x-Thema zu finden, falls es eines gibt.

Das Kriterium für einen "großen" Monitor ist die Anzahl der Zeilen über 1100, denn bei mir auf einem 1080 Linien, reicht ein kleines Thema völlig aus. Die Frage für die Designer ist, ab wann sie große Analogien zu ihren Themen erstellen sollen. Ab 4862 werden skalierbare Themen erstellt. Das ist ein eigenes Kapitel. Was diesen Konfigurationspunkt betrifft, so steht er für Raster und Vektor-Themen, der einzige Unterschied ist der Name. Clover selbst unterscheidet das eine vom anderen durch

den Inhalt des Ordners. Wenn eine theme.plist vorhanden ist, handelt es sich um ein Bitmap-Thema. Wenn es eine theme.svg gibt, ist es ein Vektor-Thema.

```
<key>EmbeddedThemeType</key>  
<string>DayTime</string>.
```

Mögliche Varianten Dunkel und Hell. Eingeführt mit Revision 4644. Die Varianten wurden erfunden, bevor es die Erfindung der Uhr gab, Sie konnten ein dunkles oder helles eingebautes Thema wählen. Und jetzt nach Stunden, Licht am Tag, Dunkelheit in der Nacht. Eingeführt mit Revision 4773.

```
<key>Timezone</key>  
<integer>3</integer>
```

Mit der modernen Revision von Clover, 4773+, wurde das Konzept der "Tageszeit" eingeführt. Beim Starten von Clover. Es liest die aktuelle Zeit aus dem BIOS, die normalerweise die Greenwich Mean Time ist, und addiert dazu Wert aus diesem Parameter, nach der Zeit-Zone. Sie können mit einem +3-Zeichen oder, sagen wir, -5 schreiben. Zusammengefasst. So erhalten Sie die Ortszeit. Clover ermittelt dann, ob es zwischen 8:00 und 20:00 Uhr ist, dann zählt es als Tag, ansonsten als Nacht. Dies betrifft das bereits erwähnte eingebaute Thema, aber wenn DayTime angegeben ist, wirkt sich das sicherlich auf jedes Vektorthema aus. Um zu bestimmen, was es beeinflusst, es beeinflusst den Sound in jedem Thema. Tageszeitliche Wiedergabe des Tons aus der Datei sound.wav, bei Nacht aus der Datei sound_night.wav. Wenn kein Nachtton definiert ist, spielt es den Tageszeit-Ton.

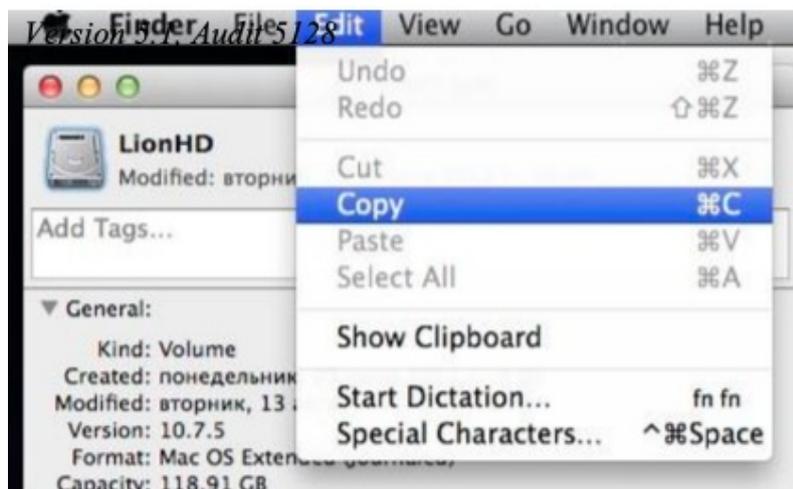
```
<key>PlayAsync</key>  
<false/>.
```

Ab der Revision 4833 erschien der Ton in Clover mit dem Treiber AudioDxe.efi, Autor Goldfish64, der sich nicht mit Clover befassen möchte, nun gut, es macht ihm nichts aus, das wir diese Lösungen benutzen. PlayAsync bestimmt, ob der Ton synchron mit dem Clover-Vorgang abgespielt wird, oder asynchron. Das heißt, bei synchroner Wiedergabe funktioniert nichts, bis der Ton enden wird. Bei asynchron fließt der Ton unabhängig davon, was auf dem Bildschirm passiert. Das heißt, Ton startet nach der Meldung auf dem Bildschirm ...scan entries... , dann erscheint die Oberfläche von Clover, der Ton läuft weiter, dann erscheinen die Zeilen der boot.efi Operation, der Ton läuft weiter, dann wird der Kernel gestartet, und die Kexte werden nacheinander gestartet. Ausgehend von einigen von ihnen, bricht der Klang ab . Dies gilt natürlich, wenn die sound.wav-Datei lang ist.

Clover 4862 hatte einen Fehler, asynchroner Sound hat den Computer aufgehängt. Ab 4870 gibt es kein Problem, können Sie PlayAsync=true einstellen.

```
<key>CustomIcons</key>  
<false/>.
```

Wenn auf <true/> gesetzt, wird jede Partition mit dem Betriebssystem suchen Sie im Stammverzeichnis der Partition nach dem Symbol **.VolumeIcon.icns** und verwenden Sie es anstelle von Symbolen, die durch das Thema festgelegt sind. Dieses Symbol ist sehr einfach mit MacOSX zu erstellen. Wählen Sie das Symbol des Laufwerks und machen Sie ein Copy-Paste für die Erstellung.



<key>screen resolution</key>

<string>1024x768</string>

können Sie die gewünschte Bildschirmauflösung größer als die Standardauflösung 1024x768 einstellen, wenn die Grafikkarte und die Bildschirm-Parameter einen solchen Modus haben. Clover versucht, die höchstmögliche Auflösung einzustellen, aber das kann falsch sein. Überprüfen Sie die Liste der verfügbaren Modi im Boot-Protokoll. Wenn im Grafikbereich PatchVBios=Yes steht, erhalten Sie die maximale Auflösung, der für diesen Monitor verfügbar ist. In diesem Fall kann der Parameter ScreenResolution unnötig sein. Bei einigen Konfigurationen kann der Parameter PatchVBios fatal sein – und zu einem schwarzen Bildschirm ohne Lebenszeichen führen.

Es gibt hier eine Feinabstimmung. Clover muss die EDID des Monitors kennen. Legacy-Clover versucht, es durch INT10-BIOS-Aufrufe zu erreichen, oft erfolgreich, manchmal nicht. UEFI Clover fordert Daten vom UEFI-BIOS an, das wahrscheinlich die EDID für die eingebettete Grafikkarte kennt, und wahrscheinlich nichts über die eingebaute weiß. Siehe preboot.log, und wenn dort kein EDID vorhanden ist, dann geben Sie ihn manuell ein. Anweisungen unten.

<key>Bildschirmauflösung</key>

<string>1024x768</string>

Hiermit können Sie die gewünschte Bildschirmauflösung einstellen, die höher als die Standardauflösung von 1024x768 ist, wenn die Grafikkarte und der Bildschirm selbst einen solchen Modus in ihren Parametern haben. Clover versucht, die höchstmögliche Auflösung einzustellen, aber das kann falsch sein. Prüfen Sie die Liste der verfügbaren Modi im Bootlog.

Wenn im Grafikbereich PatchVBios=Yes eingestellt ist, erhalten Sie die maximal verfügbare Auflösung für diesen Monitor. In diesem Fall kann der Parameter ScreenResolution überflüssig sein. Bei einigen Konfigurationen kann der Parameter PatchVBios fatal sein – es kann zu einem schwarzen Bildschirm ohne Lebenszeichen führen.

Es gibt hier eine Feinabstimmung. Der Clover muss die EDID des Monitors kennen. Legacy Clover versucht es über INT10-BIOS-Aufrufe zu erreichen, oft erfolgreich, manchmal nicht. UEFI Clover fragt das UEFI-BIOS ab, das wahrscheinlich die EDID für die eingebettete Grafikkarte kennt, für die eingebaute aber nicht. Sehen Sie sich preboot.log an und geben Sie die EDID manuell ein, wenn sie dort nicht vorhanden ist. Anweisungen unten.

<key>ProvideKonsoleGop</key>

<true/>.

(Neu in der Version 5128 s. a.:

https://uefi.org/sites/default/files/resources/UPFS11_P4_UEFI_GOP_AMD.pdf)

Erzeugt ein GOP-Protokoll für den Konsolenmodus, d. h. für eine Textausgabe nicht im Textmodus, wie Sie es vom PC-BIOS gewohnt sind, sondern im grafischen Modus, wie es Apple macht.

In einigen Revisionen von Clover wurde dieser Parameter durch den Quirk ProvideConsoleGopEnable dupliziert, aber seit Revision 5129 wurde dieser Parameter aus der Liste der Quirks entfernt und ist nur noch im GUI-Abschnitt mit einem etwas anderen Namen vorhanden. ProvideConsoleGop von der GUI überschreibt ProvideConsoleGopEnable aus der Liste der Quirks, falls Sie vergessen haben, diesen Parameter bei der Aktualisierung des Laders aus der Konfiguration zu entfernen.

Der Parameter sollte besser nicht ausgeschaltet werden, es gibt keine ausreichenden Argumente für das Ausschalten.

<key>KbdPrevLang</key>

<false/>

Aktivieren Sie diese Option, wenn Sie die Systemsprache beim Upgrade von macOS mit dem eingebauten NVRAM speichern möchten.

- ist der Schlüssel zur Behebung von macOS-Sprachproblemen, wenn Sie Ihr eigenes NVRAM verwenden.
- Dieser macOS-Fehler existiert schon seit langer Zeit.
- Fix funktioniert nur bei Verwendung von AptioMemoryFix oder OsxAptioFixV3 ohne EmuVariableuefi.efi, d. h. Sie benötigen ein Hardware-NVRAM
- Arbeiten in einer anderen Sprache als Englisch fixieren

Mit diesem Schlüssel können Sie die folgenden Probleme beheben:

```
<key>GUI</key>
  <dict>
    <key>KbdPrevLang</key>
    <true/>
    <key>Language</key>
    <string>ru:0</string>< Sie -----wollen Language
```

</dict>

Khaki Klee. Version 5.1, Revision 5129 Moskau, 2021

1. Gemischte Sprache im Neustart-Pop-up-Fenster nach macOS-Update.



2. Behalten Sie beim Upgrade von macOS im Recovery-Modus immer die englische Sprache bei, auch wenn der Benutzer bereits eine andere Sprache in macOS verwendet.

Zur Revision 4719 hinzugefügt:

<key>Language</key>

<string>ru:0</string>

Die Einstellung der Sprache ist derzeit nur für das Menü "Hilfe" sinnvoll, das mit der Taste F1 aufgerufen wird. Dieser Wert wird jedoch an das System gesendet und kann sich auf die Sprache in Standard auswirken.

<key>Mouse</key>

</dict>

<key>Enabled</key>

<true/>

<key>Speed</key>

<integer>2</integer>

<key>Mirror</key>

<false/>

<key>DoubleClick</key>

</dict> <integer>500</integer>

Enabled - es gibt Konfigurationen, bei denen die Maus nicht funktioniert oder sich aufhängt, dann kann sie verbannt werden.

Speed 2 - Geschwindigkeit der Cursorbewegung, sinnvolle Werte sind 2 - 8. Für einige Mäuse ist eine negative Geschwindigkeit erforderlich, die sich in die entgegengesetzte Richtung bewegt. Wert 0 bedeutet, dass die Maus deaktiviert ist.

Mirror - und auch die umgekehrte Richtung um nur eine Koordinate machen.

DoubleClick 500 - Pause in Millisekunden für die Doppelklick-Erkennung. **Wert 500 hat bisher jedem gepasst. Wurde in den neuen Revisionen entfernt.**

In der Clover-Oberfläche sehen Sie die Legacy- und Uefi-Loader für die installierten Betriebssysteme. In diesem Fall können sich mehrere Boot-Lader auf einer Partition befinden. Vielleicht brauchen Sie nicht alles, was Clover gefunden hat. Wenn Sie nur die von ihnen benötigten Systeme sehen möchten, können Sie sowohl einzelne Partitionen als auch ganze Klassen aus dem Clover-Menue ausblenden. Die folgenden Abschnitte befinden sich in der Konfiguration:

Hide - Blendet Volumes nach Namen oder nach ihrer UUID aus.

<key>Hide</key>

```

<array>
  <string>WindowsHDD</string>
  <string>BOOTX64.EFI</string>
  <string>E223FF7F-F2DA-4DBB-B765-756F2D95B0FE</string>
</array>

```

Khaki Clover. Version 5.1, Revision 5129
Moskau, 2021

44

Dies ist ein Array von Zeichenfolgen, wie sie im boot.log enthalten sind. So auf diese Weise können Sie unnötige Abschnitte wie Recovery oder Preboot aus dem Menü entfernen. Oder geben Sie alternativ an, was gescannt werden soll:

<key>Scan</key>

```

<dict>
  <key>Legacy</key>
  <string>First</string>
  <key>Entries</key>
  <true/>
  <key>Tool</key>
  <true/> <key>Linux</key>
  <false/>
</dict>

```

Für **Legacy** (d. h. von PBR gestartete Bootloader) gibt es Optionen für die Werte No, First, Last - überhaupt nichts anzeigen, an den Anfang der Liste setzen oder an das Ende.

Linux - suchen Sie nicht auf jeder Partition nach Linux-Bootloadern, das dauert sehr lange. Die Schnittstelle kann auch feiner abgestimmt werden, wenn Sie verstehen, wie und was getan werden muss. Für Sie nächsten Abschnitt (auch im Abschnitt GUI)

<key>Custom</key>

```

<dict>

```

Darin werden sind die Arrays enthalten:

<key>Entries</key>

```

<array>

```

<key>Legacy</key>

```

<array>

```

<key>Tool</key>

```

<array>

```

Ein Element des Arrays enthält eine Beschreibung des ausgewählten Elements in Form eines Wörterbuchs

```

<dict>
  <key>Volume</key>
  <string>454794AC-760D-46E8 .....2</string>
  <key>Type</key>
  <string>OSX</string>
  <key>Title</key>
  <string>OS X 10.8.5 (12F36) Mountain Lion</string>
  <key>InjectKexts</key>
  <true/>
  <key>NoCaches</key>
  <false/>
  <key>BootBgColor</key>
  <string>0x2C001EFF</string>
  <key>Hidden</key>
  <false/>
  <key>SubEntries</key> <array>
    <dict>
      <key>Title</key>

```

```

    <string>Boot OS X 10.8.5 </string></array>
  </dict>
  <key> AddArguments </key>
  <string> -v </string>
</dict>
</array>
</dict>

```

Clover in khakifarben. Version 5.1, Revision 5129
Moskau, 2021

45

Und jeder Menüeintrag kann mehrere Untereinträge (SubEntries) enthalten, die sind verschiedene Varianten des Hauptmitgliederaufrufs. Für InjectKexts gibt es möglicherweise eine Option Detect.

Sie können es auch auf eine Art von Eintrag legen.

Der für diese Parameter zuständige Code wurde von Apianti entwickelt.

Beschreibung der internen Keys:

- Disabled - deaktiviert die gesamte Struktur und sie wird nicht berücksichtigt
- Volume - Volume-Name oder GUID, die beim Scannen des Bootloaders verwendet werden soll
- VolumeType - Der Typ des Volumes, der bei der Aufnahme. Kann eine Zeichenkette eines Typs oder ein Array mit mehreren Typen sein. Gültige Volume-Typen: Intern, Extern, Optisch, FireWire
- Path - Ort, an dem nach dem efi-Bootloader gesucht werden soll
- Argumente - Argumente, die an den Bootloader übergeben werden. Überschreibt die Standard-Boot-Argumente in Boot / Argumente
- AddArguments - Fügt Standard-Boot-Argumente hinzu. Bei SubEntry fügt er Argumente zum Hauptsatz hinzu
- Titel - Ändert den angezeigten Titel. Verwendet das Format "Boot <Titel> von <VolumenName> "
- FullTitle - Setzt den angezeigten Titel nur auf "<FullTitle>" ohne jegliches Format
- Image - Der Pfad zur benutzerdefinierten Bilddatei. Der Suchpfad ist das Stammverzeichnis Volume-Verzeichnis, Theme-Verzeichnis, Clover-Verzeichnis, Clover-Volume-Root und schließlich OS-Icon-Namen
- ImageData - Eingebettetes benutzerdefiniertes Bild. Kann im PNG-Format sein oder BMP, definiert in der Konfiguration als binäres Array.
- DriveImage - Pfad zur benutzerdefinierten Laufwerksabbilddatei. Der Suchpfad ist das Stammverzeichnis des Aufzeichnungsvolumes, das Themes-Verzeichnis, das Clover-Verzeichnis, das Stammverzeichnis des Clover-Volumes und schließlich die OS-Symbolnamen
- DriveImageData - Eingebautes benutzerdefiniertes Laufwerksbild. Kann im PNG-Format oder BMP sein
- Hidden - Blendet den Eintrag aus. Wenn der Wert true ist, kann der Eintrag durch Drücken von F3 im Clover-Menü angezeigt werden. Wenn der Wert "Always" ist, wird der Eintrag nie angezeigt
- InjectKexts - Injiziert Kexts. Gültige Optionen sind Yes, No, or Detect. und auch auf einige Eintrag kann geliefert werden. Verwenden Sie Detect, um Kexte nur zu injizieren, wenn FakeSMC im KernelCache oder / S / L / E fehlt. Für Einträge vom Typ OSX, OSXInstaller und OSXRecovery.
- NoCaches - Caches überspringen. Für Einträge vom Typ OSX, OSXInstaller und OSXRecovery
- Kernel - Legt die Priorität der Überprüfung des Linux-Kernels fest. Wenn diese Option nicht angegeben wird, zeigt Clover alle gefundenen Kernel an. Gültige Optionen sind Newest, Oldest, First, Last, MostRecent, Earliest. Für LinuxKernel-Datensätze
- All - jeder Kernel Newest
- der neueste Oldest - der älteste
- der älteste
- First - zuerst gefunden Last – zuletzt gefunden
- MostRecent - Neueste Version Earliest - Älteste Version None - Nicht nach Kerneln suchen.
- •
- Typ - Der Typ des Betriebssystems, nach dem gesucht werden soll. Gültige Typen: OSX, OSXInstaller, OSXRecovery, Windows, Linux, LinuxKernel. Wenn der Typ nicht einer der aufgelisteten ist, wird er als **alle Typen (?)** betrachtet.
- BootBgColor - Legt die Hintergrundfarbe des Startbildschirms fest. Für OSX-Records, OSXInstaller und OSXRecovery
- CustomLogo - Legt den Logostil des Startbildschirms fest. Für weitere Informationen lesen Sie Boot / CustomLogo
- SubEntries - (Standard: true) Deaktiviert standardmäßig Untereinträge, wenn

auf false gesetzt sind. Kann auch verwendet werden, um benutzerdefinierte verschachtelte Datensätze zu erstellen, die die gleiche Struktur wie der Master-Datensatz verwenden; alle Einstellungen, die im verschachtelten Datensatz nicht gesetzt sind, werden vom Master-Datensatz übernommen

- KernelAndKextPatches - Verwendet die gleiche Hierarchie und bietet die gleiche Funktionalität wie KernelAndKextPatches, so dass Sie Patches selektiv auf jeden Eintrag anwenden können.

Funktioniert nur mit Version 2797 oder höher

- Ignore - Der Eintrag wird ignoriert bzw. nicht verwendet, also egal, was auf

key>ShowOptimus</key>

<true/>

Die Aufgabe war einfach. Mein BIOS schaltet spontan Optimus auf meinem Laptop aus, und ich möchte in der Lage sein, zu sehen, ob sie ein- oder ausgeschaltet ist, wenn Clover geladen wird, mit dem key korrigieren Sie die Situation.

Das Kriterium ist die Anzahl der Videokarten im System, bzw. der erste Intel. Intel + Diskret = Optimus.

Auf dem Bildschirm erscheint das Wort Intel Discrete only Auf dem Bildschirm erscheint das Wort "Diskret".

Sehr **praktisch!**(?)

Design: theme.plist

Nun das eigentliche Design entsprechend dem gewählten Thema. Die Datei theme.plist wird aus dem Themenordner geladen und ist für jedes einzigartig. Der Pfad für das Metallthema ist wie folgt:

```
/EFI/CLOVER/themes/metal/theme.plist
```

Die ersten Parameter des Themas das sind Texte, wie hier:

```
<key>Author</key>
<string>Slice</string>
<key>Year</key>
<string>2012</string> <key>Description</key>
<string>Main metallic looking theme</string>
```

Es folgt der Abschnitt mit den Auslegungsparametern

```
<key>Theme</key>
<dict>
```

Das Format aller genannten Bilder ist PNG, und Sie müssen den richtigen Titel haben.

Zum Beispiel speichert der Previewer Dateien im richtigen Format, aber nicht immer. Manchmal müssen Sie über Photoshop neu speichern.

Einige Oberflächenelemente können durch den folgenden Set ausgeschlossen werden:

<key>Components</key>

```
<dict>
  <key>Banner</key>
    <true/>
  <key>Functions</key>
    <true/>
  <key>Label</key>
    <true/>
  <key>Tools</key>
    <true/>
  <key>Revision</key>
    <true/>
  <key>MenuTitle</key>
    <true/>
  <key>MenuTitleImage</key>
    <true/>
  <key>Help</key>
    <false/>
</dict>
```

Wenn <true> , ist das Element vorhanden, sonst nicht.

Themenstil (beginnend ab Version 3586, author Needy)

```
<key>BootCampStyle</key>
<true/>
```

Standardmäßig wird der Refit-Text in eine separate Zeile geschrieben und kann sehr lang sein "Boot Recovery von RecoveryHDD". Aber in BootCamp, wie auch in Chameleon, ist es üblich, die Beschriftung direkt unter den Symbolen zu schreiben, aber nur kurz: MacOS, Linux, Windows, weil es sonst nicht passen würde. Das gilt aber nur, wenn es nur wenige Partitionen gibt und jede Partition ihren eigenen und einzigartigen Booteintrag hat. Wenn wir z. B. zwei Systeme haben, und eines davon ist geschlossen zu FileVault2, haben wir die folgenden Elemente

```
"Boot Recovery von RecoveryHDD"
"macOS von RecoveryHDD booten"
"macOS von SierraHDD booten"
```

Der Bootcamp-Stil ist hier irgendwie ungewöhnlich. Sie haben also die Wahl.

Der Hintergrund des Bildschirms:

```
<key>Background</key>
  </dict>
    <key>Typ</key>
    <string>Crop</string>
    <key>Path</key>
    <string>MetalBack.png</string>
    <key>Sharp</key>
    <string>0x80</string>
    <key>Dark</key>
    <true/>
  </dict>
```

Der Parameter Path gibt den Namen der Datei (genauer gesagt den Pfad!) an, in der sich das Hintergrundbild für den gesamten Bildschirm befindet. Dabei kann es vorkommen, dass der Bildschirm kleiner oder größer als das Bild ist, und dann, wird durch den Parameter

Type

Crop - zum Zuschneiden eines großen Bildes, damit es auf den Bildschirm passt, oder um es mit einem Hintergrund zu füllen.

Tile - Kacheln mit einem Mosaik aus Fliesen.

Skalieren - Proportional strecken, so dass das Bild den gesamten Bildschirm und mehr einnimmt, unter *Beschneiden*.

Gewöhnliches Dehnen erzeugt quadratische Pixel, so dass normalerweise wird eine gewisse Glättung vorgenommen, die jedoch die Kanten verfälscht.

In Clover wird die Kantenerkennung durchgeführt, ihr Wert wird durch den Parameter

Sharp bestimmt

Wenn er 0 ist- keine Erkennung, dann werden die Kanten verwischt. Maximalwert 0xFF = 255 - keine Unschärfe. 0x80 - erzeugt eine intelligente Unschärfe mit scharfen Kantenlinien. Auch im Paar Parameter arbeitet mit ihm

Dark

Wenn <true/> bedeutet, dass Sie ein dunkles Bild mit weißen Linien haben, <false/> bedeutet helles Bild mit dunklen Linien. Dies wirkt sich auf die Kantenerkennung aus.

```
<key>Banner</key>
  <string>logo-trans.png</string>
```

Banner ist ein zentrales Bild, es hat Größenbeschränkungen, abhängig von der Bildschirmgröße. Das dawn theme (Morgengrau)-Thema hat zum Beispiel eine Bildgröße von 672 × 190 Pixel. Dieser Wert kann als Maximum angesehen werden. Das Logo sollte entweder aus undurchsichtig, wenn wir keinen Hintergrund verwenden wollen. Dann wird die erste Pixel des Logos definiert die Hintergrundfarbe. Entweder hat das Logo ein blickdichtes Element auf der transparenten Hintergrund, und der gesamte Bildschirm wird vom Hintergrundbild bedeckt. Trick von Eps: Oben links macht das Pixel um 1 % undurchsichtig.

In den neuen Revisionen hat "Banner" dies Parameter:

```
<key>Banner</key>
</dict>
  <key>Pfad</key>
  <string>logo_trans.png</string>
  <key>ScreenEdgeX</key>
  <string>Links</string>
  <key>BildschirmrandY</key>
  <string>Oben</string>
  <key>DistanceFromScreenEdgeX%</key>
  <Ganzzahl>10</Ganzzahl>
  <key>DistanceFromScreenEdgeY%</key>
  <Ganzzahl>10</Ganzzahl>
  <key>NudgeX</key>
  <Ganzzahl>8</Ganzzahl>
  <key>NudgeY</key>
```

```
<Ganzzahl>5</Ganzzahl>
</dict>
```

Khaki Clover. Version 5.1, Revision 5129
Moskau, 2021

48

Path- Pfad zur Datei einschließlich Ordner, z. B. VariantA\Logo.png

ScreenEdgeX - Horizontaler Berichtspunkt (links/rechts/mittig)

DistanceFromScreenEdgeX - Position des Banners relativ zum Berichtspunkt, in Prozent um die Größe des Bildschirms. Dies gewährleistet eine korrekte Positionierung beim Wechsel Genehmigung.

NudgeX - 1% ist viel, bei einem 1920er Bildschirm sind es schon 19 Pixel, also in diesem Parameter geben wir in Einheiten von Pixeln an.

Ähnlich vertikal.

<key>Selection</key>

```
<dict>
  <key>Color</key>
  <string>0xF3F3F380</string>
  <key>Small</key>
  <string>Select_trans_small.png</string>
  <key>Big</key>
  <string>Select_trans_big.png</string>
  <key>OnTop</key> <true/>
</dict>
```

Color - Farbe der Linienauswahl im Menü. Der Künstler stellt die Farbe entsprechend der allgemeinen Farben den Farbton des Themas. Der Wert 0x11223380 bedeutet Farbe rot=0x11, grün=0x22, blau=0x33, alfa=0x80. Die letzte Zahl ist der Grad der Deckkraft, 0x80 entspricht 50%. 0x00 zeigt an keine Auswahl. 0xFF schließt das Hintergrundbild (Buchstaben auf einem undurchsichtigen Balken).

Groß und **Klein** sind Zeichnungen, die die Symbole im Hauptmenü in der oberen Zeile hervorheben – groß, und in der unteren, die kleinen.

OnTop - Position des Auswahlmusters (Roar 1983). Falsch - Auswahl unterhalb des Symbols Festplatten-Icon (traditionell für Refit), True - über dem Icon (traditionell für Chameleon).

<key>Font</key>

```
<dict>
  <key>Type</key>
  <string>Load</string>
  <key>Path</key> <string>BoG_LucidaConsole_10W_NA.png</string> <key>CharWidth</key>
  <integer>10</integer>
  <key>Proportional</key>
  <true/>
</dict>
```

- **Typ** – bedeutet die Schriftart. Es gibt zwei eingebaute Schriftarten Black und White (Rev. 3706+), und ein Dutzend, die geladen werden - Laden. In diesem Fall wird der Dateiname im folgenden Parameter angegeben Pfad – BoG_LucidaConsole_10W_NA.png. Für jedes Thema hat sein Autor die Schriftart gewählt, die am besten zu seiner Idee passt, siehe angehängte Datei. Für die Namen der Schriftarten (blackosx) haben sich folgende Konventionen durchgesetzt:

BoG - Black On Gray - Schwarz auf grauem Hintergrund. LucidaConsole - Name der Originalschriftart. 10W - Briefbreite

NA - Kein Antialiasing. Auch durchdacht.

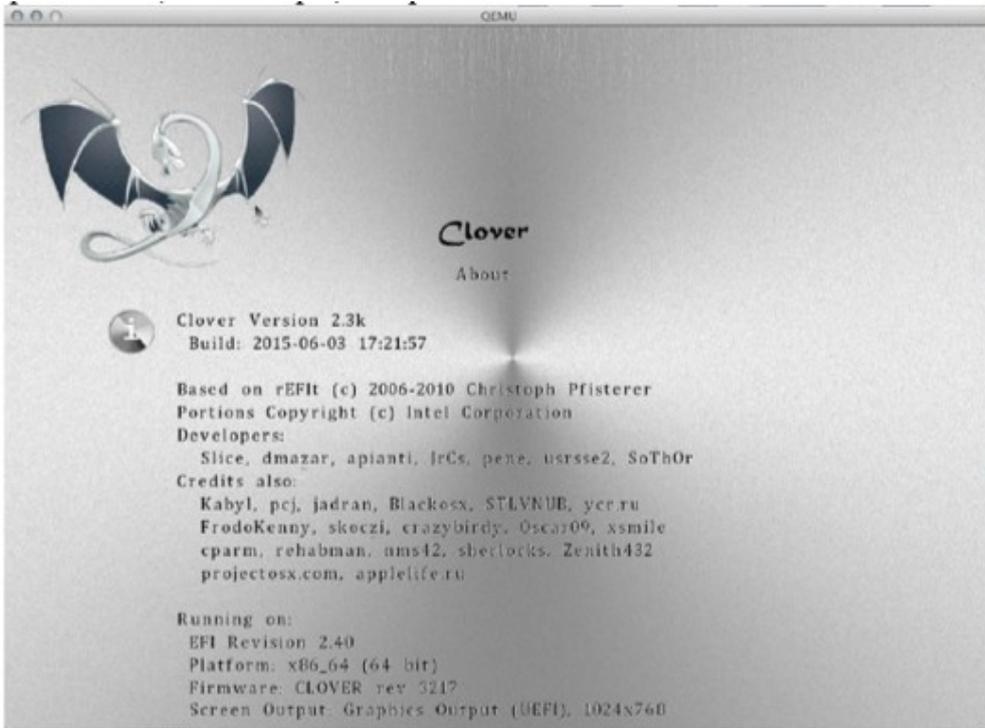
Die Größe eines Zeichens in der Datei beträgt 16 Pixel, aber die Zeichen selbst nehmen weniger Platz ein, Der nächste Parameter ist also die optimale Breite, und das wiederum, hängt von der Absicht des Autors ab.

Proportional - seit Revision 3217 ist es nun möglich, die Proportionalschriften zu verwenden. Für diejenigen, die es nicht wissen: Dies ist der Zeitpunkt, an dem das i einen großen

Khaki Clover. Version 5.1, Revision 5129
Moskau, 2021

49

weniger Platz als m. Eine Monospaced-Schriftart ist z. B. Courier, proportional wie z.B. Times. Allerdings ist Clover auch in der Lage, monospaced Schriften zu kondensieren, aber trotzdem, Das Ergebnis wird mit speziell zubereiteten Schriften besser sein. Hier ist zum Beispiel ein Bild



vergleichen Sie die Breite der Buchstaben i und m.

CharWidth 10 - Sie können die vom Autor der Schrift empfohlene Breite verwenden, oder Sie können um es nach Ihren Wünschen zu ändern. 9 - straffer, 11 - schlanker.

Ab Revision 3537 beeinflusst dieser Parameter die Breite des Textes im Optionsmenü, da der Parameter immer Proportional=false ist, und die Breite der Buchstaben wird vollständig durch diesen Parameter bestimmt, und in Zeile der Information ist der Text proportional, und wird dadurch komprimiert, dass für jeden Buchstaben, auch wenn die Breite in der Matrix viel größer ist.

<key>Badges</key>

```

<dict>
  <key>Show</key>
  <true/>
  <key>Inline</key>
  <true/>
  <key>Swap</key>
  <false/>
  <key>OffsetX</key>
  <integer>32</integer>
  <key>OffsetY</key>
  <integer>32</integer>
  <key>Scale</key>
  <integer>7</integer>
</dict>

```

Ein Badge ist ein kleines Bild in der rechten unteren Ecke des Hauptbildes. Ursprünglich ist das Hauptsymbol so gestaltet, dass es die Festplatte darstellt (wie in Bootcamp), und das Abzeichen zeigt an, welche es mit einem Betriebssystem gibt.

Anzeigen - ob das Badge angezeigt werden soll.

Swap - to change the meaning - Ändern Sie die Bedeutung des Symbols und des Abzeichens. Jetzt repräsentiert das Symbol das Betriebssystem, und das Abzeichen repräsentiert das Gerät (in diesem Fall ist es nicht interessant, es anzuzeigen).

Inline - zeigt das Abzeichen in der Zeile mit Informationen über das ausgewählte Symbol an. Hier ist es immer OS,unabhängig vom Parameter "Swap". Siehe Bildschirmfoto des iClover-Themas.

OffsetX und **OffsetY** sind Offsets des Badges von der oberen linken Ecke. Wenn keine Offsets angegeben werden, wird das Abzeichen in der unteren rechten Ecke platziert.

Scale - die Größe des Ausweises in Einheiten von X/16 der Originalgröße (7/16 im Beispiel). Das heißt, das Standardmotiv hat eine Größe von 48 Pixeln, was 6/16 des Standardmotivs- Icons entspricht.

<key>Scroll</key>

```
<dict>
  <key>Width</key>
  <integer>N</integer>
  <key>Height</key>
  <integer>N</integer>
  <key>BarHeight</key>
  <integer>N</integer>
  <key>ScrollHeight</key>
  <integer>N</integer>
</dict>
```

Da das Einstellungsmenü länger sein kann als die vertikale Größe des Bildschirms, wird in dem Menü eine Bildlaufleiste (Scroll) erscheinen, deren Parameter durch das Thema festgelegt werden, und es gibt Parameter standardmäßig für Bilder, die im Thema enthalten sind.

<key>Anime</key>

```
<array>
  <dict>
    <key>ID</key>
    <integer>1</integer>
    <key>Path</key> <string>logo_3D</string> <key>Frames</key> <integer>15</integer>
    <key>FrameTime</key> <integer>200</integer>
    <key>Once</key>
    <false/>
    <key>ScreenEdgeX</key>
    <string>left</string>
    <key>ScreenEdgeY</key>
    <string>top</string>
    <key>DistanceFromScreenEdgeX%</key>
    <integer>20</integer>
    <key>DistanceFromScreenEdgeY%</key>
    <integer>20</integer>
    <key>NudgeX</key>
    <integer>1</integer>
    <key>NudgeY</key>
    <integer>1</integer>
    <key>RelativeXPos</key>
    <string>50%</string>
    <key>RelativeYPos</key>
    <string>10%</string>
  </dict>
</array>
```

Animierte Bilder (Clips) können Teil des Themas sein. Die folgenden Serien werden unterstützt PNG-Bilder mit fortlaufenden Nummern. ID - definiert die Verwendung dieses Clips.

#Logo	(1)
#About	(2)
#Help	(3)
#Options	(4)
#Graphics	(5)

Clover in khakifarben. Version 5.1, Revision 5129

Moskau, 2021

51

#Binaries	(7)
#BOOT Sequence	(9)
#SMBIOS	(10)
#Drop ACPI Tables	(11)
#RC Scripts	(12)
#USB	(13)
#Themes	(14)
#Apple	(21)
#WinXP	(22)
#Clover	(23)
#Linux	(24)
#LinuxEFI	(25)
#BootX64.efi	(26)
#Windows UEFI	(27)
#Recovery	(30)

Kopfzeilenbilder in jedem Untermenü sind animiert, ebenso wie diese Animation wird es auf dem ausgewählten Hauptmenüpunkt wiedergegeben.

1-10 - Liste der vorhandenen Einstellungsuntermenüs.

21-27, 30-39 - dies ist das Detailmenü "Download-Optionen", das mit der Space auf dem Symbol im Hauptmenü oder durch einen Rechtsklick mit der Maus aufgerufen wird.



D.h. dieser Screenshot animiert einen Löwen, wenn die ID 37 eingestellt ist

Path- ML_Anim - Name der Animation, definiert den Namen des Ordners, in dem sich die einzelnen Animationen befinden

ML_Anim_000.png

ML_Anim_001.png

ML_Anim_008.png

ML_Anim_014.png

Bei fehlenden Frames wird das letzte gültige Frame verwendet, d. h. Als Die Rahmen 002-007 werden als Rahmen 001 und der Rahmen 008 wird als Rahmen 009-013 verwendet. Das ist praktisch, wenn sich die Handlung des Bildes in diesem Zeitraum nicht verändert.

Frames - 15 - Gesamtzahl der Frames in der Animation. Die fehlenden werden aufgefüllt durch im obigen Algorithmus.

FrameTime - 100 - Zeitintervall zwischen den Frames in ms. Variables Intervall wird mit Hilfe von übersprungenen Frames realisiert.

Once - wenn `<true/>` angegeben ist, wird die Animation nur einmal abgespielt, bevor sie beendet wird des Hauptmenüs (klicken Sie mit der rechten Maustaste in den milchigen Bereich auf dem Hauptbildschirm, oder drücken Sie die Escape). Wenn `<false/>` angegeben ist, wird die Animation in einer Endlosschleife abgespielt, für das letzte Bild ist nach dem gleichen Intervall Null, ohne zusätzliche Pause.

ScreenEdgeX - Horizontaler Berichtspunkt (links/rechts/mittig)

DistanceFromScreenEdgeX - Filmposition relativ zum Berichtspunkt, in Prozent für die Größe des Bildschirms. Dies gewährleistet eine korrekte Positionierung beim Wechsel .

NudgeX - 1% ist viel, bei einem 1920er Bildschirm sind es schon 19 Pixel, also in diesem Parameter geben wir in Einheiten von Pixeln an.

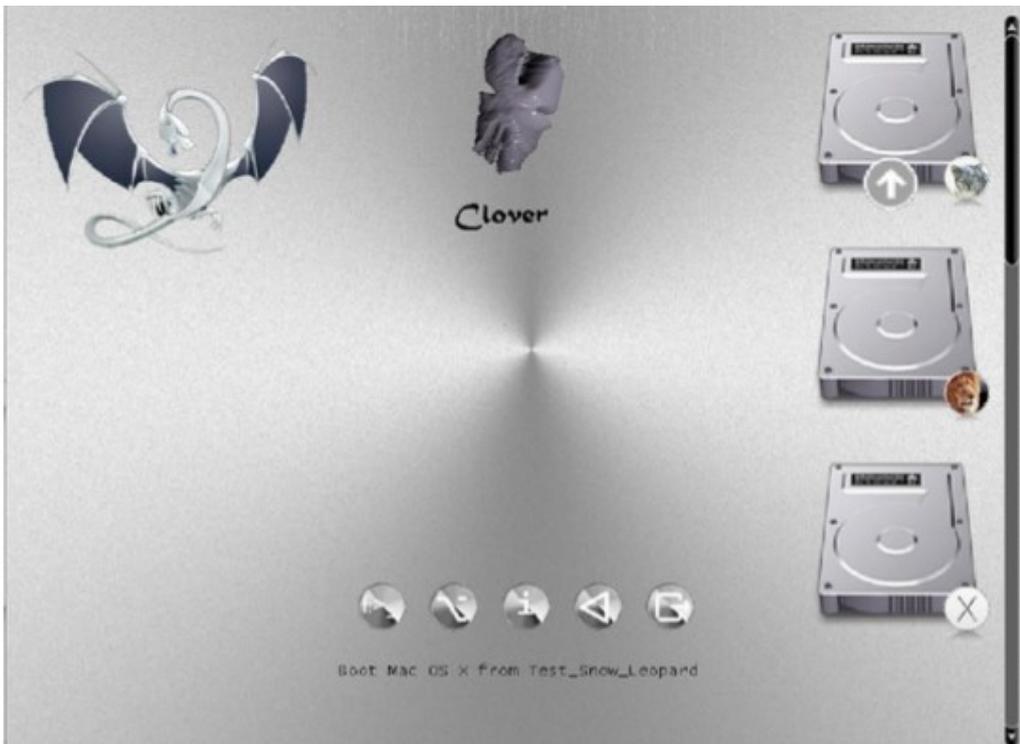
In den letzten Überarbeitungen haben sie begonnen, mit der Änderung des Layouts selbst zu experimentieren

```
<key>Origation</key>
  <dict>
    <key>DesignWidth</key>
      <integer>1920</integer>
    <key>DesignHeight</key>
      <integer>1080</integer>
```

Mit diesen Parametern legen wir fest, für welche Bildschirmauflösung das Thema ursprünglich konzipiert wurde, um die Anordnung der Elemente bei einer anderen Auflösung korrekt neu zu berechnen.

Ein großer Abschnitt über den Standort selbst.

```
<key>Layout</key>
  <dict>
    <key>Vertical</key>
      <true>
```



<key>BannerOffset</key>
 <integer>80</integer>

Dies ist der Abstand vom Banner zum Hauptmenü, manchmal müssen Sie ihn ersetzen, damit der Animations-Banner nicht mit den Symbolen des Hauptmenüs überlappt.

Ähnlich

<key>ButtonOffset</key>
 <integer>20</integer>

<key>TextOffset</key>
 <integer>30</integer>

<key>AnimAdjustForMenuX</key>
 <integer>30</integer>

Sie können auch die Symbole des Hauptmenüs skalieren

<key>MainEntriesSize</key>
 <integer>200</integer>

Der Standardwert ist 128, wie zuvor.

Mit dem Wechsel der Symbole können Sie auch die Abstände zwischen den Symbolen ändern

<key>TileXSpace</key>
 <integer>20</integer>

<key>TileYSpace</key>
 <integer>20</integer>

Sie können auch die Größe der Auswahl ändern.

<key> SelectionBigWidth </key>
 <integer> 288 </integer>

Der Standardwert ist 144. Es trifft zu, wenn sich die Auswahl im Hintergrund befindet.

Vector themes

Warum werden sie benötigt

Die Entscheidung, das GUI-Design von Clover auf Basis von Vektorgrafiken zu erstellen, kam zustande da es Beschwerden eines Anwenders über den Betrieb vom selben Flash-Laufwerk auf verschiedenen Computer gab, einer hat einen kleinen Bildschirm und dann passt das große Thema einfach nicht, und der andere großen Bildschirm, und dann sieht das normale Thema zu klein aus, und der Text ist überhaupt unlesbar. Wie kann man das kombinieren? Schnell wurde eine Variante erstellt, ein Theme mit dem Präfix @2x. Dann gibt es, in der Clover-Konfiguration das Metallthema festgelegt wird, und je nach Monitorauflösung entweder Metall oder [metal@2x](#) geladen. Der Nachteil dieses Ansatzes ist, dass sich niemand überstürzt um doppelte Themen zu erstellen. Und im Allgemeinen ist es nicht allzu sympathisch ausgefallen.

Dann hatte ich die Idee, wirklich ein skalierbares Thema zu machen das auf Vektor-Graphik basiert. Für diejenigen, die den Punkt überhaupt nicht verstehen, ein Beispiel: Ein Kreis kann dargestellt werden als PNG-Datei, die eine vollständig definierte Größe in Pixeln hat, oder Sie können Anweisung `<circle cx="100" cy="200" r="50">` und lassen Sie das Zeichenprogramm Folgendes tun Kreis auf dem Bildschirm mit einer solchen Position und einem solchen Radius in abstrakten Einheiten, die entsprechend der Bildschirmgröße auf die richtige Pixelanzahl umgerechnet. Im Original es könnte z.B. eine Graphikmaschine sein, die in Millimetern arbeitet.

Mit Kreisen und Quadraten allein können Sie kein gutes Design erstellen, daher hier Sie benötigen eine Art Vektorgrafik-Standard mit einer großen Anzahl von Funktionen.

Für ein offenes Projekt wie Clover ist natürlich ein offener Standard notwendig, nun dokumentiert und durch Zeichenprogramme unterstützt. Die Auswahl ist eindeutig - SVG-Grafiken.

https://en.wikipedia.org/wiki/Scalable_Vector_Graphics

Jetzt ist die nächste Frage, wie mache ich, das es unterstützend wird? Ich meine, es muss doch diese Befehle zu interpretieren und entsprechend etwas zu zeichnen ist. Das ist gelinde gesagt, nicht einfach zu lösende Aufgabe. Und dann fand ich das nanosvg-Projekt, <https://github.com/memononen/nanosvg> Es ist optional, erlaubt Ihnen, sich zu kopieren, es enthält nur zwei C-Dateien, dann gibt es fast keine Änderungen, die verwendet werden können. Der Autor hat die Entwicklung gestoppt, obwohl zahlreiche Verbesserungsvorschläge, wie es so gemacht wird gab. Verwenden Sie es so wie es ist. Aber das ist mir nicht genug, ich brauche mehr. Außerdem reicht fast keine Änderung mehr aus, in Es ist elementar nicht möglich, die Standard-C-Funktionen `sin()`, `malloc()` zu verwenden, `scanf()`, `qsort()` und so weiter, musste ich noch deren Implementierung innerhalb von der Möglichkeiten des EDK2 machen. Ich habe das getan und mir auch Pool-Anfragen angesehen, mit sehr guten mit Verbesserungsvorschlägen und nahm auch eigene Ergänzungen vor. Also, die Vektorbilder kann Ich rastern lassen und auf dem Bildschirm anzeigen. Jetzt müssen sie verwendet werden, um skalierbares Thema zu erzeugen. Vorzugsweise, ohne die Unterstützung für vorhandene PNG-Themen zu beeinträchtigen. Aber Theorie ohne Praxis ist tot, ich kann keinen Themensupport machen, wenn ich keine eine Probe von einem habe.

Die Theorie ist folgende, in der ersten Stufe. Lassen Sie das gesamte Thema als eine einzige Datei mit dem Namen `theme.svg`, die jeder Viewer, wie z. B. GoogleChrome, Safari oder einfach durch das Leerzeichen-Taste angezeigt wird. QuickLook, wird wie die Clover-Oberfläche aussehen. Und schon Clover wird sich mit dieser Datei befassen, sie in Symbole zerlegen und diese ausgeben, als Ausgaben Raster-Thema. Dank an Clovy. Er nahm die Herausforderung an und begann, eine solche Datei zu zeichnen, und während wir arbeiteten, fanden wir bereits gemeinsam heraus, was in Clover getan werden muss und wie es sein sollte. um eine Feile zu bilden, so dass die Enden zusammenpassen. Darüber im Forum *.How to make a vector theme in Adobe Illustrator*

Die Skalierung des Themas bedeutet, dass die Oberfläche auf jeder Bildschirmgröße gleich aussehen, werden die Symbole den gleichen Prozentsatz der Bildschirmgröße einnehmen, in Sinn der Bildschirmhöhe. Das Thema ist nicht in der Breite skalierbar, zugunsten der Beibehaltung Quadrate. Das heißt, auf dem Bildschirmen mit unterschiedlichem Seitenverhältnis wird das Bild anders aussehen, streng genommen gesagt, aber ein Kreis ist immer noch ein Kreis und ein Quadrat ist immer noch ein Quadrat.

Vektorthemen haben immer noch den Vorteil, dass sie Schriften verwenden. Im Raster. Thema gab es nur eine Schriftart, irgendwie proportional gemacht oder nicht, während im Vektor kann es mehrere verschiedene Schriftarten sein, wirklich proportional, und sehr glatt. Eine weitere Verfeinerung ist in Zukunft möglich, z. B. mit der Animation entfalten sich die Perspektiven im Allgemeinen. Obwohl ja, ich verstehe für diejenigen, die Timeout=0, oder TextOnly, oder eingebettet setzen, dieses Kapitel ist überhaupt nicht interessant.

Wie man ein Vektordesign erstellt

Zuerst müssen Sie ein Zeichenprogramm einsetzen, mit dem Sie vertraut sind. Um zu Ihre Grafik zu zeichnen, und die diese Zeichnung in das SVG-Format Version 1.1 exportieren. Obwohl mit mit einigen Verbesserungen gegenüber diesem Standard. Es gibt Optionen, Sie können Googeln, was es auf dem Markt gibt. Mein kurzer Überblick

(1) Adobe Illustrator. Je höher die Version, desto besser. Dies ist, was ich für die Kompatibilität mit gesucht in erster Linie. Sie ist komfortabel und funktional.

2. inkscape. Der Vorteil Nummer 1 ist, dass es kostenlos und damit legal ist. SVG sein natives Format, aber es hat eine Reihe eigener SVG-Erweiterungen, die nicht unterstützt werden von niemandem sonst. Juhu! Version 1.0 funktioniert jetzt unter macOS 10.11 bis 10.15!

3. libreOffice Draw. Ebenfalls kostenlos und seriös. Aber irgendwie klobig und die resultierenden SVGs sind nicht sehr gut. Unterstützt keine eingebetteten SVG-Schriften.

4. CorelDraw. Ich weiß es einfach nicht, ich habe es nicht benutzt.

5. BoxySVG. Nettes, einfaches und preiswertes Programm, mit dem Sie einzelne SVG-Zeichnungen bearbeiten und sofortige Kontrolle über den erstellten Code haben, im Gegensatz zu ihren älteren Brüdern, die nur beim endgültigen Export Code erzeugen. Ein Nachteil des Programms ist, dass es keine Möglichkeit gibt, das Problem zu lösen (*ich will das sehen, wie das zu tun?*). Im Kronleuchter ist es klar, Sie zeichnen, und Sie sehen, was Sie bekommen, wenn Sie zeichnen, durch was die Bearbeitung angeht, und man muss eine ziemlich gute Vorstellungskraft haben, um das Ergebnis zu erraten. Ich werde keine anderen Programme berücksichtigen, es liegt in Ihrem Ermessen, folgen Sie einfach dem Standard. Version 1.1, und die folgenden Empfehlungen.

Zweitens muss das Bild theme.svg aus klar definierten Komponenten bestehen mit ganz bestimmten Namen, da Clover die Zeichnungen dort namentlich auswählt.

Vollständige Liste:

Background, Banner, selection_big, selection_small, selection_indicator, pointer, scrollbar_background, scrollbar_holder, checkbox, checkbox_checked, radio_button, radio_button_selected,

durch einen Fehler wurde die Bildlaufleiste mit einem Minuszeichen statt einem Unterstrich definiert, da das in Rasterthemen richtig ist. Das wurde in Vektorthemen korrigiert.

Festplatten-Symbole

vol_internal - internes Festplattenlaufwerk;

vol_external - externe Medien (normalerweise USB);

vol_optical - CD/DVD-Laufwerk;

vol_clover - Bootloader-Diskette (*nicht verwendet?*)

vol_internal_hfs - Festplattenpartition mit HFS+-Dateisystem;

vol_internal_apfs - Festplattenpartition mit APFS-Dateisystem;

vol_internal_ntfs - Festplattenpartition mit NTFS-Dateisystem;

vol_internal_ext3 - Festplattenpartition mit EXT-Dateisystem;
vol_recovery - Partition von Recovery, RecoveryHD und anderen Laufwerken

Betriebssystem-Symbole

os_Clover

os_legacy

os_unbekannt

os_tiger Mac OS X 10.4 Tiger

os_leo Mac OS X 10.5 Leopard

os_snow Mac OS X 10.6 Snow Leopard

os_lion Mac OS X 10.7 Lion

os_cougar OS X 10.8 Mountain Lion

os_mav OS X 10.9 Mavericks

os_yos OS X 10.10 Yosemite

os_cap OS X 10.11 El Capitan

os_sierra macOS 10.12 Sierra

os_hsierra macOS 10.13 High Sierra

os_moja macOS 10.14 Mojave

os_cata macOS 10.15 Catalina

os_bigsur macOS 10.16/11.0 Big Sur

os_win Obligatorisch, wird verwendet, wenn es kein Symbol für das aktuelle Betriebssystem Windows gibt (bei NTFS-Dateisystem-Volume)

os_vista Windows Vista, Windows 7, Windows 8, Windows 10

os_freedos

os_freebsd

os_linux Obligatorisch, wird verwendet, wenn es kein Symbol für das aktuelle Betriebssystem Linux gibt (auf dem Volume mit EXT-Dateisystem)

os_ubuntu

os_suse

der Rest der Linuxe wird vielleicht eines Tages aufgelistet werden.

Service-Symbole der zweiten Reihe

tool_shell, func_clover, func_options, func_about, func_reset and func_shutdown, and func_help die angezeigt wird, wenn die Hilfe mit F1 aufgerufen wird.

Drittens spielt die Position dieser Symbole innerhalb der Datei theme.svg eine Rolle für die Vorschauen, aber nicht für Clover selbst. Außer bei Banner. Dies ist das mittlere Bild, und es ist nicht nur an einer festen Stelle in der Mitte des Bildschirms, wie es in Rasterthemen wird das Banner nun so auf dem Bildschirm positioniert, wie es im Thema vorgesehen ist. Beispiel wie im Cäsium-Thema gesehen werden kann. Weder die Blume noch der Drache sind zentriert.

Noch etwas: Alle Arten von Schaltflächen und Bildläufen sind in der Regel bei Voransichten unnötig, ebenso wie zahlreiche Achsen und Festplattensymbole, so dass sie aus den Vorschauen ausgeblendet werden sollten, aber links verfügbar für Clover sind. Dabei hilft das Attribut `visibility="hidden"`, das allerdings Clover ignoriert, wenn es dieses Symbol auf den Bildschirm zeichnen muss.

Vektorthemen unterstützen die neue Funktion "Nachtdesign".

Die eingebaute Uhr sagt Clover, wie spät es ist, berechnet sie lokal neu

Zeit, wobei die Zeit von 8:00 bis 20:00 Uhr als Tageszeit und der Rest als Nachtzeit zählt. Darüber ein wenig später.

Vektorthemen können alle Symbole mit einem an den Namen angehängten `_night` dupliziert werden. Zum Beispiel können Sie neben dem Hauptbanner mit dem Namen "Banner" ein zweites Bild mit dem Namen "Banner_night". Und so mit allen Symbolen, z. B. "pointer" zu "pointer_night", "os_mac" auf "os_mac_night". Sie können sich einfach in der Helligkeit unterscheiden, oder sie können völlig anders sein. Wir haben also eigentlich zwei verschiedene Themen in einem, eins wird tagsüber angezeigt, das zweite in der Nacht.

Jedes Symbol muss auch ein unsichtbares Rechteck enthalten, um seine Grenzen zu definieren. Der Grund dafür ist einfach, es kann weniger Inhalt in einem Symbol enthalten sein als seine Größe, und Clover muss seine volle Größe kennen, um skaliert zu werden. Name eines solchen Rechtecks enthält notwendigerweise "BoundingRect_", aber da die Namen eindeutig sein, beliebige andere Zeichen, nur Zahlen, oder der vollständige Name des Icons.

Beispiel:

```
<g id="tool_shell" transform="translate(300, 600)">
  <rect visibility="hidden" id="BoundingRect_ts" y="0" width="64" height="64"/>

  <g transform="matrix(0.55 0 0 0.55 15 10)">
    <use xlink:href="#knopka" width="85.5" height="85.5" />
  </g>
  <g transform="matrix(0.55 0 0 0.55 25 43)">
<text class="st35 st36 st38 st39">$_</text>
</g>
</g>
```

Entschlüsseln Sie Wort für Wort.

<g ...> ist eine Gruppe von Bildern, die ein einzelnes Übersichtsbild bilden. Sein Name ist "tool_shell", das Clover verwendet Shell-Symbole in der zweiten Reihe zum Zeichnen.

transform="translate(300, 600)" - legt den Versatz des Symbols in der Vorschau fest. Clover ignoriert es.

Die zweite Zeile <rect...> ist das unsichtbare Rechteck des Dienstes, das angibt Größe des Symbols, in unserem Fall 64x64. Die Maßeinheit ist Pixel, im Vergleich zu der Größe des Themes selbst, die 1600x900 beträgt. Entsprechend wird auf einem Bildschirm mit einer Höhe von 1800 das Symbol wäre z.B. 128x128.

In der dritten Zeile gibt es wieder eine Zusammenführung von Bildern und eine allgemeine Transformation: Größe um in Bezug auf die oben genannten Grenzen und die Verschiebung innerhalb dieser Grenzen.

Tatsache ist, dass in Grafikeditor können Sie bequem in einem anderen Maßstab zeichnen, aber dann müssen Sie einzelne Symbole auf den Maßstab des gesamten Themas skalieren.

<use xlink:href="#knopka" width="85.5" height="85.5" /> - dies war nicht da im Originalprojekt von nanosvg, und es ist wahrscheinlich von SVG2.0 - Symbolunterstützung. Der Punkt ist, das gleiche Objekt nicht mehrmals zu zeichnen, sondern nur das gleiche Bild. Meine Schaltflächen der zweiten Reihe verwenden alle das gleiche Bild mit dem Namen "knopka". Und das nächste Bild mit seiner eigenen Transformation wird darüber gelegt, und als seinen aus zwei Buchstaben bestehenden Text, z. B. "\$_". Die Klassen dieses Textes definieren Schriftart, Größe, Füllfarbe und Strichfarbe. Die Klassen sind in der Datei selbst definiert theme.svg, irgendwo am Anfang.

Es gibt ein Problem mit der Transformation in Lustra, wahrscheinlich das gleiche wie in anderen Vektor-Editoren. Es neigt dazu, die Koordinaten aller internen Bildobjekte neu zu berechnen, anstatt um das Transformationsattribut außerhalb der Gruppe zu lassen. Ich habe nicht gelernt, damit umzugehen, also habe ich alle Objekte am Anfang der Koordinaten gezeichnet und dann den Maßstab und den Offset vorgegeben nur in einer Textverarbeitung. Vielleicht hat jemand mehr Glück mit dieser Übung. Warum spielt das eine Rolle? Doch, denn entweder in Clover oder in den Vorschauen fangen die Teile an, sich in verschiedenen Richtungen zu verteilen, und das Einfangen mit Illustrator-Werkzeugen verdirbt das ganze Bild noch mehr.

Unterstützung für den [SVG-Standard in Clover](#)

Alles andere als perfekt. Aber wir können sehen, dass ein großer Teil von dem Illustrator funktioniert wie erwartet.

Shapes: <Rechteck>, <Kreis>, <Ellipse>, <Polylinie>, <Polygon> und nur <Pfad>. Diese Formen können zu Gruppen <g> und Symbolen <Symbol> zusammengefasst werden. Formen können zu Gruppen <g> und Symbolen <Symbol> zusammengefasst werden. Jede Zusammenfassung kann den Namen und die Transformation enthalten

Eine Gruppe ist eine statische Vereinigung, wie sie definiert und angezeigt wird. Für Clover sind die Gruppennamen von Bedeutung, da sie das Bild in Symbole unterteilen, die sie dann verwenden, um ein GUI zu zeichnen.

So definiert `<g id="vol_internal">` eine Gruppe von Figuren, die zusammen ein Bild ergeben interne Festplatte. Innerhalb dieser Gruppe wird das obligatorische Element `<rect id="BoundingRect_001" class="st0" width="128" height="128"/>`.

Sie wird benötigt, um die Größe der Gruppe zu bestimmen, denn für Clover, um dieses Bild korrekt zu positionieren, für die inneren Zeichnungen sind wahrscheinlich kleiner als die Gesamtgröße. Der Name jedes Elements ist eindeutig in einem Design, daher sollte der Name eines solchen Rechtecks BoundingRect enthalten, und beliebige zusätzliche Zeichen für Eindeutigkeit haben. Dies widerspricht nicht dem SVG-Standard, sondern Clover muss die einzelnen Elemente manipulieren.

Das Symbol ist eine dynamische Assoziation. Die Browser zeigen keine Zeichen an, die durchaus im Thema ist dieser Satz von Objekten eine Voreinstellung, die von anderen Objekten verwendet werden kann. Sie wird vom `<use>`-Operator verwendet

Beispiel:

```
<symbol id="HDIconBase" viewBox="-64 -64 128 128">
  <rect id="BoundingRect_3_" x="-64" y="-64" class="st0" width="128" height="128"/>
  <path id="bottom_3_" class="st1" d="M26,49h-52c-3.9,0-7-3.1-7-7v-66c0-3.9,3.1-7,7-
7h52c3.9,0,7,3.1,7,7v66C33,45.9,29.9,49,26,49
z"/>
  <path id="top_3_" class="st2" d="M26,47h-52c-2.8,0-5-2.2-5v-55c0-2.8,2.2-5,5-
5h52c2.8,0,5,2.2,5,5v55C31,44.8,28.8,47,26,47z"
/>
</symbol>
```

Hier wird ein Symbol definiert, das aus zwei geschlossenen Kurven (Weg) und einem Dienst Platz. Stile/Klassen sind in der Datei definiert, ich werde sie hier der Kürze halber weglassen. Zeichnen Sie diese Figur innerhalb des Symbols

```
<g id="osx_moj">
  <rect id="BoundingRect-2" class="st0" width="128" height="128"/>

  <use xlink:href="#HDIconBase" width="128" height="128" x="-64" y="-64"
transform="matrix(1 0 0 -1 64.0876 64)" style="overflow:visible;"/>
xlink:href ist ein Link und zeigt auf ein Zeichen namens HDIconBase.
```

Anders als der SVG-Standard versteht Clover keine Gruppenreferenzen, sondern nur Symbole. Leider und obacht, aber das ist etwas, das Sie bei der Erstellung Ihres Designs im Hinterkopf behalten sollten. Das heißt, Sie verschieben die Gruppe auf das Symbol, und nehmen Sie Bezug auf dieses Symbol.

Fillings (fill) and strokes (stroke).

Ich kann nichts Genaues über den Umriss sagen. Sie wurde vom Ersteller des Projekts erstellt nanosvg Mikko Memononen, scheint vollständig mit dem Standard konform zu sein, und unterstützt sogar Strichlinie. Aber lassen Sie uns über Füllungen sprechen.

Eine einfache Variante ist eine homogene Farbfüllung. Die Farbe wird über 16 RGB-Konstanten eingestellt: Füllung: "#ffcc00" ist orange. Es gibt sowohl reale Zahlen als auch Prozentwerte.

Oberhalb des Standards in Clover die Farbnamen: fill: "white" und viele weitere Namen, die ich nicht aufführe, um 100 Wörter zu kopieren. Verwenden Sie den SVG-Standard, und schauen Sie gegebenenfalls in den Quelltext von Clover.

Eine andere Art von Füllung sind Farbverläufe.

linearGradient - hier 100%ige Einhaltung der Norm.

radialGradient - leider wird der Fokuspunkt noch nicht unterstützt, daher Der radiale Farbverlauf in Clover sieht möglicherweise nicht genau so aus, wie er entworfen wurde. **conicGradient** - dies ist (noch?) nicht im Standard enthalten. Diskussion im Internet geht weiter, aufstellen sehr interessante Zeichnungen mit dieser Methode gemacht, aber das W3C-Konsortium hat noch nicht die Aufnahme in den Standard in Betracht gezogen, und folglich unterstützen die Browser dies nicht.

Und wie kann ich nicht unterstützen, wenn mein Metall-Thema auf konischen Farbverläufen aufgebaut ist! jUnd jetzt ist das Thema Cäsium! Eine andere Frage: Wie habe ich es denn gezeichnet? In Photoshop kann man es das tun. Adobe hat diese Funktion nicht in Illustrator integriert, wahrscheinlich in Erwartung von Standardisierung. Syntaktisch ist es dasselbe wie radialGradient, nur die Farbe wird nicht durch Radius und nach Winkel, wobei Winkel=0 gleich Anschlag="0.0" und Winkel 360 gleich Anschlag="1.0" ist. Und... Zwischenwerte so viel wie Sie wollen. Wenn Sie den Startwinkel drehen wollen, gibt es gradientTransform="rotate(45)" und andere Transformationen wie in der Norm.

Für das Design können Sie einen radialen Farbverlauf erstellen und dann einen Texteditor verwenden, um die in eine verjüngte. Vorschau, leider nur in Clover verfügbar. Wenn jemand einen zweiten Browser mit Kegelunterstützung findet, lassen Sie es mich wissen

Dithering. Und es steht auch nicht in der Norm, obwohl es meiner Meinung nach darum bettelt. Kurze Theorie, für diejenigen, die es noch nie gehört haben. Wenn sich in einem Farbverlauf eine Reihe von Punkten mit einer Helligkeit von 150 befindet, gefolgt von Helligkeit 151, wird eine Stufe sichtbar, aber es gibt keine Zwischenwerte (erkennbar am Motiv Clovy-Hintergrund mit Kreisen bei frühen Versionen von Clover, (keine Dithering-Unterstützung). Die Methode ist, statt des gewünschten Helligkeitspunktes 150 entweder 151 oder 150 auszugeben 151.3 zufällig, mit einer Wahrscheinlichkeit von 0,3. Wenn die Anzahl der Punkte groß ist, wird sie summiert in 150,3!

Der Standard hat keine Beispielsyntax für diesen Fall, also haben wir unsere eigene entwickelt. Und es benötigen einen clover namespace (Clover-Namensraum).

Code:

```
<radialGradient clover:ditherCoarse="16" id="GrayRadialBackground_5_" cx="441.2867"
cy="0.7502" r="1.0023" gradientTransform="matrix(5.400000e-14 768 -874.24 4.700000e-14
1338.8547 -338690.875)" gradientUnits="userSpaceOnUse">
<stop offset="0" style="stop-color:#7C7C7C"/>
<stop offset="1" style="stop-color:#5E5E5E"/>
</radialGradient>
```

Digitaler Parameter coarse - Grobheit, definiert die Länge des Blickwinkels, zwischen was und was zu wählen. Für große Zeichnungen wurde als Hintergrund ein Wert von 16 gewählt. Für Icons reicht 1, und so sieht es sehr glatt und fast unsichtbar aus. Standardmäßig ist der Wert=0, was bedeutet, dass die Methode abgebrochen wird.

Die Abbildungen zeige ich aber später, sie waren zu unterschiedlichen Zeiten [auf dem Schlitten \(?\)](#).

Fill with a pattern

Hier sind wir noch sehr weit von einer vollständigen Implementierung entfernt. In der Norm wird das Bild wiederholt so oft, wie es zum Füllen der zu füllenden Figur notwendig ist. In Clover ist nur eine Option:

Die Vorlage ist ein PNG-Bild und füllt den gewünschten Umriss maßstabsgetreu aus, um ihn vollständig zu füllen. Dabei wird das Rasterbild skaliert mit Glättung durch Dithering-Algorithmus.

Wozu ist das gut? Der Punkt ist, dass viele der Apple-Logos Bilder sind, d.h. Raster, und wir haben die Optionen: das Raster zu vektorisieren (nicht sehr schön), zu zeichnen etwas völlig anderes, oder binden Sie das Raster in den Vektor ein. Ich habe beides gemacht.

Illustrator hat diese Funktion

1. Fügen Sie ein PNG-Bild in den Entwurf ein. (einbetten).
2. Übersetzen Sie die Form in ein Muster
3. Legen Sie in den Eigenschaften der zu füllenden Figur fest, dass sie mit einem Muster mit so-und-so ID gefüllt werden soll.

Das einzige Problem ist, dass die Abbildung nicht so wiedergegeben wird, wie es nach der Spezifikation sein sollte, sondern es füllte den gesamten Raum, skaliert auf die Maße der Leinwand.

Oh, ja, es gibt auch eine Einschränkung

```
<pattern id="pattern_1234">
```

Der Pattern-Identifikator muss nach den Regeln von Clover das Wort Pattern enthalten, ansonsten hat Clover keine Möglichkeit zu wissen, womit wir die Form füllen. Die Verwendung eingebetteter PNGs macht nicht viel Sinn, in diesem Fall war es besser, nur PNG-Thema zu machen.

Clips (clipPath)

Im Gegensatz zum ursprünglichen nanosvg-Projekt habe ich Clips unterstützt, aber der Algorithmus stammt nicht von mir, sondern wurde von den Pool-Anfragen übernommen. Ich weiß nicht, wie es funktioniert oder wie angemessen es ist, aber hier ist drin.

Das Clover-Motiv wird mit ihnen gezeichnet, und es ist fast richtig geworden. Es gibt einige Unterschiede im gerenderten Bild im Vergleich zur Vorschau, nur weiß ich nicht, wo der Fehler sitzt, bei der Clip-Unterstützung oder das Rendering selbst. Im Allgemeinen ist die Verwendung möglich, aber seien sie vorsichtig.

Texte und Fonts (Texts and SriLanka (?))

Das ursprüngliche nanosvg-Projekt hat diese Funktion nicht, aber Clover ohne Text? Udenkbar, also habe ich es getan. Normale Anzeige von Bildern mit eingebetteten Die Bilddatei-Schriftarten sind in Safari, aber nicht in Chrome und Explorer. Sie sind zeigen den Text in ihren eigenen Schriftarten, natürlich mit Designverzerrung. SVG eingebette Schriftarten sind in Illustrator verfügbar, aber nicht z. B. in BoxySVG, das nur TTF-Schriften einbetten. Ich kann mich nicht einmal mehr an den Vergleich mit anderen Vektor-Editoren erinnern, es ist für jeden auf die eine oder andere Weise ein Problem.

Das Vektorthema von Clover hat ein eigenes Regelwerk.

1. Es können mehrere Schriftarten verwendet werden, während bei Rasterthemen nur eine Schriftart verwendet werden kann. Eine von ihnen, die wichtigste sollte in der Datei theme.svg selbst enthalten sein. Sie können auch alle einschließen, nur dann wird die Datei in der Größe aufgebläht. Der Rest kann in den Theme-Ordner gelegt werden, neben die Datei theme.svg. Dabei gehen jedoch die Unterschiede in den Schriftarten in der Vorschau verloren. Also Clover sieht alle Schriftarten, Safari sieht nur eingebaute Schriftarten und Chrome sieht keine.

2. Derzeit können drei Textstile (Schriftart, Größe und Farbe) definiert werden: für das Optionsmenü, für Hilfe- und Info-Screenshots und für Meldungen am unteren Rand des Bildschirms den gleichen Stil in der Kopfzeile. Vielleicht erweitern Sie in Zukunft die Verwendung von Stilen. Dies geschieht mit einem besonderen Trick.

```
<g id="MenuRows" class="st0"><text class="st1">Menü</text></g>
<g id="HelpRows" class="st0"><text class="st2">Hilfe</text></g>
<g id="MessageRow" class="st0"><text class="st3">Boot macOS von Festplatte</text></g>
```

Hier sollen die Klassen st1, st2, st3 Textstile definieren und die Klasse st0 soll einfach unnötiger Text aus der Vorschau ausblenden (Deckkraft:0;).

Wie Sie sehen können, ist hier alles Standard, nur der Zweck ist anders. Hier ist der Text nur für um die Stile anzuzeigen. Und Clover selbst findet, was mit diesen Stilen zu schreiben ist, per id Gruppen, d. h. für das Menü "MenuRows" und für die F1-Hilfe "HelpRows".

3. aber im Allgemeinen können Sie Text in Symbolen als Dekorationselement verwenden. Sie können eine beliebige Schriftart verwenden, aber machen Sie es inline, und betten Sie vielleicht diese Schriftart mit der Option "nur verwendete Glyphen" ein - also nicht die gesamte Schriftart, die im Durchschnitt 500kb ausmacht, sondern Chinesisch und alle 30MB, aber nur die im Symbol verwendeten Buchstaben. Außerdem können Sie Transformationen auf den Text anwenden: Drehen, Neigen usw. Sie können den Schriftzug ohne Füllung verwenden, dann werden die Buchstaben "weiß" - ich weiß nicht mehr, wie es heißt, in Windows ist es so.

Der Stil ist nicht da, Mac schon, und das schon [seit den Pelzjahren\(?\)](#).

Im Gegensatz zum Standard gibt es keinen textPath - die Positionierung von Text entlang einer Linie, z. B. Entlang des Umfang des Kreises. Vielleicht.kommt das.eines Tages

Die Textgröße wird in Pixel eingestellt. Das ist .st8{font-size:16px;}, aber das ist in Designgröße Themen. Bei der eigentlichen Arbeit wird das Motiv auf die Monitorgröße skaliert, zusammen mit Briefgrößen. Bei einem Designthema von 1366x768 sieht der Text also normal aus mit 12px, auf einem Monitor mit einer Vertikalen von 2144 wären es fast 36px.

4. Ich benötige eine Schriftart für mein Design, die spezielle Unicode-Zeichen enthält. Zum Beispiel ☘. Ich habe eine solche Schriftart gefunden, sie benötigt 30Mb, wo soll ich sie ablegen? Ich beziehe von dort nur einzelne Schriftarten. Zeichen, und biete sie in mein Thema ein. Wenn der Illustrator eine unbekannte Schriftart sieht, ersetzt er sie durch seine eigene, und es tötet alles, was ich in meinem Thread getan habe

Die erste Regel ist, dass die verwendete Schriftart im System installiert sein muss, nicht nur im Thema. Aber, ich habe diese Schriften nicht als SVGs auf dem System installiert. Was ich also sagen will, ist... Ich rufe meine Inline-Schriftart ArialMT, Helvetica u. ä. und füge dort die Glyphen aus der gewünschten Schriftart, zeigt Illustrator sie falsch an, weil sie in einer anderen Schriftart ist, aber wenigstens bringt es mich nicht um. Quicklook zeigt es korrekt an, und ich kann die Buchstaben dort mit den Händen bewegen, wo sie sein sollten.

Ich habe dann auch die Beschriftung auf den Icons richtig. Ich habe das gerade mit einem Illustrator gemacht, seine Schriftart, und festgelegt, dass die Schriftart "nur Glyphen verwendet" exportiert wird. Dabei gehe ich das Risiko ein, dass um das Thema auf einen Buchstaben zu aktualisieren, den wir bisher noch nicht verwendet haben. Also, wir müssen Text mit allen englischen Buchstaben in das ursprüngliche Design des Themas implementieren, Zahlen und Interpunktion. Wie dies

```
<text class="st1">ABCDEFGHIJKLMNOPQRSTUVWXYZ...
```

und so weiter. Dann vergisst Illustrator beim Importieren des Themas nicht alle diese Symbole. Und, äh. platzieren Sie diesen Text in einer unsichtbaren Gruppe.

5. Nachdem Sie Illustrator verwendet haben und ihn gebeten haben, den SVG-Hintergrund einzubetten, nur verwendete Schriftarten oder überhaupt alle, dann sollten Sie prüfen, ob sie vorhanden ist sonst funktioniert es nicht.

Hier zum Beispiel:

Code:

```
<font horiz-adv-x="2048">
```

```
<!-- Helvetica is a registered trademark of Linotype AG -->
```

```
<!-- Copyright: Copyright 2018 Adobe System Incorporated. All rights reserved. -->
```

```
<font-face font-family="MyriadPro-Regular" units-per-em="2048" underline-position="-155" underline-thickness="101"/>
```

```
<missing-glyph horiz-adv-x="1298" />
```

```
<glyph unicode=" " horiz-adv-x="569"/>
```

```
<glyph unicode="E" horiz-adv-x="1366" d="M175,1469l1071,0l0,-180l-877,0l0,-446l811,0l0,-170l-811,0l0,-498l892,0l0,-175l-1086,0z"/>
```

```
<glyph unicode="H" d="M161,1469l201,0l0,-607l764,0l0,607l201,0l0,-1469l-201,0l0,687l-764,0l0,-687l-201,0z"/>
```

Erstens fehlt dem `bbox=""` das `font-face-Element` und das ist unbedingt erforderlich, suchen Sie , was dort stehen sollte, und dann sollte es handschriftlich eingetragen werden.

Zweitens: Blättern Sie durch die gesamte Liste, so dass jede Glyphe ein `horiz-adv-x=""` Attribut hat, definiert die Breite des Buchstabens, d. h. wo der nächste Buchstabe gezeichnet werden kann. Es gibt einen Standardwert, der aber nicht immer mit der tatsächlichen Breite des Buchstabens übereinstimmt, als Ergebnis eines solchen Buchstabens wäre, den Nachbarn-Buchstaben zu überschreiben.

In diesem Beispiel fehlt dem Buchstaben H dieses Attribut, und sein tatsächlicher Wert ist 1479, was nicht offensichtlich ist. Es ist möglich, von einem Buchstaben mit annähernd gleicher Breite auszugehen, und es ist möglich, durch Serienfen ein bisschen mehr oder ein bisschen weniger experimentieren.

Theme attributes

Clover braucht seine Ergänzungen über den Standard, und dies geschieht in seinem Namensraum. Dazu wird am Anfang der Datei Folgendes hinzugefügt:

```
xmlns:clover="https://sourceforge.net/projects/cloverefiboot"
```

Dies ist ein rechtmäßiger Vorgang. Danach können wir illegale Dinge hinzufügen, aber mit dem Präfix Clover: wie oben bei Dithering.

Ein weiteres Beispiel für spitz zulaufende Farbverläufe, dies ist eine Abweichung vom Standard, daher musste ich ein zusätzliches Attribut, das vom Standard erlaubt ist, aber Clover anzeigt, dass man über den Tellerrand zu schauen muß.

```
radialGradient clover:conic="1" id="knopkaUp" cx="14142.7324" cy="-40300.8711" r="37.5003"
gradientTransform="matrix(-0.706 -0.7082 0.7082 -0.706 38525.8438 - 18436.5312)" gradientUnits="userSpaceOnUse">
<stop <stop <stop <stop <stop
offset="0" style="stop-color:#FFF8D4"/> offset="0.1" style="stop-color:#EEE0B4"/> offset="0.5" style="stop-
color:#161616"/> offset="0.9" style="stop-color:#EEE0B4"/> offset="1" style="stop-color:#FFF8D4"/>
</radialGradient>
```

Radialer Gradient wird hier beschrieben, und so wird er auch von anderen Programmen verstanden werden. Clover interpretiert dies jedoch als spitz zulaufende Steigung. Und nun die allgemeinen Theme-Einstellungen

```
<clover:theme
BootCampStyle="0"
SelectionOnTop="0"
SelectionColor="0x80808080"
NonSelectedGrey="0" VerticalLayout="0"
BackgroundScale="crop"
BackgroundDark="1"
BackgroundSharp="0x80"

Badges="show"
BadgeOffsetX="0x0"
BadgeOffsetY="0xA"
BadgeScale="0x10"
LayoutBannerOffset="10"
LayoutButtonOffset="0" CharWidth="16"
AnimeFrames="39" FrameTime="2000"

Version="100500" Year="2018"
Author="Me"
Description="My cool vector theme for Clover"/>
```

Alle diese Attribute werden aus den Einstellungen des Bitmap-Designs kopiert und dienen demselben Zweck. Es werden jedoch nicht alle dieser Attribute benötigt und verwendet.

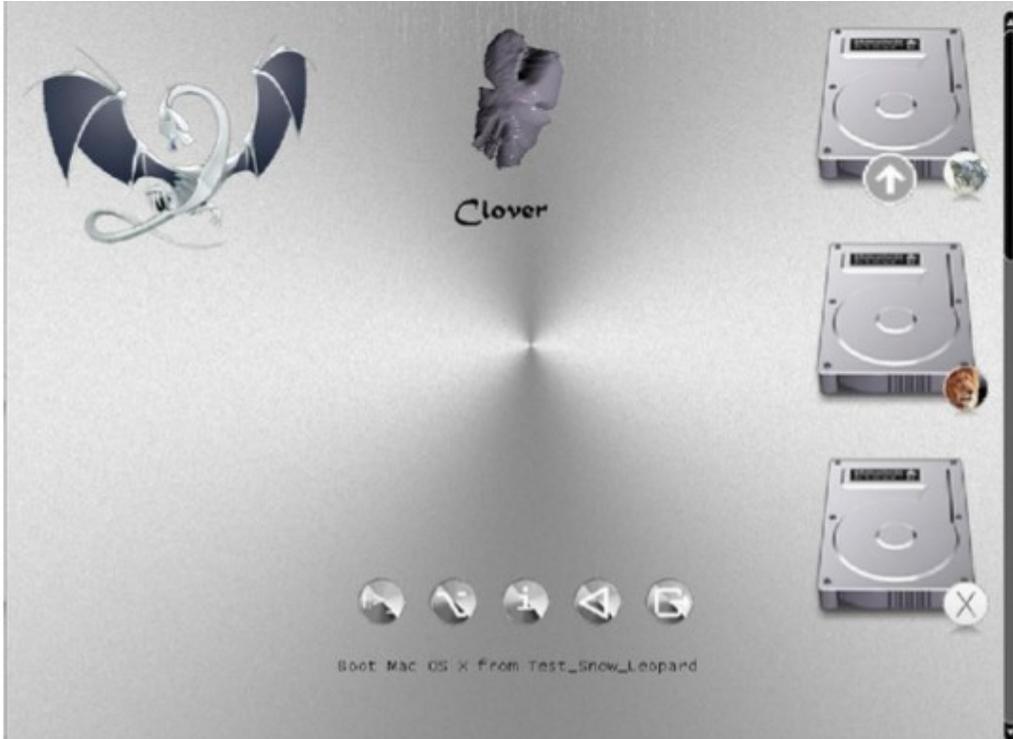
Fazit

Vektorthemen sind vielversprechender, und ich habe nicht den geringsten Wunsch mit Bitmap zu arbeiten. Es gibt einen Nachtmodus, schöne Schriftarten und Skalierung. Es ist möglich, dynamische Bilder wie Animationen zu erstellen, nur auf einer anderen Ebene. Für dies erfordert wieder die Implementierung von Abweichungen vom SVG-Standard. Aber das Erstellen eines Vektor-Themas ist eine viel zeitaufwendigere Aufgabe, und nur drei Leute haben es bisher geschafft: Clovy mit dem Thema Clovy, ich mit dem Cäsium-Thema, und blackosx mit dem BGM_SVG-Thema, obwohl er sich noch nicht für die endgültige Version entschieden hat. Wir haben die Animation genau an seinem Thema getestet, aber jetzt hat er es aus irgendeinem Grund zurück gezogen, wahrscheinlich um auf die Vektoranimation zu warten. Der vierte Designer übernahm die Aufgabe: pkdesign mit dem Thema "Purple Swirl". Wir freuen uns, eine neue Generation kennenzulernen!

Cesium theme, Tagesansicht



Nachtsansicht



Hardware-Konfiguration

Erstellen einer config.plist-Datei

Eigentlich übernimmt Clover die Konfiguration automatisch. Aber der Automat ist nie perfekt, daher hat der Benutzer die Möglichkeit, die verschiedenen Parameter über die Datei config.plist oder einfach im Menü Optionen zu ändern, wenn er in der grafischen Benutzeroberfläche arbeitet. Sie glauben, Sie können Einstellungen besser konfigurieren als ein Automat? Probieren Sie es aus! Probieren Sie es aus!

Die config.plist ist eine xml-Datei, aber an dieser Stelle ist es praktisch, sie als Textdatei zu zeigen. Diese Datei kann mit einem Texteditor bearbeitet werden oder durch ein spezialisiertes Programm wie PlistEditor oder Xcode, falls vorhanden. Jetzt verfügt die Clover.app, über eine Plist-Bearbeitungsfunktion, und ist im Clover-Paket enthalten ist. Sie ist so gut wie die oben genannten Programme für diesen Zweck. Nutzen Sie es aus! Zusammen mit Clover wird eine Variante dieser config.plist mit den auskommentierten Einstellungen mitgeliefert, damit Sie sehen können, welche Einstellungen es prinzipiell gibt, aber wenn sie so verwendet werden, wie sie sind, werden sie nicht berücksichtigt werden. Beispiele:

```
<key>#SuspendOverride</key>
<false/>
```

Das #-Gitter bedeutet, dass dieser Schlüssel überhaupt nicht in der Konfiguration enthalten ist. Clover verwendet die Standardeinstellung.

```
<key>GUI</key> <dict>
<key>#Custom</key>
<dict>
```

Und hier wird ein ganzer Abschnitt von Custom ausgeschlossen.

Die allgemeine Regel ist, wenn Sie nicht wissen, welcher Wert einem Parameter gegeben werden soll, schließen Sie diesen Parameter ganz aus der Datei aus, oder setzen Sie #. Lassen Sie den Parameter nicht ohne Wert! Und setzen Sie auf keinen Fall einen Wert ein, den Sie nicht verstehen!

Die folgende Variante zur Erstellung einer solchen Konfiguration für Ihren Computer wird angeboten:

- installieren Sie die standardmäßig mitgelieferte Beispieldatei, sie enthält nur sichere Parameter;
- Booten Sie in die grafische Clover-Shell und rufen Sie das Menü Optionen auf (es gibt ein Schaltfläche in der unteren Zeile oder einfach durch die Taste "O");
- Benutzen Sie die Tasten Auf/Ab/Enter/Escape, um durch das gesamte Menü zu gehen, und versuchen Sie, sich zurechtzufinden, was darin steht und warum;
- Was verständlich ist, reparieren wir, was nicht verständlich ist, lassen wir so, wie es ist.
- in das System booten. Wenn dies fehlschlägt, wiederholen Sie den Vorgang, aber ändern Sie den Parameter, bis zum vollen Erfolg.

Wenn Sie angemeldet sind, gehen Sie zum Terminal und geben Sie den Befehl:

```
cd ~/App/clover-genconfig >config.plist
```

Vorausgesetzt, Sie haben das Dienstprogramm genconfig zuvor in den Ordner ~/App gelegt.

Auf diese Weise erhalten Sie eine nahezu vollständige config.plist mit Ihren erfolgreichsten die Parameter, mit denen Sie booten konnten.

Warnung! Das Dienstprogramm clover-genconfig ist abhängig von der Clover-Revision!

Ab der Revision 5100 haben wir ein Programm Clover.app, das die gleiche Funktion hat. genconfig, und dieses Dienstprogramm ist nun ausgeschlossen.

Ein bisschen mehr Handarbeit für totalen Perfektionismus. Im Folgenden finden Sie eine Beschreibung von Konfigurationsparameter.

Alle Parameter sind in Gruppen zusammengefasst: ACPI, Boot, CPU, Geräte, DisableDrivers, GUI, Graphics, KernelAndKextPatches, RtVariables, SystemParameters, SMBIOS, BootGraphics.

Boot

```
<key>Timeout</key>  
<integer>5</integer>
```

Nachdem der Bootloader die grafische Oberfläche anzeigt, pausiert er für 5 Sekunden, bevor er das standardmäßig ausgewählte System startet. Wenn der Benutzer während dieser Zeit eine beliebige Taste drückt, wird der Zeitablauf unterbrochen. Optionen: wenn sie 0 sec eintragen, wird - GUI nicht aufgerufen, und das System wird sofort startet, wenn Sie jedoch vorher die Leertaste drücken, gelangen Sie in die GUI. -1 (minus eins) - der Bootloader geht in das Menü, er macht keinen Startversuch.

Es wird eine Pause für 25 Sekunden in der Konfiguration standardmäßig gemacht, damit der Benutzer die Animation bewundern kann.

Timing und Autostart erfolgt nur, wenn das richtige DefaultVolume eingestellt ist und nicht die in der Standardkonfiguration. Manchmal tickt der Timer wegen der starken Animation langsamer als 1Hz, nicht erschrecken, es hängt vom Motiv ab.

Die Zeitüberschreitung funktioniert nicht, wenn das Standardsystem nicht im NVRAM definiert ist. Weiter zu System, in die Systemablage der Startdiskette und starten Sie damit neu. B

Das nächste Mal wird ein Timeout funktionieren!

Die Variante mit Timeout=0 kann ersetzt werden durch die Variante

```
<key>Fast</key>  
<true/>
```

In diesem Fall ergibt sich eine zusätzliche Einsparung von Ladezeit durch das nicht Laden der Schnittstelle und ihre Elemente. D.h. ohne die Möglichkeit, die GUI zu betreten. Und keine Chance, etwas zu tun zur Korrektur im Falle eines Fehlers. Die Einsparung einer ganzen Sekunde wird Sie wahrscheinlich bereichern?! Das System beginnt sofort mit dem Booten von der im folgenden Parameter angegebenen Partition

```
<key>DefaultVolume</key>  
<string>MacHDD</string>
```

Partitionsname, wie Sie ihn benannt haben, wie er in Ihrem Bootloader-Protokoll erscheint. Allerdings, der Name kann auch nach einem Neustart über das Bedienfeld im NVRAM gesetzt werden "Die "Startup Disk". Der im NVRAM angegebene Name hat Priorität. Es gibt die Option "LastBootedVolume". Das heißt, wir booten von dem Volume, von dem wir das letzte Mal gebootet haben. Wenn der Parameter überhaupt nicht eingestellt ist, wird nur über das Bedienfeld gebootet.

Sie können auch den Standard-Bootloader festlegen

```
<key>DefaultLoader</key>  
<string>bootmgfw.efi</string>
```

Das heißt, wenn es mehrere Bootmöglichkeiten auf einer Partition gibt, wählen wir standardmäßig den zum Booten erforderlichen Eintrag. In diesem Beispiel gehen wir davon aus, UEFI zu laden Windows ist der Standard. Wenn nicht gesetzt, boot.efi. Nochmals: "Bootlaufwerk". ändert diese Einstellung für die Neustartzeit und stellt damit sicher, dass Windows automatisch über den Clover-Bootloader geladen wird. Leider gibt es einen solchen Dienst in Windows nicht, gehen Sie zurück zu Mac wollen, kann das nur manuell durchgeführt werden.

```
<key>Legacy</key>  
<string>PBR</string>
```

Legacy Boot, das zum Ausführen älterer Versionen von Windows und Linux benötigt wird, ist in hohem Maße abhängig von der Hardware, vom BIOS-Design, so dass mehrere Algorithmen entwickelt wurden, und die Wahl des Algorithmus wird mit dieser Einstellung getroffen.

Optionen:

LegacyBiosDefault - für die UEFI-BIOS, die das LegacyBios-Protokoll verwenden.

PBRtest, **PBR**, **PBRsata** sind Varianten des PBR-Boot-Algorithmus, je nachdem, wie viel Glück Sie haben. (Und im Allgemeinen wurde der bedingungslose legacy boot nie erreicht). Einfacher und schon besser Vergessen Sie die Altsysteme und setzen Sie UEFI-Versionen von Systemen ein. Der älteste von ihnen ist Windows 7-64, und ich persönlich sehe keinen Grund, bei WindowsXP zu bleiben. Hat jemand eine 32 bit Maschine nur noch am arbeiten? Na dann, viel Glück!

<key>LegacyBiosDefaultEntry</key>

<integer>0</integer>

Optional können Sie beim UEFI-Boot angeben, von welcher Festplatte gebootet werden soll (nicht nur von der ersten).

<key>Arguments</key>

<string>-v arch=i386</string>

Dies sind die Argumente, die an boot.efi übergeben werden, das wiederum einige von ihnen an den Kernel weiter gibt Systeme. Eine genaue Liste der Kernel-Argumente finden Sie in der Dokumentation von Apple. Gelistet bei Argumente, die von boot.efi selbst benötigt werden, finden Sie in der Anleitung com.apple.Boot.plist.

Die bekanntesten sind

Kernel=mach_kernel.amd

slide =0

darkwake=0

nvda_drv=1

Der Schlüssel slide=0 wird für den UEFI-Boot auf einem 10.8- oder 10.9-System benötigt. Beginnend mit der Revision 1887 Sie wird bei Bedarf automatisch hinzugefügt. Ab der Revision 4369 gibt es einen Treiber AptioMemoryFix, und damit funktioniert dieser Weg nicht mehr, es erfolgt eine automatische Berechnung. Ab Revision 3712 (aufgrund von Fehlern tatsächlich später) haben Sie die Möglichkeit, die gewünschten Argumente mit der Maus aus der Liste auszuwählen. Dieses Menü wird durch Space-Taste aufgerufen:

```
L "arch=i386",           //0
L "arch=x86_64",       //1
L "-v ",               //2
L "-s ",               //3
L "-x ",               //4
L "nv_disable=1",     //5
L "Dia=0",             //6
L "darkwake=0",       //7
L "-xcpm",            //8
L "-gux_no_idle",     //9
L "-gux_nosleep",    //10
L "-gux_noms",       //11
L "-gux_defer_usb2", //12
L "keepsyms=1",      //13
L "debug=0x100",     //14
L "kextlog=0xfffff", //15
L "-alcoff",         //16
L "-shikioff",       //17
L "nvda_drv=1"       //18
```

ZS. In Clover 4200 und höher wird die Anzahl der Argumente auf rein Kernel-bezogene reduziert, oder das Menü würde zu lang, und viele dieser Argumente sind nicht besonders notwendig.

<key>Debug</key>

<false/>

Dieser Key wurde früher Log genannt, was zu Verwirrung führte, warum und wie.

Das Setzen eines Wertes auf `<true/>` verlangsamt die Arbeit erheblich, bietet aber die Möglichkeit, des Neustarts, um herauszufinden, was das Problem war, denn nach jedem Schritt folgt ein Zugriff auf die Festplatte indem Sie die Datei `debug.log` auf die Festplatte geschrieben wird. Und wenn Sie mit einem USB-Stick beginnen, dann auf diesem. Aber mit einem USB-Stick wird es noch langsamer sein. Die reale Zahl ist 10 Minuten, nur um die GUI zu betreten. Das Gute daran, ist:

Wenn Sie nicht weiterkommen, können Sie Reset drücken und dann sich die Datei `/EFI/CLOVER/misc/debug.log` ansehen, das alle Protokolle für alle Booteinträge enthält, solange dieser Parameter gesetzt ist. Ab der Revision 3063 werden Sie nicht mehr vor dem schwarzen Bildschirm sitzen - Sie sehen den Bootvorgang auf dem Bildschirm! Aber wenn Sie nicht wie unnötige Schrift auf dem Bildschirm haben wollen, können Sie den Schlüssel benutzen:

`<key>NoEarlyProgress</key>`

`<true/>`

Diese unnötigen Beschriftungen werden verschwinden

Entfernen Sie die Beschriftungen vor dem Laden der Laderoberfläche, z. B. "Welcome to Clover".

`<key>CustomLogo</key>`

`<true/>` OR `<false/>` OR `<string>Apple/Alternate/Theme/None/Path</string>` OR
`<data>PNG/BMP/ICNS base64-Daten</data>`

true - Standardstil

false - Logo verbieten

Apple – Applegrau auf Grau

Alternate - alternativ Apple ist weiß auf schwarz

Theme - durch das Thema festgelegt

Keine - kein Logo, aber es gibt einen Hintergrund

Pfad - Pfad zur Logodatei

`<Daten>` - Abbildung ist als base64 kodiert und enthält PNG-Daten.

`<key>XMPDetection</key>`

`<string>-1</string>`

Der Parameter gibt an, ob XMP zur Boot-Zeit erkannt werden soll. Es hängt vom BIOS ab, und betrifft hauptsächlich die korrekte Erkennung des installierten Speichers. Darüber hinaus ist es möglich numerische Werte 1 oder 2 - welches XMP-Profil verwendet werden soll. Möglicherweise wird dieses Profil in Zukunft für andere Zwecke verwendet werden.

`<key>Secure</key>`

`<true/>`

"Secure Boot". Diese Erfindung von Microsoft hat eine heftige Reaktion in in der Computerwelt ausgelöst, da das bedeutete, daß auf neuen Computern nur noch Windows 8 laufen wird. Das führte in der Welt der Hackintosher dazu, daß einige "das Ende des Hackens!" ausriefen. Aber so traurig war es dann doch nicht. Natürlich war es das nicht. Die BIOS-Hersteller haben eine Möglichkeit vorgesehen, diese Funktion zu deaktivieren. Sie haben auch vorgesehen dafür Zertifikate laden. Bei mir zum Beispiel haben diese BIOS-Einstellungen keinen Einfluss auf Ladeerfolg.



Apiani beschloss, etwas mehr zu tun als das. Lassen Sie uns fortfahren und Clover unterschreiben, mit der Hilfe von irgendein "Signing Tool"-Programm, laden Sie das Zertifikat herunter und lassen Sie das BIOS in SecureBoot. Ich verstehe nichts davon, also bringe ich Ihnen nur, was bereits in Clover ist, kein Kommentar. Ich hoffe, dass ich in Zukunft Kommentare hinzufügen kann.

<key>Policy</key>

<string>Deny/Allow/Query/Insert/WhiteList/BlackList./User</string>

Deny - nur signierte Dateien hochladen.

Allow - Hochladen beliebiger

Query - fragen Sie den Host

Insert - fügt die Unterschrift in die Datenbank ein

WhiteList - nach Liste zulassen

BlackList - nach Liste ausschließen

User - prüfen Sie zuerst die Listen, und fragen Sie dann den Benutzer.

Die Syntax lautet

<key>WhiteList</key>

<array>.

<string>SOMEPATH.efi</string>

</array>.

<key>BlackList</key>

<array>

<string>USB(0x1)/HD(0x0,0x1038833...) \EFI\BOOT\BOOTX64.efi</string>

</array>

Möglicherweise benötigen Sie auch die Taste ignore hibernate, und zwar aus dem einfachen Grund, dass Das Bild ist gut, aber die Technik selbst funktioniert auf diesem Computer nicht.

<key>NeverHibernate</key>

<false>

Oder vielleicht sind wir wirklich begeistert von Hibernate, und wir wollen nicht fünf Sekunden warten müssen bis es wirksam ist. Dann schreiben wir

<key>SkipHibernateTimeout</key>

<true/>

Die zweite Hibernate Methode (des Winterschlafs):

<key>StrictHibernate</key>

<true/>

Dies funktioniert nur, wenn Sie ein echtes NVRAM haben, aber es ist kompatibel mit der FileVault2-Technologie, wo der alte Weg nicht funktioniert. **Hier ist ein Update**. Die Lilu+HibernateFixup-Kext erlauben es, die nvram.plist zu speichern, wenn man in den Ruhezustand geht, d.h. sie emulieren teilweise das echte NVRAM und helfen so, StrictHibernate mit mod 25 auf Maschinen mit emuliertem NVRAM zu verwenden.

<key>RtcHibernateAware</key>

<true/>

Ein Schlüssel zum sicheren Betrieb des RTC während des Ruhezustands. Der Autor ist vit9696, Fragen dazu bitte an ihn richten. Er behauptete, dass dies nur für 10.13.4-Systeme gilt, aber mein 10.13.6 behält den Schlüssel im NVRAM dennoch. Indes ist es notwendig, <true> aus anderen Gründen zu setzen, die nicht mit RTC zusammenhängen, und mit einem anderen Weckalgorithmus in 10.13.6. zu tun haben

<key>HibernationFixup</key>

<true/>

Autor: lvs1974, erklärt in seinem Thread, wie und wann es funktioniert. Etwas in der Art von. der oben beschriebenen Situation, in der kein echtes-NVRAM vorhanden ist.

<key>SignatureFixup</key>

<true/>

Beim Ruhezustand hinterlässt das System eine Signatur im Abbild, die anschließend durch boot.efi überprüft wird. Das ist der Schlüssel, mit dem wir das Problem beheben wollten. Wahrscheinlich nicht. Das Richtige wäre es, es dabei zu belassen die Vorgabe ist einfach Null zu setzen, und es funktioniert. Dieser Key ist meiner Meinung nach unnötig.

Außerdem versucht das System manchmal, in den Recover-Modus zu wechseln, aber es bootet vom falschen des Laufwerks, wenn wir mehrere Systeme haben, es ist ein Hackintosh. Um aus dem verwünschten Ring zu kommen, schreiben wir in die Konfiguration

<key>NeverDoRecovery</key>

<true/>

Mit FileVault2 ist es nun möglich, einen Hotkey zu verwenden, aber dies erfordert, daß bereits in Clover gemachte Eingaben zu storniert werden:

<key>DisableCloverHotkeys</key>

<false/>

BootGraphics

<key>DefaultBackgroundColor</key>

<string>0xF0F0F0</string>

Die Hintergrundfarbe beim Booten von Mac OS X, wenn ein Apfel mit einem Ladebalken erscheint. Wird im HEX-Format eingestellt.

<key>EFILoginHiDPI</key>

<integer>1</integer>

Aktiviert HiDPI für Apple mit einem Ladebalken, wenn Mac OS X geladen ist.

<key>UIS-Skala</key>

<integer>1</integer>

Skaliert die Größe des Apfels mit dem Ladebalken, wenn Mac OS X geladen ist.

SystemParameters

<key>CustomUUID</key>

<string>511CE200-1000-4000-9999-010203040506</string>

Die eindeutige Identifikationsnummer Ihres Computers. Wenn Sie diesen Key nicht einstellen werden einige der Hardware-Informationen generiert, aber wenn Sie die volle Kontrolle haben wollen über das, was vor sich geht, schreiben Sie Ihre eigenen 16 Ziffern.

Aber, um Gottes Willen, kopieren Sie nicht meine beispielhaften Figuren! Sie sind schon lange nicht mehr einzigartig, (Dummköpfe. Wie viele, haben sie wohl kopiert ?!)

<key>InjectSystemID</key>

<false/>

Die gleiche Nummer wird auf eine andere Weise injiziert und in den Eigenschaften des Systems in etwas anderes umgewandelt. Der Sinn dieses Vorgangs ist es, die vom Chameleon generierte UUID genau zu treffen. Dazu setzen wir <true />, und als CustomUUID verwenden wir den Wert, der beim Chameleon im registryIODeviceTree vorhanden ist: / efi / platform => system-id... Im Profiler sehen wir dann einen anderen Wert, aber den gleichen wie vorher mit dem Chameleon

Neue Benutzer wissen nicht mehr, was "wie vorher ..." ist. Sie sind schon bei Clover. Es ist fast immer ausreichend, <true /> in diesen Parameter zu schreiben und keinen Custom. Wenn aber eine nicht eindeutige Nummer erzeugt wird, dann wird es schwierig zu verstehen, woher die Beine des Problems kommen. Also setzen Sie ihn besser auf false.

<key>BacklightLevel</key>

<string>0x0101</string>

Diese Eigenschaft wird in das System eingespeist und das System ist sich ihrer Existenz bewusst. Allerdings, der Effekt ist nur bei sehr seltenen Konfigurationen spürbar. Was ist das? Bildschirmhelligkeit... wie. ergibt sich aus dem Namen. Diese Eigenschaft wird ebenfalls aus dem NVRAM gelesen, und zwar standardmäßig, wird der vom System eingestellte Wert verwendet. Der in der Konfiguration vorgegebene Wert, oder im Menü eingestellt ist, überschreibt den Standardwert.

Ab der Revision 1865 wurden zusätzliche Clover Keys eingeführt:

<key>InjectKexts</key>

<string>Detect</string>

Caches werden immer geladen, wenn kein FakeSMC im Cache vorhanden ist. Ansonsten wird davon ausgegangen, dass sich alle Caches im Cache befinden.

Wann immer möglich, startet das System mit einem Cache, und es geht darum, dass das System eventuell ob der Cache verwendet oder neu erstellt werden soll. Wenn der Wert Ja ist, dann wird Clover Injektion von Caches erzwingen, auch wenn sie sich im Cache befinden.

Nein, nicht! Schreiben Sie besser:

<true/>

Dies kann auch im Menü Details geschehen, das über die Leertaste auf dem System-Symbol aufgerufen wird. Diese Schlüssel werden vom Treiber FSInject.efi analysiert, sein Vorhandensein ist zwingend erforderlich, obwohl im Allgemeinen cksts aus den Clover-Ordnern und ohne ihn geladen werden. Dabei können jedoch Abhängigkeiten verloren gehen. In der Revision 5125 wird das Laden der Token von OpenCore übernommen, FSInject ist also nicht erforderlich, ebenso wie Detect.

<key>NoCaches</key>

<true/>

Der Parameter funktionierte auf Systemen vor 10.7. Es handelt sich um ein reines Cache-Laden, ohne Caches. Im Allgemeinen ein seltsamer Wunsch, diesen Parameter in die Konfiguration zu schreiben. Wahrscheinlich hat jemand anderes es nicht anders geladen bekommen. Aber heute ist der Parameter nutzlos, das System wird sowieso aus dem Cache geladen.

<key>NvidiaWeb</key>

<true/>

Setzen Sie das Flag für das Laden des Nvidia-Web-Treibers, den Standardwert, den Sie in der Clover-Oberfläche im Menü Details (aufgerufen durch die Leertaste auf dem System-Icon) ändern können. Die alte Methode von boot-arg funktionierte nur vor Captain. In Sierra und höher funktioniert diese Methode nicht, daher ist sie in einem separaten Konfigurationspunkt untergebracht.

Smbios

Diese Gruppe von Einstellungen wird benötigt, um Ihren PC einem Mac nachzubilden. Clover tut dies automatisch auf der Grundlage des erkannten CPU-Modells, der Grafikkarte und der Mobilitätsfunktion. Möglicherweise möchten Sie jedoch eine andere Wahl treffen. Nehmen Sie die MacTracker-Software und wählen Sie das Mac-Modell, das Ihnen am besten gefällt, und suchen Sie dann im Internet oder bei Ihren Freunden nach allen Modell- und Seriennummern. Hier gibt es nicht viel zu kommentieren. Diese Parameter sind nichts für Dummies. Wenn Sie sie kennen, ändern Sie sie. Sie können sie aber auch nicht berechnen.

<key>ProductName</key>

```
<string>MacBook1,1</string>
```

SMBIOS.table1->Produktname

Sie können nur den Produktnamen angeben und Clover berechnet aus seinen eigenen Tabellen alle andere Parameter, die diesem Modell entsprechen. Der Rest der Parameter kann weggelassen werden. Wenn Sie jedoch andere Parameter als den Standardwert wünschen, geben Sie diese ebenfalls ein.

Die neuen Einstellungen haben dann Vorrang. Allerdings ist die Liste der Namen, die Clover bekannt sind, begrenzt, 83 Modelle insgesamt, ich werde sie nicht alle aufzählen.

Für andere Optionen füllen Sie alle Felder manuell aus. Für den Fall, dass Sie unbekannt für die Clover-Tabelle sind, werden alle Felder zunächst mit Daten aus MacPro3.1, iMac13.1 ab Revision 3900, und dann mit Ihren Daten überschrieben, wenn Sie die einsetzen. Mit Ausnahme der BoardVersion, die das Modell automatisch replizieren wird. Wenn kein Modell angegeben wird, ersetzt Clover etwas aus dieser Liste, schauen Sie im Menü nach, wie wohl Sie sich bei dieser Wahl fühlen. Ändern Sie, wie Sie es für richtig halten.

Die Seriennummern sollten vorzugsweise in Ihrer eigenen Sprache geschrieben sein. Sie können ein Beispiel nehmen und einen Buchstaben in der Mitte ändern. Dies funktioniert normalerweise. Die ersten drei und letzten vier können nicht geändert werden.

<key>SmUUID</key>

```
<string>00000000-0000-1000-8000-010203040506</string>
```

SMBIOS.table1 → Uuid

Es scheint sinnvoll zu sein, die Mac-Adresse Ihrer Netzwerkkarte (die letzten sechs Zeichenpaare) hier einzutragen. Diese GUID wird auch verwendet, wenn CustomUID nicht angegeben ist. Besser noch: Geben Sie diesen Parameter nicht an.

<key>FirmwareFeatures</key>

```
<string>0xC0001403</string>
```

SMBIOS.table128 → FirmwareFeatures

Diese Zahlen gehen über den SMBIOS-Standard hinaus, es ist etwas, das spezifisch für Apple ist. Verschiedene echte Macs können in verschiedenen Nummern gefunden werden, es gibt nirgendwo eine Beschreibung, es sei denn, Sie können es im Quellcode finden

```
&& (featureFlags & 0x00000001) {contextprintf(context, kBLLogLevelVerbose, "Legacy mode supported\n");
```

Daher müssen wir auch hier eine ungerade Zahl haben. In den neuen Revisionen von Clover hat Sherlock eine automatische Berechnung des "besten" Wertes vorgenommen. Ich weiß nicht, was das für Auswirkungen hat. Vit9696 hat einige Arbeiten zur Bestimmung dieser Bits durchgeführt, siehe

<https://github.com/acidanthera/EfiPkg/blob/master/Include/IndustryStandard/AppleFeatures.h>

<key>PlatformFeature</key>

```
<integer>3</integer>
```

SMBIOS.table133->PlatformFeature

Dieser Parameter ist in den aktuellen Macs zu finden und wird vom Captain verwendet, aber was bewirkt er? Das ist noch nicht gefunden worden. Wenn der Wert nicht vorgeschrieben ist, wird die Tabelle 133 nicht erstellt.

<key>BoardSerialNumber</key>

`<string>C02032101R5DC771H</string>`

SMBIOS.table2->SerialNumber

Dieser Clover-Parameter liefert eine bestimmte Zahl. Sie müssen aber Ihre eigenen Zahlen verwenden. Diese Zahl ist notwendig, damit iCloud und iMessage funktionieren. Die Länge beträgt notwendigerweise 17 Buchstaben, große Lateinische Buchstaben und Zahlen. **Die in Clover festgelegte Zahl ist höchstwahrscheinlich seit langem verboten.**

<key>BoardType</key>

<integer>10</integer>

SMBIOS.table2->BoardType

Dieser Parameter wird für den MacPro eingeführt, der hier nicht 10 - Motherboard, sondern 11 - ProcessorBoard ist, offenbar aus historischen Gründen. Die Bedeutung ist nicht klar, aber Sie können sie im System Profiler sehen.

<key>BoardVersion</key>

<string>MacBook1,1</string>

SMBIOS.table2->BoardVersion

Ja, auch hier sollte ein Modell vorgeschrieben sein.

<key>Mobile</key>

<true/>

Eigentlich berechnet Clover immer korrekt, ob eine bestimmte Plattform mobil ist (d. h. batteriebetrieben, was Energieeinsparungen erfordert), oder nicht. Und der Parameter wird benötigt, wenn wir aus irgendeinem Grund das System austricksen wollen, damit es anzeigt, dass wir keine Batterie haben, oder andersherum.

<key>ChassisType</key>

<string>0x10</string>

SMBIOS.Tabelle3->Typ

Dieser Parameter dient dazu, festzustellen, ob wir eine mobile Plattform haben Hier ist die Tabelle nach SMBIOS-Standard

MiscChassisTypeOther	=	0x01,
MiscChassisTypeUnknown	=	0x02,
MiscChassisTypeDeskTop	=	0x03,
MiscChassisTypeLowProfileDesktop	=	0x04,
MiscChassisTypePizzaBox	=	0x05,
MiscChassisTypeMiniTower	=	0x06,
MiscChassisTypeTower	=	0x07,
MiscChassisTypePortable	=	0x08,
MiscChassisTypeLapTop	=	0x09,
MiscChassisTypeNotebook	=	0x0A,
MiscChassisTypeHandHeld	=	0x0B,
MiscChassisTypeDockingStation	=	0x0C,
MiscChassisTypeAllInOne	=	0x0D,
MiscChassisTypeSubNotebook	=	0x0E,
MiscChassisTypeSpaceSaving	=	0x0F,
MiscChassisTypeLunchBox	=	0x10,

Clover nimmt den Wert auf, wie er in den aktuellen Macs ausgestellt ist, entsprechend des von Ihnen gewählten Modell. Was es, außer der Mobilität, beeinflusst, weiß ich nicht.

<key>ChassisAssetTag</key>

<string>LatitudeD420</string>

SMBIOS.table3->AssetTag

Dieses Feld wird in echten Macs nie ausgefüllt, daher können wir es für unsere Bedürfnisse anpassen, z. B. um an das Projekt HWSensors3 anbinden.

<key>SmbiosVersion</key>

<string>0x0300</string>

Apple hat seinen eigenen SMBIOS-Standard, der nicht mit den anderen übereinstimmt, früher war er am ähnlichsten auf den 2.4-Standard, und Apple hat diese Zahl eingesetzt. Es war aber von der Bedeutung her eher mit 2.6 vergleichbar, Deshalb hat Clover die Zahl 2,6 eingesetzt. Auf echten Macs können Sie jetzt die Nummer 3.0 haben, aber das ist eine Lüge, der Standard ist immer noch 2.x. Trotzdem können wir unsere verwenden. Die Nummer scheint keine Auswirkungen zu haben.

<key>BiosVersion</key>

<string>IM131.88Z.F000.B00.1907241303</string>

Dieser Satz von Zahlen und Buchstaben bestimmt, ob Ihr Hackintosh nach einem BIOS-Update fragt, wenn sie einen realen Mac haben. Sie brauchen es aber nicht, deshalb wollen Sie die letzten Werte hier einfügen, wenn Sie die wissen. Und standardmäßig gibt Clover seine Zahlen an, die mit jeder Revision aktualisiert werden. Sie dürfen keine eigenen Zahlen eingeben, sondern müssen lediglich Clover aktualisieren.

<key>EfiVersion</key>

<string>288.0.0.0.0</string>

Ähnlich, aber ich bin mir nicht sicher, welchen Einfluss diese Zahl auf die Notwendigkeit eines Updates hat. Sie wird als Boot-ROM-Version angezeigt. Bei verschiedenen Modellen wird entweder die BiosVersion oder die EfiVersion verwendet.

<key>BiosVendor</key>.

<string>Apple Inc.</string>.

<key> BoardManufacturer </key>

<string>Apple Inc.</string>

<key>FirmwareFeaturesMask</key>

<string>0xFFFFFFFFFFFF</string>

<key> LocationInChassis </key>

<string>MLB</string>

<key> MemoryRank </key>

<integer> 2 </integer>

<key>Version</key>

<string>1.0</string>

<key> Manufacturer </key>

<string>Apple Inc.</string>

Die oben aufgeführten Parameter (BiosVendor, BoardManufacturer, FirmwareFeatureMask, LocationInChassis, MemoryRank, Version, Manufacturer) sind verantwortlich für. Es ist besser, sie nicht manuell einzustellen, wenn Sie nicht genau wissen, was diese Änderungen zur Folge haben werden. Noch besser ist es, diese Parameter zu entfernen oder auszukommentieren, dann wird Clover sie automatisch generieren, wenn das System hochfährt, vertrauen Sie darauf.

<key>NoRomInfo</key>

<false/>

Clover ist in der Lage, SMBIOS.table11 mit eigenen Firmware-Nummern zu erzeugen. Sehr praktisch im Fenster „Über Mac“ sieht man diese Werte (AppleROM-Informationen)

Аппаратные средства:

Название модели:	iMac
Идентификатор модели:	iMac17,1
Имя процессора:	Intel Core i5
Скорость процессора:	3,3 GHz
Количество процессоров:	1
Общее количество ядер:	4
Кэш 2-го уровня (в каждом ядре):	256 КБ
Кэш 3-го уровня:	6 МБ
Память:	8 ГБ
Версия Boot ROM:	178.0.0.0.0
Информация Apple ROM:	Apple ROM Version. Board-ID : Mac-27ADBB7B4CEE8E61 ⌘ Powered by Clover revision: 5119 (master, commit 0f5da727e)
Версия SMC (система):	2.33f12
Серийный номер (система):	C02QF3E1007L
UUID аппаратного обеспечения:	1C021B03-0D04-6605-5B06-710700080009

Мerkwürdigerweise haben die echten Macs diese Information nicht, obwohl es eine solche Tabelle gibt. Wenn jemand möchte, das es wie ein echter Mac aussieht, setzen Sie diesen NoRomInfo-Schlüssel auf <true/>.

<key>Trust</key>

<true/>

Der Parameter wird verwendet, um einen Streit zwischen SMBIOS und dem SPD zu schlichten, dessen Speicherparameter genauer erkennen, zusätzlich zu der Tatsache, dass es auch interne Checks gibt. Es steht standardmäßig auf true, d. h. SMBIOS (DMI)-Werte sind genauer.

Wenn Sie jedoch nicht die "richtige" Speicherzuordnung mit entweder true oder false im System haben, können Sie alles manuell verschreiben (ab Revision 1896)

<key>Memory</key>

<dict>

<key>Channels</key>

<integer>1/2/3</integer>

<key>SlotCount</key>

<integer>24</integer>

<key>Modules</key>

<array>

<dict>

<key>Slot</key>

<integer>0</integer>

<key>Size</key>

<integer>2048</integer>

<key>Frequency</key>

<integer>1600</integer>

<key>Vendor</key>

<string>Some Company</string>

<key>Part</key>

<string>123456ABCDEF</string>

<key>Serial</key>

<string>ABCDEF123456</string>

<key>Type</key>

<string>DDR/DDR2/DDR3</string>

</dict>

Einige Klarstellungen:

Kanäle - die Anzahl der Speicherkanäle. Auf sehr alten Computern gab es nur einen Kanal. Auf modern 2. Es gibt einzelne Konfigurationen (z. B. Clarkdale), bei denen gibt es drei Kanäle, d.h. dreikanaliger Speicher. SlotCount - Gesamtzahl der Steckplätze, in die Speichersteckplätze eingefügt werden können. Angezeigt im Fenster „Über“. Zeichnen Sie nun ein Array von Modulen, das nur belegte Steckplätze beschreibt. Leere gerade nicht erwähnen. Schreiben Sie im Key Slot seine Nummer von 0 beginnend.



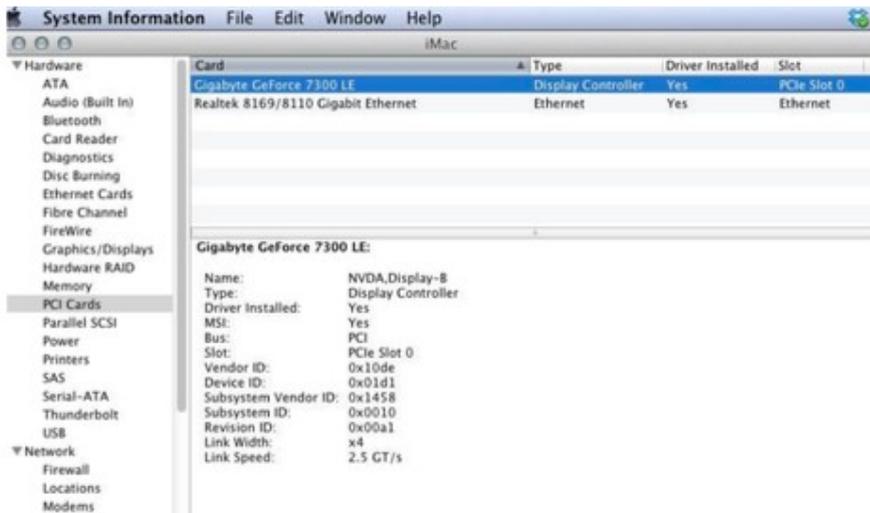
Schreibgröße in Megabyte und Geschwindigkeit in Megahertz. Lassen Sie keine Felder leer.
Khaki Clover. Version 5.1, Revision 5129 75

In der Seriennummer (Serial) und Inventarnummer (Part) sind nur Großbuchstaben erlaubt Buchstaben, Zahlen, Minuszeichen und Punkt. Lassen Sie mich an dieser Stelle die Frage nach der Korrektheit der Speicherzuordnung im System abschließen. (und fand sich noch einen Pudel, der behauptete, dass es nicht so angezeigt wurde, wie er es vorschrieb! Er hatte es wirklich nicht richtig geschrieben)

`<key>Slots</key>`

```
<array>
  <dict>
    <key>Device</key>
    <string>Nvidia</string>
    <key>ID</key>
    <integer>2</integer>
    <key>Type</key>
    <integer>16</integer>
    <key>Name</key>
    <string>PCIe Slot 0</string>
  </dict>
```

Dadurch werden die PCI-Geräte in den System Profiler geschrieben. So sieht es aus:



Um diese Eigenschaften einzutragen, schreiben Sie in die config:

```

<key>SMBIOS</key>
<dict>
  <key>Slots</key>
  <array>
    <dict>
      <key>Device</key>
      <string>Nvidia</string>
      <key>ID</key>
      <integer>2</integer>
      <key>Type</key>
      <integer>16</integer>
      <key>Name</key>
      <string>PCIe Slot 0</string>
    </dict>
  </array>
  <dict>
    <key>Device</key>
    <integer>3</integer>
    <key>Typ</key>
    <integer>1</integer>
    <key>Name</key>
    <string>Ethernet</string>
  </dict>
</dict>

```

Und Clover wird solche Tabellen generieren. Damit die entsprechenden _SUN-Eigenschaften in der DSDT erscheinen. Wenn sie nicht bereits vorhanden sind, müssen Sie die Patch-Maske speziell für diese Geräte einstellen. Für dieses Beispiel ist es

```
<key>ACPI</key>
</dict>
  <key>DSDT</key>
  </dict>
    <key>Fixes</key>
    </dict>
      <key>FixDisplay_0100</key>
      <true/>
      <key>FixLAN_2000</key>
      <true/>
      <key>NewWay_80000000</key>
      <true/>
    </dict>
```

Wenn Sie diese Eigenschaften manuell schreiben, müssen sie mit der ID

```
DEVICE (GFX0)
{
  Name (_ADR, Null) // _ADR: Adresse Name (_SUN, 0x02).....
DEVICE (GIGE)
{
  Name (_ADR, Null) // _ADR: Adresse Name (_SUN, 0x03)
```

Vermeiden Sie ID = 0x00 und 0x01 wegen der Optimierung in Zero und One. Clover kann möglicherweise nicht mit mit so einem Patch umgehen.

Im Moment ist dieser Trick nur mit ATI-, NVidia-, LAN-, WIFI-Geräten und Firewire möglich

Dies sind vordefinierte Namen, Clover wird ein Gerät finden, das diesem Namen entspricht. Wenn Sie eine genauere Übereinstimmung wünschen, sehen Sie zuerst in Ihrem OEM-SBMIO nach, das Sie über den DarwinDumper-Bericht erhalten können, um zu sehen, welche No9-Tabellen Sie haben, welche SUNs auf welche Geräte abgebildet sind. Passen Sie dann Ihr DSDT an diese Zuordnungen an und fügen Sie Ihre eigenen hinzu.

Beispiel:

```
Handle 0x0905, DMI-Typ 9, 17 Bytes
0000: 09 11 05 09 01 a6 08 03 03 02 00 04 02 00 00 00
0010: fe
```

```
System Slot Information
Designation: Ethernet
Type: x1 PCI Express x1
Current Usage: Available Length:
Short
ID: 2
Characteristics:
  3.3 V is provided
  Hot-plug devices are supported
Bus Address: 0000:00:1f.6
```

Das heißt, ich habe einen Ethernet-Controller an der Adresse 0000:00:1f.6 und er soll _SUN=2 (grün hervorgehoben).

Dies ist das Gerät in DSDT, aber es hat nicht die Eigenschaft _SUN!:

```
Gerät (GLAN)
{
  Name (_ADR, 0x001F0006) // _ADR: Adress
```

Man muss es richtig machen! Vielleicht lernt Clover in Zukunft, dies automatisch zu tun.

Slot->Typ ist der Steckplatztyp aus der Liste PCI, PCIe x1, PCIe x2,... PCIe x16, die der Einfachheit halber mit den Ziffern 0, 1, 2, ...16 verschlüsselt sind.

In diesem Beispiel wird PCIe x1, als Slot->Type=1 verschlüsselt. Aber da Geräte-Tabelle #9 bereits vorhanden ist, muss es nicht in die Konfiguration geschrieben werden, es genügt, sie in DSDT zu schreiben

```
Device (GLAN)
{
    Name (_ADR, 0x001F0006) // _ADR: Adresse
    Name (_SUN, 0x02)
```

Khaki Clover. Version 5.1, Revision 5129
Moskau, 2021

77

Cpu

Diese Gruppe von Parametern hilft bei der Definition der CPU, wenn die internen Algorithmen scheitern.

```
<key>FrequencyMHz</key>
<string>3200</string>
```

Die Basisfrequenz des Prozessors in MHz. Normalerweise erhält Clover diesen Wert durch Berechnungen auf der Basis des ACPI-Timers, aber wenn er sich als falsch erweist, können Sie es durch diesen Key ändern. Diese Taste wirkt sich nur auf die Nummer im System-Profiler aus. Kosmetisch!
Bei Hasswells beträgt die Nennleistung z. B. 1800 MHz und die Anfangsgeschwindigkeit 2400 MHz. Wir werden mit 2400 MHz arbeiten, und für den Profiler, sagen wir aber 1.800 MHz.

```
<key>BusSpeedkHz</key>
<string>133330</string>
```

Dieser Parameter, die Basis-Busfrequenz, ist für den Systembetrieb entscheidend und wird vom Bootloader an den Kernel übergeben. **Wenn die Frequenz falsch ist, startet der Kernel überhaupt nicht, wenn die Frequenz leicht daneben liegt, kann es zu Taktproblemen und sehr seltsamem Systemverhalten kommen.** Der Wert im DMI wird in MHz gespeichert, und das ist ungenau, richtiger berechnet wird er aus der CPU-Frequenz, also können Sie Ihren Wert genauer abholen, und ihn in diesem Schlüssel in Kilohertz angeben. Zum Beispiel sagt mein DMI 100MHz, und es wurde besser für den Takt, als ich 99790kHz vorgeschrieben habe.

Ein Punkt. Einige Hersteller haben eine andere Vorstellung davon, was BusSpeed und was FSBSpeed ist, und schreiben ins BIOS einen viermal höheren Wert. Um das richtig zu verstehen, können Sie den Bereich verwenden: er muss zwischen 100 und 400MHz liegen, oder Sie können die Formel CPU-Frequenz=Busfrequenz*BusBSpeed-Multiplikator verwenden.

Es ist klar, dass wenn ASUS die Busfrequenz 1600MHz schreibt, aber der CPU-Multiplikator 8 ist, dann geht die Formel nicht auf, weil es keine 12,8GHz-Prozessoren gibt. In Wirklichkeit müssen Sie es durch 4 teilen. Ab der Revision 1060 gibt es eine Frequenz-Autodetect, die auf dem ADCP-Timer basiert und diese Werte besser berechnet als im DMI vorgeschrieben.

Khaki Clover. Version 5.1, Revision 5129
Moskau, 2021

78

<key>UseARTFrequency</key>

<false/>

SkyLake-Prozessoren haben einen neuen Basisfrequenz-Parameter, der mit kleineren Schritten als die Busfrequenz, die so genannte ARTFrequency, arbeitet. ihr Wert ist normalerweise gleich 24 MHz. Clover kann es berechnen und an den Kern weitergeben, und Captain wird es verstehen und verwenden. In der Praxis führt die berechnete Frequenz jedoch zu einem ungenauen Betrieb, daher kann man es einfach deaktivieren, in diesem Fall wird der Systemkernel sein eigenes Ding machen. In den neuen Versionen von Clover gibt es eine Aufrundung dieser Zahl, wie es vit9696 macht, da es seiner Meinung nach nur drei Werte geben kann, und sie werden aufgerundet auf 1 MHz Schritten.

<key>QPI</key>

<string>4800</string>

Im System-Profiler wird dieser Wert als Prozessor-Bus-Geschwindigkeit oder einfach als Bus-Geschwindigkeit bezeichnet. Chameleon hat einen Algorithmus zur Berechnung für Prozessoren der Nehalem-Familie (und der ist falsch!). Clover hat einen korrigierten Algorithmus gemäß den Datenblättern von Intel erstellt. Für die Quellen des AppleSmbios kext kommen zwei Möglichkeiten in Frage: entweder ist der Wert bereits vorgegeben im SMBIOS wie vom Hersteller vorgeschrieben, oder berechnen Sie einfach den $\text{BusSpeed} * 4$. Danach wird dieser Wert in die config gesetzt - schreiben Sie, was Sie wollen (in MHz). Bei der Arbeit es hat keinerlei Wirkung - reine Kosmetik. Nach den neuesten Informationen macht QPI nur für Sinn Nehalems, alle anderen hier müssen den $\text{BusSpeed} * 4$ haben. Oder ganz allgemein nichts. Wird das Schreiben von 0 erzwungen, wird die DMI-Tabelle 132 überhaupt nicht generiert. Jemand behauptet, dass man das auf modernen Macs tun sollte. (Ich hab's!)

<key>Type</key>

<string>0x0201</string>

Diese Option wurde von Apple erfunden und wird im Feld "Über diesen Mac" verwendet, durch die Übersetzung einer solchen internen Konstante in eine Prozessorbezeichnung. Ansonsten wird angezeigt "Unbekannte CPU". Warum konnte die CPUID nicht aufgerufen werden? (denn da gab es noch PowerPC). Oder sollten wir im SMBIOS in Tabelle 4 nachsehen? Nein, Apple hat seine eigene Weltsicht, und wir müssen einstellen, welcher Prozessor verschlüsselt wird. Grundsätzlich weiß Clover alle Chiffren, aber da Fortschritte gemacht wurden, ist die Fähigkeit eingeschränkt. Manuell können sie diesen Parameter ändern. Die Korrektheit der Einstellung dieses Parameters wird im Fenster "Über diesen Mac." sichtbar. Auch hier gilt: Reine Kosmetik. Wir haben die Informationen von vit9696

<https://github.com/acidanthera/EfiPkg/blob/master/Include/IndustryStandard/AppleSmbios.h>

Die Parametergruppe, die den C-Zustand betrifft, ist nun wieder in dem ACPI-Abschnitt, ACPI->SSDT.

Die folgenden Keys sind hier definiert. **Jetzt ungültig!**

<Taste>C2</key>.

<true/>.

Für moderne Computer setzen Sie false.

<key>C4</key>.

<true/>.

Je nach Spezifikation entweder C3 oder C4. Wir wählen C4. Für Evie setzen Sie ihn auf false.

<key>C6</key>

<true/>.

C6 ist nur auf mobilen Computern bekannt, Sie können es jedoch auf Desktop zu aktivieren. Auf Evie und Haswell setzen wir true.

Ich möchte anmerken, dass sich die Leute bei diesen C-states oft über schlechten Sound/Grafik/Schlaf beschweren. Seien Sie vorsichtig, oder eliminieren Sie sie ganz.

<key>Latency</key>

<integer>250</integer>

Dies ist die Verzögerung zum Einschalten des C3-Status. Der kritische Wert ist 0x3E8=1000. Bei weniger als diesem Wert ist Speedstep aktiviert, bei mehr als diesem Wert ist Speedstep nicht aktiviert. Immer 0x03E9 bei den Antivarianten, d.h. Speedstep funktioniert nicht. Bei Hacks müssen wir wählen, was wir wollen, den echten Macs ähnlich sein, oder das Powermanagement aktivieren. Ein sinnvoller Wert im zweiten Fall ist 0x00FA, wie er auf einigen Laptops zu finden ist.

MacPro5.1 = 17

MacPro6.1 = 67

iMac13.2 = 250

<key>SavingMode</key>

<integer>7</integer>

Ein weiterer interessanter Parameter für das speedstep management . Es betrifft das MSR-Register 0x1B0 und bestimmt das Verhalten des Prozessors:

0 - maximale Leistung

15 - maximale Energieeinsparung.

Ich habe mit dem iMac12-Modell Zwischenwerte mit den letzten beiden Tasten erhalten. Ich habe jedoch keinen endgültigen Beweis dafür, was was beeinflusst.

<key>QEMU</key>

<true/>

Beim Testen von Clover in der virtuellen Maschine QEMU habe ich festgestellt, dass QEMU den Intel-Prozessor ungenau emuliert. Als Zwischenmaßnahme wurde dieser Schlüssel gemacht, er behebt nicht alles. Es ist noch kein Wunder geschehen. Aber im singlemodus kann ich das System booten.

<key>TurboDisable</key>

<true/>

Nützlich für Laptops, damit diese sich nicht überhitzen.

<key>HWPEnable</key>

<true/>

Beginnend mit der Revision 3879 wurde die Intel Speed Shift Technologie für Skylake Prozessoren implementiert. Autor goodwin_c. Wenn der Key auf wahr steht, wird 1 in das MSR-Register 0x770 geschrieben. Ein Problem, das noch nicht geklärt ist, was man dagegen tun kann. Wenn Sie den Computer in den Ruhezustand versetzen und ihn dann wieder aufwecken, ist der MSR-Wert 0x770 und wird auf Null zurückgesetzt. Es gibt keine Möglichkeit, dass Clover es zurücksetzen kann. Bisher gibt es nur die Möglichkeit, immer immer 0 einzusetzen, d.h. in diesem Element <false/>, und dann wird bereits irgendwo im System auf andere Weise versucht, in diesem Register 1 zu setzen. Es wird davon ausgegangen, dass das System selbst dieses Bit setzt, wenn es weitere Bedingungen folgen, wie z. B. das richtige Modell im SMBIOS.

Richtiger ist es, dies mit einer kext zu arbeiten, die auch nach dem Aufwachen arbeitet

<https://github.com/headkaze/HWPEnable>

<key>HWPValue</key>

<string>0x30002a01</string>

Dieser Wert erwies sich als der am besten geeignete. Dieser Wert wird in das MSR-Register 0x774 geschrieben, aber nur wenn MSR 0x770 auf 1 gesetzt ist. Ansonsten ist dieses Register nicht verfügbar.

<key>TDP</key>

<integer>95</integer>

Dies ist die thermische Auslegungsleistung, die in p-Zuständen bei der Generierung von Steuertabellen die die Leistung des Prozessors berücksichtigt.

Graphics

Diese Gruppe von Parametern wird für die Injektion von Grafikkarteneigenschaften verwendet, wie es z. B. der Natit.kext tut. Es gibt eine Menge Parameter, die tatsächlich injiziert werden, aber sie sind meist Konstanten, einige werden berechnet, einige sind in der internen Tabelle definiert, und nur sehr wenige Parameter werden über die Konfiguration eingegeben.

<key>GraphicsInjector</key>

<true/>

Aktivieren Sie diese Injektionsfunktion tatsächlich. Übrigens ist sie standardmäßig aktiviert, weil die Injektion muss mit einer sauberen Konfiguration arbeiten - eine Voraussetzung für den Start des Systems. Deaktivieren der Injektion lohnt sich, wenn Sie den besten Weg kennen. Für einige moderne Karten wie Nvidia 6xxx oder Radeon 6xxx ist die Standardinjektion deaktiviert, da die nativen Werks-Einstellungen funktionieren. Unvollständig, aber auf dem Desktop können Sie kommen.

In der Revision 1921+ ist dieser Parameter veraltet, wird aber unterstützt, jetzt werden Grafikkarten separat, nach Hersteller, injiziert, da es auf modernen Computern fast immer eingebauten Intel-Grafik gibt es, und es ist manchmal nicht nötig, seine Injektion zu aktivieren.

<key>Inject</key>

<dict>

<key>Intel</key>

<false/>

<key>ATI</key>

<true/>

<key>NVidia</key> <false/>

</dict>

Injektion von zwei Dutzend Parametern, die nicht nur vom Kartenmodell, sondern auch aus den internen Eigenschaften der Karte berechnet werden, z. B. nach Analyse des Video-Bios.

Für Nvidia berechnen Sie dessen NVCAP, für Intel nehmen Sie zig Parameter auf (das Verdienst vonSherlocks), für ATI berechnen Sie connector-abhängige Parameter. Diese ganze Auflistung würde weitere zweihundert Seiten in Anspruch nehmen. Zumal man das bei modernen Karten nicht mehr braucht, Apple hat dafür gesorgt, dass sie out of the box funktionieren.

Diese Injektionen wurden vor dem Aufkommen von WEG verwendet. Da die gesamte Grafikanpassung durch das WhatEvergreen-Kontrollkästchen erfolgt, wird empfohlen, alle diese Injektionen bei Verwendung dieses Kontrollkästchens zu deaktivieren.

<key>VRAM</key>

<integer>1024</integer>

Die Größe des Videospeichers in MB. Eigentlich wird es automatisch erkannt, aber wenn Sie den richtigen Wert angeben - wird es niemand verletzt. Realistisch betrachtet, kann ich mich jedoch an keinen einzigen Fall erinnern, dass dieser Parameter für irgendjemanden von Nutzen ist. Wenn Sie jedoch 7Mb sehen, versuchen Sie nicht, die Einstellung. Dieser Parameter ist unbrauchbar. Sie müssen Ihre Grafikkarte starten. Für mobile Radeon gibt es z.B. einen Trick, LoadVBios=true zu verwenden und der Speicher wird korrekt erkannt.

<key>LoadVBios</key>

<true/>

Laden des Video-Bios aus einer Datei, die sich im Ordner EFI/CLOVER/OEM/xxx/ROM befinden muss oder EFI/CLOVER/ROM und sie haben den Dateinamen vendor_device.rom, zum Beispiel 1002_68d8.rom. Dies macht manchmal Sinn, wenn Sie ein gepatchtes Video-Rom verwenden. Auf Wunsch von Ermak ab 3222 Revision, kann ein längerer Dateiname verwendet werden, der den sub-vendor und sub-revision

10de_0f00_1458_3544.rom. Er braucht es zum Testen an einem der Computer, der unterschiedliche Grafikkarten hat.

Es gibt auch Probleme, dass eine Grafikkarte ihr Videochip dem System nicht anzeigt, obwohl das System es erfordert, z. B. im Fall von mobile radeon. In diesem Fall ist es möglich, diesen Parameter auf Yes zu setzen, aber schreiben Sie keine Datei in den Clover-Ordner. Clover übernimmt das VideoBIOS von dem Legacy-Speicher an Adresse 0xc0000, seltsamerweise ist er fast immer dort, und jetzt injiziert es Clover in das System, und der mobile radeon schaltet sich ein!

Khaki Clover. Version 5.1, Revision 5129
Moskau, 2021

81

Eine weitere Klarstellung. Es stellt sich heraus, dass das BIOS, das in das Karten-ROM geflasht wurde, nicht mit dem übereinstimmt, das an der Adresse 0xc0000 gebildet wird - der ROM-Schatten. Also, wir brauchen ihn, den Schatten, nicht das BIOS, das der Programmierer geflasht hat.

Um es kurz zu machen. Für mobile radeon setzen wir Yes, obwohl es keine Datei gibt, für andere Karten No. Andere Varianten sind historisch nicht festgelegt.

Eine neue Zeit ist angebrochen. Für Computer mit reinem UEFI-BIOS gibt es keine VideoBIOS auf der Legacy-Adresse. Legen wir es zu den Akten und warten wir auf neue Lösungen. Oder. Funktioniert es so wie es ist?

```
<key>Connectors</key>  
<array></array>
```

Der Parameter ist für die Zukunft reserviert, die noch nicht gekommen ist. Enthält noch keine Funktionalität.

```
<key>DualLink</key>  
<integer>0</integer>
```

Standardmäßig wird er mit 1 injiziert, aber bei einigen älteren Konfigurationen führt der Parameter=1 zu einer Vervierfachung der Anzeige (?). Es ist hilfreich, ihn auf 0 zu setzen, wie im obigen Beispiel.

```
<key>BootDisplay</key>  
<integer>1</integer>
```

Gibt an, welches der Displays das Hauptdisplay ist. Es ist diejenige, die beim Start leuchtet und nach dem Schlaf aufwacht. Normalerweise ist es die Nummer 0, aber manchmal sind die Ausgänge in der falschen Reihenfolge nummeriert, siehe ioreg wie Sie es haben. Revision 3399

```
<key>PatchVBios</key>  
<true/>
```

Clover nimmt eine Korrektur des VideoBIOS-Schattens an Adresse 0xC0000 vor, so dass er den für den angeschlossenen Monitor maximalen Videomodus unterstützt. Zum Beispiel hat die EDID des Monitors einen 1920x1080-Modus, aber die VideoBio nicht. Clover wird ihn als ersten Mod verschreiben und einsetzen. Wenn der Monitor selbst keine EDID generiert, kann sie wie unten dargestellt injiziert werden.

Es gab Fälle, in denen die Aktivierung dieses Patches zu einem schwarzen Bildschirm, beim Versuch zu booten, führte. Für den ersten Versuch schalten Sie diese Einstellung aus.

Entweder verwenden Sie den Wert aus der Datei config.plist für den Patch oder der Wert wird aus der Datei config.plist für den Patch verwendet:

```
<key>GUI</key>  
<dict>  
  <key>ScreenResolution</key>  
  <string>1440x900</string>
```

Wenn die Automatisierung einen Fehler gemacht hat, können Sie die VideoBios manuell patchen, indem Sie den Standard des Algorithmus "Suchen/Ersetzen" wählen.

Key	Type	Value
AppleKTC	String	Yes
▼ KextsToPatch	Dictionary	(3 items)
▼ 0	Dictionary	(3 items)
Name	String	VoodooHDA
Find	Data	<48656164 70686f6e 657300>
Replace	Data	<54656c65 70686f6e 657300>
▶ 1	Dictionary	(3 items)
▶ 2	Dictionary	(3 items)
▶ SystemParameters	Dictionary	(7 items)
▶ SMBIOS	Dictionary	(18 items)
▶ CPU	Dictionary	(5 items)
▼ Graphics	Dictionary	(10 items)
GraphicsInjector	String	Yes
InjectEDID	String	Yes
CustomEDID	Data	<00ffffff 000000 320c00df 00000000 00120103 80211578 00>
VRAM	String	1024
LoadVBios	String	No
PatchVBios	String	No
VideoPorts	String	2
FBName	String	Makaka
NVCAP	String	040000000000003000C0000000000000A000000000
display-cfg	String	03010300FFFF0001
▶ PCI	Dictionary	(6 items)

Eine weitere Möglichkeit, EDID zu erstellen, ist die Verwendung des ViewSonic EDID Editors (Version 3.1.5), der, falls gewünscht, leicht auf OSX portiert werden kann. Es gibt auch OSX-Editoren, zum Beispiel AW_EDID_Editor. Aber das betrifft nicht Clover selbst. Studieren Sie die Theorie. Clover gibt Ihnen die Möglichkeit, Ihre EDID zu injizieren, eine gute, qualitativ hochwertige. Es gibt auch Berichte, dass Apple-Treiber den Hersteller überprüfen, so dass ein Patch erfunden wurde, damit die automatisch extrahierte EDID bei Apple korrigiert wird.

```
<key>ProduktID</key>
<string>0x9221</string>
```

```
<key>VendorID</key>
<string>0x1006</string>
```

Ich kann nicht sagen, dass es jemandem in irgendeiner Weise geholfen hat. Eher das Gegenteil, substituierte Werte, und ist die Helligkeit nicht mehr einstellbar.

Neue Keys:

```
<key>HorizontalSyncPulseWidth</key>
<string>0x11</string>
```

Es gibt einen solchen Parameter in der EDID-Spezifikation, er steht in den Bytes 63 und 65 im Abschnitt Detailed Timing. Einige Hacker entdeckten, dass der Parameter ein bekanntes Problem von acht Äpfel sind. Suchen Sie in den Foren nach Erklärungen, wer was mit der Änderung dieser Parameter macht. Ja, das tut es, und zwar sehr sogar!

```
<key>VideoInputSignal</key>
<string>0x80</string>
```

Der zweite Parameter stammt von denselben Hackern. Es ist Byte 0x14 in der EDID und definiert die Eigenschaften des Anschlusses:

Bit 7 - analog=0 oder digital=1.

Die Bits 6 - 1 sind nur für analoge Signale definiert.

Bit 0 für digital bedeutet, dass das Signal mit dem VESA DFP 1.x Standard kompatibel ist

```
<key>VideoPorts</key>
<integer>2</integer>
```

Anzahl der Videoausgänge auf der Karte, einschließlich TVO und/oder HDMI. Ausgewählter Rahmen aus Die Liste von Apple stimmt möglicherweise nicht mit unserer aktuellen Karte überein. Beeinflusst die Anzahl der von injizierbaren Steckern. Kann helfen, imaginäre Monitore zu bekämpfen.

```
<key>FBName</key>
<string>Makaka</string>
```

Dieser Parameter ist spezifisch für ATI Radeon, für die es drei Dutzend verschiedene Framebuffer gibt, ohne dass es ein Muster gibt. Clover wählt automatisch den am besten passenden Namen aus der Tabelle auf den meisten bekannten Karten. Andere Benutzer der exakt gleichen Karte haben, argumentieren jedoch, dass sie einen anderen Namen wünschen. Schreiben Sie also in diesen Parameter, was Ihrer Meinung nach am korrektesten ist. Generell gilt: Wenn Sie nicht wissen, was Sie schreiben sollen, löschen Sie den Parameter ganz.

Aber schreiben Sie nicht diesen Makaken! Speziell für Absurdität vorgeschrieben - nein, kopieren Sie es trotzdem in Ihre Config!

Es gibt einen Gedanken, dass der ganze Unterschied in diesen Rahmen in der Zahl der Anschlüsse besteht, und da Sie daran gehen, sie sowieso zu patchen, macht es kein Unterschied, welchen sie als Grundlage nehmen. Außer, dass der Rahmen zur Grafikkartenfamilie passen muss. Zum Beispiel wird Wormy nicht mit Radeon 6670 funktionieren.

```
<key>RadeonDelnit</key>
<true/>
```

Dieser Schlüssel funktioniert mit ATI/AMD Radeon-Karten 6xxx und höher. Oder vielleicht auch 5xxx, ich habe keine Bewertungen darüber gesehen. Es fixiert den Inhalt der GPU-Register, so dass die Karte ordnungsgemäß initialisiert und die MacOSX-Treiber ordnungsgemäß mit ihm arbeiten kann. Die Karte ist beim Start und beim Aufwachen nach dem Schlaf aktiviert. Dank an vit9696 und Mieke.

<key>NVCAP</key>

<string>0400000000003000C0000000000000A0000000</string>

Dieser Parameter für NVidia-Grafikkarten konfiguriert die Typen und Ziele der Video-Ports. In diesem Zeile mit 40 hexadezimalen Ziffern in Großbuchstaben. Es gibt hier keine Theorie, es gibt Empirie, und das mit widersprüchlichen Ergebnissen. Es gibt eine solche Tabelle, aber ihre die Korrektheit ist umstritten.

Yellow = Desktop 1 (primary) Green = Desktop 2 (secondary)

DVI+VGA						
	TV	DVI 2	VGA 2	VGA 1	Bin string	Conversion to hex
Desktop 1	0	0	0	1	1	1
Desktop 2	1	1	1	0	1110	0e
NVCAP	TV + DVI + VGA2			VGA1	04000000 00000100 0e000000 00000007 00000000	

DUAL DVI						
	DVI 2	VGA 2	DVI 1	VGA 1	Bin string	Conversion to hex
Desktop 1	0	0	1	1	11	3
Desktop 2	1	1	0	0	1100	0c
NVCAP	DVI2+VGA2		DVI1+VGA1		04000000 00000300 0c000000 00000007 00000000	

DUAL DVI + TV on hardware channel 2 (maybe not supported)							
	TV	DVI 2	VGA 2	DVI 1	VGA 1	Bin string	hex
Desktop 1	0	0	0	1	1	11	3
Desktop 2	1	1	1	0	0	11100	1c
NVCAP	DVI2+VGA2+TV			DVI1+VGA1		04000000 00000300 1c000000 00000007 00000000	

DUAL DVI + TV on hardware channel 1 (maybe not supported)							
	DVI 2	VGA 2	TV	DVI 1	VGA 1	Bin string	hex
Desktop 1	1	1	0	0	0	11000	18
Desktop 2	0	0	1	1	1	111	7
NVCAP	DVI2+VGA2		DVI1+VGA1+TV			04000000 00001800 07000000 00000007 00000000	

The hardware channel and desktops are intentionally swapped so that TV out stays in use for secondary desktop.

DUAL DVI + TV on hardware channel 1 & 2								
	TV2	DVI2	VGA2	TV1	DVI1	VGA1	Bin string	to hex
Desktop 1	0	0	0	1	1	1	111	7
Desktop 2	1	1	1	0	0	0	111000	38
NVCAP	TV2+DVI2+VGA2			TV1+DVI1+VGA1			04000000 00000700 38000000 00000007 00000000	

Laptop with VGA + TV out						
	TV	DVI 2	VGA 2	LVDS	Bin string	Conversion to hex
Desktop 1	0	0	0	1	1	1
Desktop 2	1	0	1	0	1010	5
NVCAP	External VGA+TV			LCD	04000000 00000100 05000000 00000007 00000000	

Laptop with DVI + TV out						
	TV	DVI 2	VGA 2	LVDS	Bin string	Conversion to hex
Desktop 1	0	0	0	1	1	1
Desktop 2	1	1	1	0	1110	0e
NVCAP	External DVI+VGA+TV			LCD	04000000 00000100 0e000000 00000007 00000000	

Das erste Byte ist immer 04 (bei MacBook 05!). Zweites Byte LID=01 für Laptops. Weitere Möglichkeiten zur Berechnung des korrekten Wertes dieser Zeichenfolge finden Sie in den Foren. Clover versucht, es selbst aus dem BIOS herauszufinden.

```
<key>Display-cfg</key>
<string>03010300FFFF0001</string>
```

Auch dies ist ein Parameter ist nur für NVidia-Karten. Siehe die Diskussionen für Details <http://www.insanelymac.com/forum/topic/215236-nvidia-injection/>

Die dort gemachten Angaben sind jedoch fragwürdig. Die aktuellen Configs können sein siehe <http://www.projectosx.com/forum/index.php?showtopic=370>

A Eigentlich scheint die Standardkonfiguration, die Clover erstellt, die beste Option zu sein. Geben Sie diesen Parameter einfach gar nicht an, lassen sie es Clover berechnen.

```
<key>NvidiaGeneric</key>
<true/>
```

Wenn ja, dann wird anstelle des Namens Gigabyte Geforce 7300LE der Name NVIDIA Geforce 7300LE verwendet Ich weiß nicht, warum. Vielleicht braucht jemand echte Daten, und jemand, der dem Gegner ähnlich ist(?).

```
<key>NvidiaSingle</key>
<false/>
```

Aus der gleichen Serie von obskuren Patches. Wenn es zu etwas nutze ist, dann injizieren sie damit nur die erste Karte, die andere nicht.

```
<key>NvidiaNoEFI</key>
<false/>
```

Fügt dem Nvidia-Injektor die Eigenschaft NVDA hinzu,noEFI Erklärungen von FredWst <http://www.insanelymac.com/forum/topic/306156-clover-bugissue-report->

and-patch/page-107?p=2443062#entry2443062
Behauptet, dass ohne es, der Bildschirm Artefakte auf seiner GT640 produziert.

```
<key>ig-platform-id</key>
<string>0x01620005</string>
```

Dieser Parameter ist für den Betrieb einer Intel HDxxxxx-Grafikkarte erforderlich, das Argument über die spezifische Werte hat nicht zu einer einzigen Regel geführt, daher wird der Parameter einfach in die config eingefügt, -benutzen sie es. Übrigens, Clover wird selbst einige Werte vorschlagen. Jetzt habe ich auch das Ergebnis. Mein Skylake startete nur mit dem Parameter 0x193b0000, es entspricht einer Konfiguration, die wie meine einen HDMI-Ausgang hat.

KernelAndKextPatches

Diese Gruppe von Parametern dient zur Implementierung von binären Patches im laufenden Betrieb. Es sollte beachtet werden, dass dies nur machbar ist, wenn das Laden über den Kernelcache oder über die **ForceKextsToLoad** läuft. Wenn ein kext nicht geladen wird und nicht im Cache vorhanden ist, führen diese Korrekturen nicht zum arbeiten. Ab Version 5119 erfolgen Patches durch interne Algorithmen, die nicht von der Systemversion abhängig sind. Und für ihre eigenen Patches machte die Fähigkeit, nach Zeichen zu suchen.

```
<key>Debug</key>
<true/>
```

Wenn Sie unterwegs beobachten wollen, wie der Kextpatch abläuft. Eigentlich ist dieser Key für Entwickler.

```
<key>KernelCpu</key>
```

```
<true/>
```

Verhindert Kernpanik auf nicht unterstützten CPUs, insbesondere Yonah, Atom, Haswell für ältere Systeme. oder einige Broadwell-E. Heute sollte dieser Patch als veraltet gelten, wenden sie daher stattdessen FakeCUID und andere Kernel-Patches an.

Es sollte klar sein, dass es andere Algorithmen im Kernel gibt, die nicht korrekt mit nicht unterstützte CPU arbeiten, erwarten sie also nicht, dass dieser Patch alle Ihre Probleme löst. Es ist sehr fraglich, ob es mit Pentium M, Pentium 4 oder AMD funktioniert, für solche Fälle ist es immer noch besser, einen speziell entwickelten Kernel zu finden.

<key>FakeCUID</key>

<string>0x010676</string>

Dieser Patch, eingeführt mit Revision 2748, dient dazu, Kernel-Cpu zu ersetzen. Es blockiert nicht nur Kernel-Panik, er fälscht auch die Prozessor-ID so, dass er bei allen Aufrufen als unterstützt gilt. Insbesondere ist auch das AppleIntelCPUPowerManagement.kext davon betroffen. B In diesem Beispiel ersetzt er die ID des Penryn-Prozessors, der von allen OSX-Versionen von 10.5 bis 10.14 funktionierte, bis dann Schluß war.

<key>AppleIntelCPUPM</key>

<true/>

Es stellt sich heraus, dass das BIOS auf ASUS-Motherboards ([wie oft verdirbt ASUS unsere Laune?](#)) das Register 0xE2 Bit 14 in MSR schreibt, und das Register wird ReadOnly, aber es wird im AppleIntelCPUPowerManagement-Kext verwendet, und es wird durch Schreiben verwendet.

Den Autoren dieses Fixes ist nichts Besseres eingefallen, als die die einzige Möglichkeit, die frühere Funktionalität des E2-Registers wiederherzustellen, die darin besteht, es zurückzusetzen.

Setzen Sie dieses Kontrollkästchen auf Ja, wenn beim Systemstart eine Panik auftritt. (Ja, das E2-Register hat WriteOnce-Eigenschaft, d. h. Sie können nur einmal darauf schreiben, bevor Sie neu booten). Relevant für Sandy-Prozessoren und höher. Entweder flashen Sie das BIOS. Und wie funktionieren die anderen Betriebssysteme in in diesem Fall? Es soll auch für Windows geeignet sein.

<key>AppleRTC</key>

<true/>

Das OSX-Betriebssystem geht irgendwie nicht so mit dem CMOS um, wie es das BIOS tut, Dadurch wird das CMOS beim Aufwachen aus dem Ruhezustand oder beim Neustart zurückgesetzt. Nicht bei von allen, Mainboards von Gigabyte sind mehr an dieser Sünde beteiligt. Außerdem wird dieses Problem oft einfach durch Patchen von DSDT: Device(RTC) gelöst, was Clover auch tut.

In manchen Fällen hilft dieser Patch jedoch nicht. Dann können Sie den Kext selbst reparieren AppleRTC, was hier auch gemacht wird. Veraltet! vit9696 untersuchte das Problem und korrigierte den RTC-Betrieb in Clover, jetzt ist der empfohlene Schlüsselwert <false/>, weil er den Winterschlaf beeinflusst. Obwohl, das ist fraglich, ich habe sowieso eine Hibernations-key, die wird im NVRAM gespeichert.

<key>KernelLapic</key>

<false/>

HP-Laptops haben ein Problem mit Lapic, das durch den Betrieb mit cpus=1 gelöst wird, oder jetzt mit mit diesem <true/> Patch. Das Problem und die Lösung gibt es schon sehr lange, seit den Tagen von Chameleon, aber bevor die neuen Entwickler an HP Laptops ernsthafter herausfinden, was der Trick hier ist.

<key>KernelPM</key>

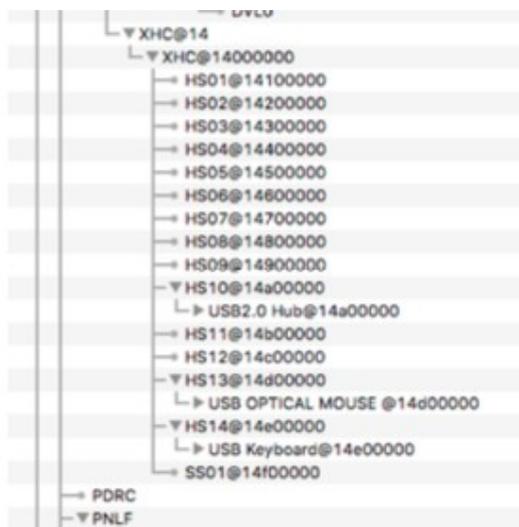
<false/>

Es stellt sich heraus, dass ab dem 10.9-System eine CPU-PM-Steuerung direkt in die Kernel existiert. Dieser Patch verhindert eine Kernel-Panik, für die Fälle, in denen 0xE2 im BIOS gesperrt ist.


```
dict>
<key>Name</key>
<string>AppleAHCIPort</string>
<key>Find</key>
<data>RXh0ZXJuYWw=</data>
<key>Replace</key>
<data>SW50ZXJuYWw=</data>
</dict>
```

Zur Auswahl eines MacPro4.1- oder 5.1-Modells ohne Speicher mit ESS. AppleTyMCEDriver patch Entdeckt und einfachere Methode -nehalem_error_disable deaktiviert den AppleTyMCEDriver Danke AkimoA.
 Einer der nützlichsten Patches ist die Aufhebung der Begrenzung der Anzahl der USB-Ports im Controller

Einer der nützlichsten Patches ist die Aufhebung des Limits für die Anzahl der Ports im USB3-Controller. Das Problem ist, dass der kext alle Ports zuerst als USB2, dann als USB3 nummeriert, und die Gesamtsumme nach Apples Vorstellung 15 Ports nicht überschreiten sollte. Ich veranschauliche das Ergebnis auf meinem Computer



Das heißt, durch die Neuberechnung der 14 HS (usb2)-Ports wurde nur ein SS (usb3)-Port akzeptiert, während es doch 10 davon gibt. Der Patch sieht wie folgt aus (abhängig von der Systemversion)

```
<dict>
<key>Find</key>
<data>g32UDw+DlwQ=</data> <key>Comment</key>
<string>USB 3.0 limit High Sierra 10.13.4</string>
<key>Disabled</key>
<false/>
<key>MatchOS</key>
<string>10.13</string>
<key>Name</key>
<string>com.apple.driver.usb.AppleUSBXHCI</string>
<key>Replace</key>
<data>g32UGA+DlwQ=</data>
</dict>
```

Das Beispiel illustriert übrigens zusätzliche Patch-Keys:

```
<key>Disabled</key>
<true/>
```

Sie können fragliche Keys in der Konfiguration deaktivieren, um sie später in der Oberfläche von Clover zu aktivieren:

```
<key>Comment </key>
<string>USB 3.0 Grenze High Sierra 10.13.4</string>
```

Der Kommentar ist auch in der Clover-Oberfläche sichtbar, hat aber keine Auswirkungen auf das System.

```
<key>MatchOS</key>
<string>10.13</string>
```

Es kommt oft vor, dass ein bestimmter Patch nur für eine bestimmte Version des Systems anwendbar ist, für eine andere Version wird einen weiteren Patch benötigt. Schreiben Sie beides, und geben Sie die Version, einschließlich der vollständigen 10.13.5 oder eine verkürzte, wie im Beispiel. Es ist jedoch unwahrscheinlich, dass Sie zwei Systeme 10.13.2 und 10.13.4 haben, Die Vollversion macht also nicht viel Sinn, vergessen Sie nur nicht, die Patches mit zu aktualisieren bei einem System-Update.

Clover in khakifarben. Version 5.1, Revision 5129
Moskau, 2021

89

Manchmal ist es notwendig, nicht den binären Teil einer Textdatei zu bearbeiten, sondern deren info.plist. In diesem Fall ist der Abschnitt ist wie folgt

```
<dict>
  <key>Name</key>
  <string>AppleHDAController</string>
  <key>Comment</key>
  <string>Patch_to_not_load_this_driver</string>
  <key>InfoPlistPatch</key>
  <true/>
  <key>Find</key>
  <string>0x04020000</string>
  <key>Replace</key>
  <string>0x44220000</string>
</dict>
```

Hier gibt es eine Komplikation. Der Patch soll in einem Kernelcache durchgeführt werden, aber wenn wir die Info-Liste patchen, damit der Kext geladen wird, ist dieser Kext noch nicht im Cache, weil er noch nicht geladen wurde. Daher müssen Sie zweimal booten. Erstes Mal mit Cache ignoriert (Taste NoCache), dann wird FSInject diesen Cache laden, und beim zweiten Mal mit dem Cache, wo es Patch erfolgreich gepatcht wurde. ForceKextsToLoad in der Config einstellen.

In der Revision 3154 und dann in der Revision 3256 wurde die Patch-Info-Liste korrigiert (Dank an Benutzer solstice). Jetzt können Sie mehr als eine Zeile in die Suche einbeziehen, indem Sie alle unsichtbaren Zeichen ausschließen, z. B. Zeilenumbruch und Tabulierung. Die Suche sollte nun in der Form <Daten> eingestellt werden, da der Dienst Zeichen, wie z. B. "<", können nicht in Textform angegeben werden. Längen der Such- und Ersetzungszeichenfolgen können unterschiedlich sein, sollten aber auf die gleiche Länge gesetzt werden, ergänzt durch Leerzeichen. Beispiel

```
dict>
  <key>Comment</key>
  <string>Power state 1 - 0</string>
  <key>Name</key>
  <string>AppleIntelHDGraphicsFB</string>
  <key>InfoPlistPatch</key>
  <true/>
  <key>Find</key>
  <key>Find</key>
  <data>PGtleT5Qb3dlclN0YXRlcwva2V5PjxpbmRlZ2VyPjE8L2ludGVnZXI+</data>
  <key>Replace</key>
  <data>PGtleT5Qb3dlclN0YXRlcwva2V5PjxpbmRlZ2VyPjA8L2ludGVnZXI+</data>
</dict>
```

Wenn Sie eine Beispielkonfiguration mit vielen Patches haben, aber in einem bestimmten Fall nur einige von ihnen gebraucht werden, können die zusätzlichen verbannt werden:

```
<key>Disable </key>
<true/>
```

Dies war sehr bequem mit dem Property List Editor zu machen, wo <true/> wird durch ein einfaches Kontrollkästchen dargestellt. In den Revisionen des Clover 3990+ können diese Patches zugelassen oder verboten werden im Clover-Menü, indem Sie dieses Kontrollkästchen aktivieren.

Patching with Mask

Beginnend mit Revision 5095 wurde die Möglichkeit eingeführt, binäre Patches mit einer Maske durchzuführen. Dies gilt für KextPatches, KernelPatches und BootPatches. Ich erkläre es Ihnen an einer anderen Stelle, wobei ich alle drei Punkte im Auge behalte. Es sieht so aus (spezifische Daten sind unrealistisch, nur als Methode). So können wir neben der Hex-String-Suche auch MaskFind einstellen, Bit. Wenn irgendein Bit = 1, nach exakter Übereinstimmung suchen, wenn =0, ignorieren den Unterschied. Und für die Zeichenkette Replace, MaskReplace bedeutet die Maske, dass Bit=1 - Ersetzung vornehmen, Bit=0 - lassen wir es so, wie es war.

Clover in khakifarben. Version 5.1, Revision 5129
Moskau, 2021

90

Beispiel:

1. Sucht alle Zeilen in dem angegebenen Kext, {funktioniert auch im Kernel oder in boot.efi} clever oder Clever. Der Unterschied ist der erste Buchstabe, er unterscheidet sich durch Bit 0x20. Das heißt, wir setzen die Suchmaske DF FF FF FF FF FF FF FF. Das bedeutet, dass wir alle Bytes (Buchstaben) abgleichen, außer des ersten, in dem Groß- oder Kleinbuchstaben ignoriert werden. Die Suchmaske kann reduziert werden, da standardmäßig alle fehlenden Bytes als FF angenommen werden. Dies bedeutet, dass nicht spezifizierte Bytes abgeglichen werden müssen

2. Ersetzen Sie den dritten Buchstaben in den gefundenen Wörtern durch "o", d. h. wir erhalten Clover oder Clover, beziehungsweise.

MaskReplace = 00 00 FF 00 00 00

Der String kann nach rechts gekürzt werden, da er mit Null gefüllt werden soll. In diesem Fall diese Bytes, die wir nicht ersetzen, könnten wir auch nicht in Ersetzen angeben, aber es gibt eine Anforderung eine exakte Längenübereinstimmung.

Um die Abwärtskompatibilität des neuen Clover mit der alten Config zu erhalten, wird angenommen, dass die nicht spezifizierte Maske (nicht vorhanden) alle FFFFFFFF sind, d. h. exakte Übereinstimmung mit der Suchzeichenfolge und ersetzt alle Bytes vollständig durch die angegebenen.

Symbolic patching.

Ab der Revision 5119 haben wir mehr Such- und Patching-Optionen.

Allgemeine Syntax

```
<dict>
  <key>Comment</key>
  <string>Symbolic patch example got lapic panic</string>
  <key>MatchOS</key>
  <string>All</string>
  <key>Disabled</key>
  <true/>
  <key>Procedure</key>
  <string>_lapic_interrupt</string>
  <key>RangeFind</key>
  <integer>200</integer>
  <key>StartPattern</key>
  <data>ACnHeAAx241H+oM=</data>
  <key>MaskStart</key>
  <data>////wA=</data>
  <key>Find</key> <data>6AAA//+DAAAAAAAA</data>
  <key>MaskFind</key> <data>/wAA/////AAAAAP//</data>
  <key>Replace</key>
  <data>6AAA//8xwJCQkJCQ</data>
  <key>MaskReplace</key>
  <data>/wAA/////////</data>
</dict>
```

MatchOS ist auf Alle eingestellt, da wir diese Methode des Patchens als unabhängig von der Systemversion betrachten.

Disabled ist erst einmal richtig, da dies ein unrealistisches Beispiel ist.

Prozedur hier ist der Name der Prozedur, nach der wir suchen. Der eigentliche Name kann länger sein, aber der Vergleich basiert auf der Teilzeichenkette. Achten Sie darauf, dass die Teilzeichenkette nur in dieser Prozedur vorkommt.

RangeFind ist die Länge der Codes, nach denen gesucht werden soll. Im Allgemeinen nur die Größe dieses Verfahrens, oder weniger. Auf diese Weise beschleunigen wir die Suche, ohne alle Millionen von Zeichenketten durchzugehen.

StartPattern wurde vor dem Zeichen-Patch erfunden. Es ist der Ausgangspunkt, von dem aus wir nach unserem Muster suchen. Wenn wir den Namen der Prozedur kennen, wird das StartPattern kaum benötigt. Trotzdem soll es so sein. Es wird auch RangeFind genannt.

MaskStart ist eine Maske für einen Startpunkt, d. h. für StartPattern. Und dann gibt es noch die Paare **Find/MaskFind** und **Replace/MaskReplace**.

Ab Revision 2814 ist es nun möglich, das Laden von Kexten zu erzwingen:

```
<key>ForceKextsToLoad</key>
<array>
  <string>\System/Library/Extensions/AppleHDA.kext</string>
</array>
```

Damit wird die Ladeunwilligkeit der der Kexte überwunden. IOXXXFamily, die zum Laden des Haupt-Kext benötigt wird, aber von dieser Familie abhängt. Zum Beispiel, IONetworkFamily.

Oder sogar einen ganzen Ordner \Extra\Extensions (Revision 2816+). Dies impliziert Ordner auf der Hauptpartition, andere Partitionen/Volumes/Disks sind nicht vorgesehen.

Achten Sie auf den Schrägstrich Schrägstrich! Der Treiber FSInject.efi wird benötigt, um diese Funktion auszuführen.

Funktioniert nicht nach dem Zusammenführen von Clover mit Opencore,(Versions 5124+).

```
<key>ATISConnectorsController</key>
```

```
<string>6000</string>
```

Karten der Serien ATI (AMD) Radeon 5000 und 6000 reichen nicht aus, um die die Eigenschaften in die Registrierung zu injizieren, müssen Sie auch die Anschlüsse in den entsprechenden Controller angeben. Geben Sie in diesem Fall den Controller 6000 an. Die folgenden Zwei Eigenschaften geben an, was zu finden und was zu ändern ist.

```
<key>ATISConnectorsData</key>
```

```
<string>00040000040300000001000021030204040000001402000000010000000004031000000010000000
00010000000000001</string>
```

```
<key>ATISConnectorsPatch</key>
```

```
<string>04000000140200000001000000000404000400000403000000010000110201050000000000000000
0000000000000000</string>
```

Diese Methode funktioniert nur für Systeme 10.7 und höher.

In 10.12 werden die Anschlüsse anders sein, daher sollte diese Methode als veraltet betrachtet werden, obwohl die Berechnungsmethodik ist immer noch die gleiche.

Lassen Sie mich Ihnen mehr darüber erzählen, wie Sie diese Zahlen erhalten. Originalartikel von bcc9

<http://www.insanelymac.com/forum/index.php?showtopic=249642>

Vollständiges Rezept von Xmedik auf Russisch mit Diskussion

<http://www.applelife.ru/threads/3авод-ати-hd-6xxx-5xxx-4xxx.28890/>

Ich mache es hier kürzer und berücksichtige dabei die Besonderheiten von Clover.

1. Zuerst müssen Sie sich Ihr Video-Bios besorgen. Booten Sie in CloverGUI und drücken Sie F6.

Ihr Bios wird in /EFI/CLOVER/misc/c0000.bin gespeichert, wenn Clover natürlich in einer Partition mit einem FAT32-Dateisystem installiert ist.

2. Laden Sie das Programm radeon_bios_decode von einem der folgenden Links herunter. Legen Sie die Bios-Datei c0000.bin in denselben Ordner wie dieses Dienstprogramm. Gehen wir davon aus, dass es sich um den Ordner ~/RadeonPatch handelt

Führen Sie im Terminal die folgenden Befehle aus

```
cd ~/RadeonPatch ./radeon_bios_decode < c0000.bin
```

3. Der Bildschirm gibt Ihnen Informationen zu Ihren Anschlüssen, die Sie für spätere Verwendung Kopieren/ Fotografieren sollten.

Hier ist, was ich bekommen habe:

```
iMac:test slice$ ./radeon_bios_decode
<c0000.bin ATOM BIOS Rom:
  SubsystemVendorID: 0x1458 SubsystemID: 0x2557
  IOBaseAddress: 0xe000
  Filename: R667D32I.F1
  BIOS Bootup Message:
GV-R667D3-2GI/F1

PCI ID: 1002:6758
Connector at Index 0
  Typ [@Offset 44282]: HDMI-A (11)
  Encoder [@Offset 44286]: INTERNAL_UNIPHY2 (0x21) i2cid [@Offset 44356]: 0x92, OSX
  senseid: 0x3
Connector at Index 1
  Typ [@Offset 44292]: DVI-D (3)
  Encoder [@Offset 44296]: INTERNAL_UNIPHY (0x1e) i2cid [@Offset 44383]: 0x95, OSX
  senseid: 0x6
Connector at Index 2
  Typ [@Offset 44302]: VGA (1)
  Encoder [@Offset 44306]: INTERNAL_KLDSCP_DAC1 (0x15) i2cid [@Offset 44410]: 0x90,
  OSX senseid: 0x1
```

Khaki Clover. Version 5.1, Revision 5129
Moskau, 2021

92

4. Laden Sie das Skript ati-personality.pl von einem der folgenden Links herunter
5. Legen Sie es in denselben Ordner und führen Sie es im Terminal aus:
perl ati-personality.pl -386 >frames.txt
wenn Sie es für ein 32-Bit-System tun, oder perl ati-personality.pl >frames.txt für 64-Bit.
Warnung. In Sierra haben sich die Texte geändert, so dass sich der Patch als systemabhängig erweist.
6. Nun müssen wir einen geeigneten Framebuffer auswählen. Apple bietet uns eine große Auswahl:
birds, fish, and even monkeys. Aber die wirklichen Unterschiede sind hauptsächlich in den
Anschlüssen, was wir ändern werden. Wenn es nicht zu
Wenn man darüber nachdenkt, ist es eine einfache Muntermacher-Option:
Serie 5000: Mobil - Alouatta, Desktop - Baboon
Serie 6000: Mobil - Cattail, Desktop - Ipomoea
Serie 7000: Mobil - Pondweed, Desktop - Futomaki.

7. Für den ausgewählten Framebuffer nehmen wir einen Ausdruck der Anschlüsse aus unserer Datei frames.txt, die in Schritt 5 erhalten wurde.

```
0000000 00 04 00 00 04 03 00 00 00 01 00 00 12 04 01 05
0000010 00 08 00 00 04 02 00 00 00 01 00 00 11 02 04 03
0000020 10 00 00 00 10 00 00 00 00 01 00 00 00 00 00 02
```

Die Zahlen, die korrigiert werden müssen, sind rot hervorgehoben. Blaue Zahlen sind nur Adressen und müssen verworfen werden. Die dritte Stelle vom Ende ist die encoderid, die letzte Stelle - senseid. Die ersten 4 Ziffern in jeder Zeile sind Typ des Monitors (genauer gesagt, Typ des Anschlusses).

ConnectorType

02	00	00	00	LVDS
04	00	00	00	DVI_DL(Dual Link)
00	02	00	00	DVI_SL(Single Link)
10	00	00	00	VGA
80	00	00	00	S-Video
00	04	00	00	DP

Khaki Clover. Version 5.1, Revision 5129
Moskau, 2021

93

00 08 00 00 HDMI

8. die Senseid, die wir in Schritt 3 für jeden unserer Anschlüsse erhalten haben. Wir können einfach alles auf Null setzen. Achten Sie nicht auf den Rest der Zahlen.

Wir erhalten die folgende Tabelle:

```
0000000 04 00 00 00 04 03 00 00 00 01 00 00 10 00 01 06
0000020 10 00 00 00 10 00 00 00 00 01 00 00 00 00 00 01
0000010 00 08 00 00 04 02 00 00 00 01 00 00 12 00 04 03
```

D.h., die erste Zeile ist DVI-D, die zweite Zeile ist VGA, die dritte Zeile ist HDMI, alle mit meinem senseid-Werte.

9. Und ein weiteres Rezept von Sergey_Galan. <http://www.applelife.ru/threads/mobility-ati-radeon-hd5650m-hd5470m-hd4570m-hd4650m.29028/page-58#post-379044>

Die zweite Ziffer vom Ende der HotPlugID muss in der Reihenfolge 00, 01, 02 erfolgen. Dies betrifft das Einschlafen und Aufwachen. (rot hervorgehoben)

```
0000000 04 00 00 00 04 03 00 00 00 01 00 00 10 00 00 06
0000020 10 00 00 00 10 00 00 00 00 01 00 00 00 00 00 01
0000010 00 08 00 00 04 02 00 00 00 01 00 00 12 00 02 03
```

10. Nachdem Sie die blauen Ziffern verworfen haben, geben Sie den Rest ohne Leerzeichen und Bindestriche und Zeilenschaltung in die config.plist ein. Die ursprüngliche Tabelle in ATIConnectorsData, nach unseren Bearbeitungen in ATIConnectorsPatch. Siehe das Beispiel oben im Text.

11. Ich habe auch schon eine Situation gesehen, in der der VGA-Anschluss unter den Anschlüssen dargestellt wurde als DVI-I (DVI-SL). Und der Patch hat bei dieser Verwendung funktioniert.

12. Rezept von eierfrucht <https://applelife.ru/threads/sony-vaio-vpceb3m1r.522504/page-3#post-537081>:

Der interessanteste Teil sind die FEATURES-Bits am LVDS-Anschluss. versuchen Sie es mit 08 01, 08 00, 09 01 und 09 00, einer davon sollte alles starten machen und um normal aufzuwachen. (Orange hervorgehoben)

```
0000000 02 00 00 00 40 00 00 00 09 01 00 00 00 00 00 07
0000010 00 04 00 00 04 06 00 00 00 73 00 00 11 02 01 01
```

13. dort, "Senseid LVDS panel put 0x7", weil es ein Sony VAIO ist.

Der neue Weg ist

```
<key>ForceKextsToLoad</key>
<array>
<string>\System\Library\Extensions\AMD6000Controller.kext</string>
<string>\System\Library\Extensions\AMDFramebuffer.kext</string>
</array>
<key>KextsToPatch</key>
<array>
<<dict>>
<key>Comment </key>
<string>ATI Connector patch new way</string>
<key>Deaktiviert</key>
<false/>
<Taste>Finden</key>
<key>MatchOS</key>
<string>10.9,10.10,10.11</string>
<key>Name</key>
<string>AMD6000Controller</string>
<key>Replace</key>
```

```
<data>BAAAABQCAAAAAAAAAAAAAEBAAEAAAAAAAAAAAAAAwAAAAAAAAABECAQUAAAAAAAAAAAAAAAAAAAAAA</data>
</dict>.
</array>.
```

Bei 10.12 duplizieren ganze <dict> mit anderen Suchen/Ersetzen-Zeilen, sie sind länger. (?)

Khaki Clover. Version 5.1, Revision 5129
Moskau, 2021

94

Wir können auch binäre Patches für boot.efi oder für den Kernel angeben. Der Weg ist das Gleiche, deshalb zeige ich Ihnen ein Beispiel.

```
<key>KernelToPatch</key>
```

```
<key>BootPatches</key>
```

Weitere Informationen zu Masken finden Sie unter Patching mit Maske. Ich möchte darauf hinweisen, dass diese Gruppe von Patches praktisch von niemandem verwendet wird, außer vom Entwickler, um in schweren Fällen nach Fehlern zu suchen, die nicht offensichtlich sind, wenn eine externe Inspektion erfolgt. So haben wir nach Hibernationsproblemen gesucht, so haben wir nach dem Moment gesucht, in dem E2 im Kernel gebrochen wird.

Geräte

Parametergruppe für andere PCI-Geräte und den Bus im Allgemeinen.

```
<key>Inject</key>
<false/>.
```

Setzen Sie diesen Wert auf true - alle internen Injektionen werden durch Einzeleingaben ersetzt Eigenschaften, was Apples Injektion APPLE_GETVAR_PROTOCOL entspricht mit GUID={0x91BD12FE, 0xF6C3, 0x44FB, {0xA5, 0xB7, 0x51, 0x22, 0xAB, 0x30, 0x3A, 0xE0}}; und die auf den echten Macs verwendet werden. Die alten Hacker nannten es EFIstrings.

```
<key>Eigenschaften</key>
<string>0207364862FA54HG345</string>
```

Veraltet! Ab der Revision 4497 wird nicht mehr eine Zeichenkette aus 16 Hex-Zeichen gebildet, sondern ein Wörterbuch aus gfxutil-Regeln.

```
<dict>
  <key>PciRoot(0x0)/Pci(0x14,0x0)</key>
  <dict>
    <key>AAPL,clock-id</key>
    <data>AA==</data>
    <key>AAPL,current-available</key>
    <data>SAQ=</data>
    <key>AAPL,current-extra</key>
    <data>vAI=</data>
    <key>AAPL,current-in-sleep</key>
    <data>6AM=</data>
    <key>built-in</key>
    <data>AA==</data>
    <key>device_type</key>
    <string>XHCI</string>
  </dict>
  <key>PciRoot(0x0)/Pci(0x19,0x0)</key> <dict>
    <key>built-in</key>
    <data>AQ==</data>
  </dict>
```

Sie können z. B. mit DarwinDumper sehen, welche plist von Clover erzeugt wurde standardmäßig, d. h. automatische Injektionen durchführen, oder wie Sie es getan haben, indem Sie die alte, booten Sie in das System, und im Systemaufruf im Terminal

```
clover-genconfig >config-gen.plist.
```

Dort suchen Sie ein solches Wörterbuch, und kopieren es in Ihre Konfiguration, und die alten Inject-Methoden schalten Sie alle aus. Sie können ein solches Wörterbuch auch im DarwinDumper-Bericht aus der Gegenwart nachschlagen auf einem Mac mit einer ähnlichen Konfiguration. Die Syntax ist die gleiche. Nur Clover versteht ein bisschen mehr,

Es versteht <string>, <integer>, <true>, <false>, <real>.

Khaki Clover. Version 5.1, Revision 5129
Moskau, 2021

95

Heute sind die Funktion clover-genconfig und der plist-Editor in der Clover.app enthalten. Nutzen Sie den Vorteil! Im Prinzip wird das gleiche Ergebnis durch Einfügen von _DSM-Methoden in DSDT erreicht, wenn es bereits ist, und wenn Sie es perfektionieren wollen. Es liegt an jedem selbst. Im Prinzip sind Properties besser, als _DSM, es ist nativ für den **Realmac(?)** und wird vor dem Start des Kernels ausgelöst.

```
<key>PCIRootUID</key>  
<ganzzahl>0</ganzzahl>
```

Es stellt sich heraus, dass die Injektion von Grafikkarteneigenschaften davon abhängt, welche Zahl in DevicePath=PciRoot(0x0) oder PciRoot(0x1). Bisher wurde angenommen, dass es sich um eine Hardware Merkmal. In den Anfängen des Hackintoshaping stellte sich jedoch heraus, dass diese Zahl einfach Kennung, die in der DSDT vorgeschrieben ist. Genau hier:

```
Gerät (PCI0)  
{  
  Name (_HID, Eisald ("PNP0A08"))  
  Name (_CID, Eisald ("PNP0A03"))  
  Name (_ADR, Null)
```

```
  Name (_UID, Null)
```

_UID=Zero - bedeutet 0, aber wenn sie gleich Eins ist, bedeutet sie 1.

Außerdem ändert sich diese Nummer, wenn sie gewaltsam geändert wird, und sie funktioniert dann erfolgreich. Also hier haben echte Macs immer 0. Und entsprechend nimmt boot.efi immer 0 an, daher ist es besser, wenn Sie Ihre DSDT korrigieren, Clover tut dies standardmäßig, und ein solcher Schlüssel gibt es in der Konfiguration nicht mehr.

```
<key>Audio</key>  
<dict>  
  <key>Inject</key>  
  <string>887</string>  
  <key>ResetHDA</key>  
  <true/>  
  <key>AFGLowPowerState</key>  
  <true/>  
</dict>
```

Injektion

Injektion der Properties der Soundkarte. True, dies funktioniert nur, wenn sich das Gerät im DSDT befindet heißt HDEF, aber wenn Sie es umbenennen, kann der Rest anders lauten Methode der Injektion. Auch dieser Aufwand entfällt bei Verwendung des VoodooHDA-Kext (Treibers)

Die Werteoptionen sind wie folgt:

NEIN - es wird nichts injiziert, wenn Sie z. B. die Eigenschaften selbst über Eigenschaften **Detect - automatische Erkennung des installierten Soundchips anhand seiner ID als Layout zu verwenden. Eigentlich Unfug, aber sehr beliebt. In vielen Fällen ist es nicht störend und beeinflusst aber die Anzeige der Soundkarte im System-Profiler.**

883 ist die Dezimalzahl des Layouts. Es bedeutet Realtek ALC883.

0x0373 - das Gleiche in 16er-Form wird unkenntlich.

Tatsächlich sind diese Zahlen falsch, das korrekte Layout ist z.B. 12 = 0x0C, aber egal wie, sie sind merkwürdigerweise zulässig.

Mit dem Aufkommen von AppleALC hat diese Einstellung ein neues Leben bekommen. Jetzt ist es wirklich Die Nummer des Layouts, und Sie müssen es aus der Liste der empfohlenen Codec für Ihren Sound auswählen. Lesen Sie in der Dokumentation zu diesem Kext nach.

ResetHDA - wenn sich der Soundchip aus irgendeinem Grund nicht einschaltet, kann diese Taste helfen bei der Erstinbetriebnahme. Betrifft auch Windows. Notwendigkeit bemerkt nach dem Neustart von Windows auf Mac.

AFGLowPowerState - es betrifft den AppleHDA-Treiber und scheint das Problem zu lösen mit Klickgeräuschen in den Lautsprechern, wenn die Soundkarte in den Ruhezustand geht/aufwacht. Bestätigungen die Wirksamkeit des Patch ist nicht ausreichend belegt.

```
<key>USB</key>
<dict>
  <key>Inject</key>
  <true/>
  <key>AddClockID</key>
  <true/>
  <key>FixOwnership</key>
  <true/>
  <key>HighCurrent</key>
  <true/>
</dict>
```

Inject

Sie können false eingeben, wenn Sie die USB-Eigenschaften aus irgendeinem Grund nicht injiziert haben möchten, Wenn Sie zum Beispiel die Eigenschaften selbst über Properties regeln.

FixOwnership

Das BIOS übernimmt die Kontrolle über den USB, und bevor der Kernel startet, müssen wir den USB von der BIOS. Für UEFI-Boot scheint es irrelevant zu sein, daher ist es standardmäßig aktiviert für Legacy-Download, und aus für UEFI. Aktuell!

AddClockID

Wenn diese Funktion vorhanden ist, schläft der SSB-Controller absichtlich ein und weckt den Computer nicht auf. Wenn Sie von der USB-Maus aufwachen und hier falsch setzen wollen. Aber seien Sie darauf vorbereitet, dass wird Ihr Computer spontan aufgeweckt, z. B. von der eingebauten Kamera.

HighCurrent

Der erhöhte Strom an diesem USB-Controller wird zum Laden der iPads benötigt, aber ich habe nicht als den Standardwert.

Eine Gruppe von Parametern, um ihre Geräte als nativ für OSX zu maskieren. (1971)

```
<key>FakeID</key>
<Diktat>
<key>ATI</key>.
<string>0x67501002</string>
<key>IntelGFX</key>
<string>0x01268086</string>
<key>NVidia</key>
<string>0x0FE210DE</string>
<key>LAN</key>
<string>0x436311AB</string>
<key>SATA</key>
<string>0x25628086</string>
<key>WIFI</key>
<string>0x431214E4</string>
<key>XHCI</key>.
<string>0x1E318086</string>
```

```
<key>IMEI</key>.  
<string>0x1E3A8086</string>  
</dict>.
```

In dieser Parametergruppe können Sie die Umetikettierung Ihrer nicht unterstützten Gerät in ein unterstütztes Gerät. Beispiele:

Khaki Clover. Version 5.1, Revision 5129
Moskau, 2021

97

- AMDRadeonHD7850 hat DeviceID=0x6819, was von den cxxts nicht unterstützt wird ATI7000Controller und ATIRadeonX3000 im System 10.8. Es gibt jedoch Unterstützung für GeräteID=0x6818. Den Wechsel vollziehen. Damit es wirksam wird, muss es diesen Fake irgendwie zu injizieren. Für Grafikkarten gibt es zwei Optionen: entweder Inject->ATI=true, oder DsdtFixMask enthält 0x0100 (FixDisplay).
- NVidia GTX660 hat DeviceID=0x1183, die Karte funktioniert so oder so, aber AGPM für ist nicht vorgesehen. Wir machen einen Swap auf 0x0fe0 und AGPM ist aktiviert. Da für einer solchen Karte Inject->NVidia=false, dann kann die Ersetzung der ID nur durch DSDT-Patch mit Maske 0x0100 (FixDisplay).
- Die WiFi-Karte im Dell-Laptop heißt Dell Wireless 1595, DeviceID=0x4315, die echte Sache ist Broadcom, die Chips 4312, 4331 und eine Reihe anderer unterstützt hat. Eine Auswechslung vornehmen. DSDT-Patch mit 0x4000-Maske (FixAirport).
- Die übliche Marvell 80E8056 DeviceID=0x4353 Netzwerkkarte funktioniert einfach nicht funktioniert, sondern mit dem AppleYukon2-Treiber, wenn Sie einen ID-Tausch auf 0x4363. DSDT-Patch mit Maske 0x2000 (FixLan).
- IMEI - dieses Gerät arbeitet mit Intel HD3000/4000, es ist jedoch nicht sicher, dass Ihr Chipsatz hat die richtige ID. Die Auswechslungen sind wie folgt:
SandyBridge = 0x1C3A8086
IvyBridge = 0x1E3A8086
Haswell = 0x8C3A8086
Funktioniert mit DSDT-Patch AddIMEI_80000 (AddIMEI)

Diese Maskierung funktioniert in zwei Fällen: beim Einspeisen oder beim Patchen des DSDT. Wenn jedoch wir keine vollständige Injektion, wie von Clover vorgesehen, wollen, können wir die folgende Eigenschaft:

```
<key>NoDefaultProperties</key>  
<true/>
```

In diesem Fall wird die Zeile für die Injektion erstellt, enthält aber keine neue Eigenschaften. Diese Eigenschaft wäre zum Beispiel FakeID. Auch diese Art, FakeID zu machen, ist veraltet, es ist besser, es über Properties auf dem folgendem Weg zu machen:

AAPL,ig-platform-id	⊕ ⊖	Data	⏴ <01001219>
model	⊕ ⊖	String	Intel HD Graphics 7000
subsystem-vendor-id		Data	<6b10>
subsystem-id		Data	<8680>
device-id		Data	<1219>
▶ PciRoot(0x0)/Pci(0x1c,0x4)/Pci(0x0,0x0)/Pci(0x0,0x0)		Dictionary	(1 item)

Sie können weitere eigene Properties, z. B. Model, im folgenden Dictionary-Array hinzufügen
Veraltet! Properties verwenden!

```
<key>Eigenschaften hinzufügen</key>  
<array>.  
<dict>.  
<key>Gerät</key>  
<string>NVidia</string>  
<key>Schlüssel</key>  
<string>AAPL,HasPanel</string>  
<key>Wert</key>  
<data>AQAAAA==</data>
```

</dict>.

...

</array>.

Khaki Clover. Version 5.1, Revision 5129
Moskau, 2021

98

Wert

Wert kann <Daten> oder Hex-String sein. Nur eine Zeichenkette ist nicht erlaubt. Das heißt Anstelle von <Zeichenfolge> ABC sollten wir also <Zeichenfolge>0x414243schreiben Konvertieren über PlistEditor oder Xcode.

Die erste Gerätetaste bestimmt, zu welchem Gerät dies hinzugefügt wird
Eigenschaft. Geräteliste:

ATI
Nvidia
IntelGFX
LAN
WIFI
Firewire
SATA
IDE
HDA
HDMI
LPC
SmBUS
USB

Die Titel sollten genau das sein, Buchstabe für Buchstabe. Ich denke, Erklärungen sind hier nicht notwendig. Auf diese Weise können Sie verschiedene Eigenschaften für analoges Audio einspeisen Device=HDA, und für digitales Gerät=HDMI. Die Unterscheidung von Clover wird laut Anbieter leider nicht sehr korrekt sein. Wenn Intel, dann HDA, wenn ATI oder Nvidia, dann HDMI. Auf Haswell gibt es zum Beispiel Intel HDMI-Ton. Diese Option ist in Clover noch nicht vorgesehen.

Es ist vorgesehen, dass mit Intel Der HDMI-Ausgang verwendet das Audio des HDA-Chipsatzes. Diese wird bedient von Parameter

<key>UseIntelHDMI</key>

<true/>

Diese Einstellung wirkt sich auf die Einkopplungseigenschaften des über HDMI übertragenen Tons sowie auf den DSDT-Patch. Soweit ich weiß, jedoch funktioniert die Soundtreiber, sowohl VoodooHDA als auch AppleHDA, nicht ganz mit HDMI-Ausgang. Nach neuen Informationen funktioniert VoodooHDA nur mit dem HDMI-Ausgang von NVIDIA, und was AMD betrifft, so wurde für das Apples Systemen 10.13+ ein neuer Treiber AppleGFXHDA.kext erstellt. Erkunden Sie die Funktionen.

key>HDMIInjection</key>

<false/>

Damit Verbieten Sie die Einspeisung von HDMI-Geräteeigenschaften ganz.

Ab Revision 3262 wird eine neue Art der Injektion von Geräteeigenschaften eingeführt, die nicht über den Namen erfolgt, wie früher, sondern nach ihrer Position auf dem PCI-Bus. Hier ist die Liste, die im Bootlog von Clover erstellt wird.

```
4:432 0:000 PCI (00|00:00.00) : 8086 2E30 class=060000
4:432 0:000 PCI (00|00:02.00) : 8086 2E32 class=030000
4:432 0:000 Gefunden GFX-Modell=Unbekannt
4:432 0:000 PCI (00|00:02.01) : 8086 2E33 class=038000
4:432 0:000 PCI (00|00:1B.00) : 8086 27D8 class=040300
4:432 0:000 PCI (00|00:1C.00) : 8086 27D0 class=060400
4:432 0:000 PCI (00|01:00.00) : 10DE 01D1 class=030000
4:432 0:000 Gefunden NVidia model=Gigabyte GeForce 7300 LE
4:432 0:000 PCI (00|00:1D.00) : 8086 27C8 class=0C0300
4:432 0:000 PCI (00|00:1D.01) : 8086 27C9 class=0C0300
4:432 0:000 PCI (00|00:1D.02) : 8086 27CA class=0C0300
```

4:432 0:000 PCI (00|00:1D.03) : 8086 27CB class=0C0300
4:432 0:000 PCI (00|02:05.00) : 10EC 8167 class=020000

Khaki Clover. Version 5.1, Revision 5129
Moskau, 2021

99

Hier sehen wir zwei Grafikkarten und vier USB-UHCI-Geräte. Der Gerätetyp wird erkannt ist gemäß seiner Klasse=040300 ein HDA-Standard-Audiogerät. In diesem Fall befindet sich auf 00:1B.00 und die Netzwerkkarte der Klasse 020000 befindet sich auf 02:05.00

Bus= 02

Gerät = 05

Funktion = 00

Wir verwenden diesen Wert, um die Eigenschaften zu injizieren

Veraltet! Eigenschaften verwenden!

```
<key>Beliebig</key>
<array>
<Diktat>
<key>PciAddr</key>
<string>02:05.00</string>
<key>Kommentar</key>
<string>Realtek LAN 8167</string>
<key>BenutzerdefinierteEigenschaften</key>
<array>
<Diktat>
<key>Deaktiviert</key>
<true/>
<key>Schlüssel</key>
<string>Modell</string>
<key>Wert</key>
<string>Realtek 8169 Gigabit Ethernet
Controller</string>
</dict>
<Diktat>
<key>Schlüssel</key>
<string>eingebaut</string>
<key>Wert</key>
<data>AQAAAA==</data>
</dict>
</array>
</dict>
<Diktat>
<key>PciAddr</key>
<string>01:00.00</string>
<key>Kommentar</key>
<string>Nvidia Geforce-Karte im PCIe-Steckplatz</string>
<key>BenutzerdefinierteEigenschaften</key>
<array>
<Diktat>
<key>Schlüssel</key>
<string>Modell</string>
<key>Wert</key>
<string>Gigabyte GeForce 7300 LE</string>
</dict>
<Diktat>
<key>Deaktiviert</key>
<true/>
<key>Schlüssel</key>
<string>AAPL,boot-device</string>
<key>Wert</key>
<data>AQAAAA==</data>
</dict>
</array>
</dict>
</array>
```

Der Abschnitt Arbitrary ist also ein Array von Dictionaries, von denen jedes entspricht einem Gerät mit der angegebenen Adresse, und jedes Gerät zu beschreiben Es wird das Array CustomProperties verwendet, das

aus Schlüssel-Werte-Paaren besteht. Auch eine spezifische Eigenschaft kann mit der Taste Disabled deaktiviert werden. Sie können die Eigenschaft im Clover-Menü dynamisch aktivieren oder deaktivieren. Schlüssel muss eine <String>-Zeichenkette sein

Khaki Clover. Version 5.1, Revision 5129
Moskau, 2021

100

Wert kann Zeichenkette, Zahl oder Daten sein <Zeichenkette>, <Ganzzahl>, <Daten>

<key>ForceHPET</key>

<true/>

Es stellt sich heraus, dass es immer noch Computer gibt, bei denen der HPET standardmäßig deaktiviert ist, und im BIOS gibt es kein Kontrollkästchen, um es zu aktivieren. Dieser Schlüssel (Revision 2789+) hilft

<key>SetIntelBacklight</key>

<false/>

Der Schlüssel wurde in Revision 3298 eingeführt. In früheren Systemen wurde die Bildschirmhelligkeit durch einen speziellen Kexts IntelBacklight oder ACPIBacklight gesteuert, sie funktionieren nicht im Captain, aber es stellte sich heraus, dass man es sehr einfach in Clover beim Systemstart machen kann, und es ist kein zusätzliche Kext erforderlich, setzen Sie einfach <true/>. Ein Trick von Ramalama und Rehabman. Und hier sind zusätzliche Keys:

<key>SetIntelMaxBacklight</key>

<true/>

Der Wert aus dem folgenden Key wird in das Chipregister geschrieben:

<key>IntelMaxValue</key>

<integer>1808</integer>

oder den für diesen Chip am besten geeigneten Standardwert. Sherlock hat Clover über gesetzte Werte für fast alle Intel-Chips informiert, basierend auf der Analyse von Dumps von echten Chips. Ob das immer richtig ist, weiß ich nicht, aber diese Standardwerte schlägt Clover vor.

<key>DisableFunctions</key>

<string>0x18F6</string>

Dies ist eine Maske, die in RCBA 0x3418 überlagert wird - legt zusätzliche Bits frei, Verbot bestimmter Geräte im Intel-Chipsatz. Nur für sehr ernsthafte Hacker.

<key>LANInjection</key>

<false/>

Standardmäßig wird die eingebaute Eigenschaft für die network card (NIC) injiziert. Sie können diese Parameter verwenden, um um diese Art der Injektion zu verbieten.

RtVariables

Die folgenden beiden Parameter wurden ab Revision 980 eingeführt und sollen die Registrierung im iMessage-Dienst ermöglichen.

Ab Revision 1129 werden die Parameter aus SMBIOS übernommen und werden hier nicht benötigt.

MLB = BoardSerialNumber

ROM = Letzte Ziffern der SmUUID oder Mac-Adresse.

<key>MLB</key>

<string>XXXXXXXXXX</string>

Ziffern und Buchstaben, 17 Zeichen lang, die die Seriennummer des Motherboards angeben. Es gibt kein Muster. Der zuverlässigste Weg ist, die reale Zahl zu nehmen und die mittleren Ziffern zu ändern, z.B. schreiben Sie ...SLICE... Was immer Sie sich vorstellen.

Sechs Paare von 16 Ziffern, die oft mit der Mac-Adresse der Netzwerkkarte übereinstimmen. Es gibt jedoch, Meldungen, dass der Dienst mit beliebigen Ziffern arbeitet. Ab der Revision 3051 ist es möglich, daß sie schreiben `<string>UseMacAddr0</string>` und Clover wird die Mac-Adresse Ihrer Netzwerk-Karte ermitteln. Das Verfahren funktioniert nicht bei jedem, also prüfen sie es. Und das Wichtigste: Wenn Sie sich für iMessage anmelden, handelt es sich um einen kostenpflichtigen Dienst, sie müssen Ihre echtes Bankkonto-Karte angeben, von der Sie mit einem Dollar belastet werden. Wer versucht es umsonst zu nutzen, erhält die Meldungen wie "Rufen Sie Apple an".

Im Jahr 2015 kam ElCapitan 10.11 mit neuen Sicherheitsanforderungen heraus. SIP = System Integrity Protection (Schutz der Systemintegrität). Standardmäßig ist der Schutz aktiviert und erlaubt es Ihnen nicht, Ihre Kexte und ihre eigenen Systemdienstprogramme zu installieren. Um sie zu deaktivieren, gibt es in Clover die Möglichkeit, neue Parameter im NVRAM zu setzen

`<key>CsrActiveConfig</key>`

`<string>0x3E7</string>`

`<key>BooterConfig</key>`

`<string>0x28</string>`

Dies sind Bitmasken mit möglichen Bitwerten

```
/* Rootless configuration flags */
#define CSR_ALLOW_UNTRUSTED_KEXTS           (1 << 0)
#define CSR_ALLOW_UNRESTRICTED_FS         (1 << 1)
#define CSR_ALLOW_TASK_FOR_PID            (1 << 2)
#define CSR_ALLOW_KERNEL_DEBUGGER        (1 << 3)
#define CSR_ALLOW_APPLE_INTERNAL          (1 << 4)
#define CSR_ALLOW_DESTRUCTIVE_DTRACE      (1 << 5) /* name      deprecated */
#define CSR_ALLOW_UNRESTRICTED_DTRACE     (1 << 5)
#define CSR_ALLOW_UNRESTRICTED_NVRAM      (1 << 6)
#define CSR_ALLOW_DEVICE_CONFIGURATION    (1 << 7)
#define CSR_ALLOW_ANY_RECOVERY_OS        (1 << 8)
#define CSR_ALLOW_UNAPPROVED_KEXTS       (1 << 9)
#define CSR_ALLOW_EXECUTABLE_POLICY_OVERRIDE (1 << 10)
```

/* Bitfields für boot_args->flags */

```
#define kBootArgsFlagRebootOnPanic        (1 << 0)
#define kBootArgsFlagHiDPI                (1 << 1)
#define kBootArgsFlagBlack                (1 << 2)
#define kBootArgsFlagCsrActiveConfig      (1 << 3)
#define kBootArgsFlagCsrPendingConfig     (1 << 4)
#define kBootArgsFlagCsrBoot              (1 << 5)
#define kBootArgsFlagBlackBg              (1 << 6)
#define kBootArgsFlagLoginUI              (1 << 7)
```

Die Standardwerte entsprechen der Deaktivierung des Schutzes. Wenn sie aufgrund einer gewissen Paranoia den Schutz aktivieren wollen, setzen sie ihn auf 0 (Null).

Die folgenden drei Parameter sind ausgenommen, da sie vom System aus der Steuerung eingestellt werden müssen Clover-Panels, so dass es keinen Konflikt gibt.

```
<key>MountEFI</key>
<string>Ja</string>.
```

Dieser Parameter teilt dem Startskript mit, dass es ESP (EFI System Partition) Partition. Diese Option ist für die meisten Menschen unnötig oder temporär, sollten Sie Nein in die Konfig schreiben und ggf. Ja in das

Menü eintragen. Ein weiterer möglicher Wert ist disk1, wenn Sie mehrere Festplatten haben und jede Festplatte ihre eigene Partition hat EFI.

```
<key>LogEveryBoot</key>
```

Khaki Klee. Version 5.1, Revision 5129
Moskau, 2021

102

```
<string>Ja</string>.
```

Das Download-Protokoll ist für Entwickler erforderlich, und normale Benutzer können setzen und Nein. Hier anstelle von Ja kann eine Zahl sein, wie viele Protokolle im System gehalten werden sollen.

```
<key>LogLineCount</key>
```

```
<string>3000</string>
```

die Anzahl der Zeilen in diesem Protokoll, dann werden die alten Zeilen durch die neuen ersetzt, so dass es keine unbegrenztes Wachstum dieser Datei.

Es ist nicht wirklich interessant, diese Protokolle zu akkumulieren. Sie können das letzte Protokoll immer mit dem Befehl:

```
$ bdmessg > ~/Desktop/boot-log.txt erhalten.
```

DisableDrivers

```
<key>DisableDrivers</key>
```

```
<array>
```

```
<string>CsmVideoDxe</string>
```

```
<string>VBoxExt4</string>
```

```
</array>
```

Der Sinn dieses Abschnitts ist es, verschiedene config.plist in verschiedenen OEM-Ordern zu haben, aber da der Treiberordner wird gemeinsam genutzt, müssen sie irgendwie unterscheiden, welcher Satz von Treibern auf einer bestimmten eine andere Konfiguration nutzt. Die eine braucht z. B. OsxAptioFixDxe, die andere braucht EmuVariableDxe. **Veraltet! Ein gemeinsamer Satz wird nun auf allen Boards verwendet.**

Quirks (Macken?)

Macken

Die Geschichte dieses Abschnitts ist wie folgt. Am Anfang der Entwicklung des Clover für UEFI-Boot durch Dmazar stand die Entwicklung eines Treibers zur Anpassung des Speichers, den das UEFI-BIOS Aptio (American Megatrend) hinterlässt. Die Sache ist die, dass die Allocate-Funktion in diesem BIOS den Speicher unten alloziert, und Sie müssen den unteren Speicher frei haben, um macOS zu starten. Der Konflikt betrifft nicht nur den Speicher, boot.efi führt dort auch die Adressvirtualisierung durch, und das wirkt sich auf Zeiger, Funktionen usw. aus. Ich werde nicht ins Detail gehen, es ist nicht meine Arbeit, es war Dmazar, der Schritt für Schritt alle Konflikte gefunden und herausgefunden hat, wie man sie lösen kann. Es wurde der Treiber OsxAptioFixDrv.efi. Es gibt 32-Bit- und 64-Bit-Varianten mit allen Unterschieden in der Adressierung.

Beachten Sie, dass dieses Problem bei Legacy Clover nicht auftrat, da in diesem Fall Allocate von EDK2 verwendet wird, und dieses alloziert den Speicher von oben nach unten. Legacy Clover arbeitet ohne diesen Treiber.

Für eine lange Zeit, nachdem Dmazar gegangen war, hat niemand diesen Treiber angefasst, außer vielleicht Einzeilerergänzungen wie Free2000. Und nun hat vit9696 sich vorgenommen, den Treiber gründlich zu überarbeiten. Zunächst nahm er eine Änderung vor, um natives NVRAM auf vielen Chipsätzen (BIOSen) zu ermöglichen, die vorher nicht funktionierten. Und dann hat er den Treiber in sinnvolle Teile (Quirks) zerlegt,

die nun bei Verwendung des OpenCore-Laders beliebig ein- und ausgeschaltet werden können. Aber dann kam der Programmierer von ReddestDream und beschloss, all diese Entwicklungen von OpenCore in einen separaten Treiber OcQuirks.efi zu isolieren, der mit dem OpenRuntime.efi-Treiber zusammenarbeitet, und alle Einstellungen werden in die Datei OcQuirks.plist geschrieben, um all dies mit dem Bootloader Clover zu verwenden.

Als nächstes steht mein Umzug an. Ich muss alle Quellen in einem Repo haben, damit ich eine Zweiteilung vornehmen kann. Und da die Lizenz für all das erlaubt, den Quellcode nach Belieben zu verwenden, und alles historisch in seine historische Heimat zurückgeführt wird, habe ich sie kopiert und so aktualisiert, dass anstelle der separaten .plist-Datei die gleiche Clover config.plist verwendet wird, und diese Einstellungen nur von der GUI Clover geändert werden können.



Das heißt, wenn Sie nicht sofort booten können, können Sie versuchen, zu diesem Menü zu gehen und um eine Art von Ja-Nein-Einstellung zu ändern. Nun die Details zu jedem Element. Es werden die Standardwerte angegeben.

```
<key>AvoidRuntimeDefrag</key>
<true/>
```

Verhindert die Speicherdefragmentierung für Runtime-Dienste wie die NVRAM-Unterstützung. Empfohlen für alle außer Apple und VMware.

```
<key>DevirtualiseMmio</key>
<false/>
```

Entfernt das Laufzeitattribut aus einigen bekannten MMIO-Bereichen. Nicht empfohlen für Systeme älter als Sandy Bridge. Die Liste der bekannten Regionen kann im Bereich MmioWhitelist Key noch ergänzt werden. Nach meinen Beobachtungen ist sie jedoch immer deaktiviert. Wird angeblich für den Z390-Chipsatz benötigt.

```
<key>MmioWhitelist</key>
<array/>
```

Die Liste der Regionen ist als ein Array von Dictionaries definiert

```
<dict>
  <key>Comment</key>
  It's such a region.
  <key>Address</key>
  <string>0xffe00000</string>
  <key>Enabled</key>
  <true/>
</dict>
```

```
<key>DisableSingleUser</key>
<false/>
```

Verbietet die Verwendung des Befehlszeilenmodus, da dieser nicht sicher ist. ;)

```
<key>DisableVariableWrite</key>
<false/>
```

Verbietet den Schreibzugriff von MacOS auf NVRAM, zur Sicherheit.

```
<key>DiscardHibernateMap</key>
<false/>
```

Verwenden Sie die alte Speicherkarte beim Aufwachen aus dem Ruhezustand. Verwendung, nur wenn Sie sicher sind, dass es bei ihnen der Fall ist. Es ist fast immer nicht der Fall.

```
<key>EnableSafeModeSlide</key>
<true/>
```

Standardmäßig ergibt das Booten im abgesicherten Modus slide=0. Dieser Patch ermöglicht einen anderen Wert, der in der Auswahl ProvideCustomSlide angegeben ist.

```
<key>ProvideCustomSlide</key>
<false/>
```

Ob die automatische Berechnung des slide=***-Wertes verwendet werden soll oder nicht. Der Bedarf ist aus dem Protokoll ersichtlich, wenn es eine OCABC-Meldung gibt: Nur N/256 Slide-Werte sind verwendbar!

```
<key>ProvideMaxSlide</key>
<integer>0</integer>
```

Wert von 1 bis 254, für den obigen Fall. Der Wert wird auf 255 gesetzt durch den Standardwerte, und dies kann zu Fehlern bei der Speicherzuweisung führen.

```
<key>EnableWriteUnprotector</key>
<true/>
```

Entfernt das Schreibschutz-Bit auf der Runtime services page . Aufgrund der Unsicherheit dieses Pick gibt es eine weitere RebuildAppleMemoryMap. Das heißt, für Systeme, die älter als 2018 sind, setzen wir:

```
EnableWriteUnprotector - True
RebuildAppleMemoryMap - False
SyncRuntimePermissions - False
Und bei neueren Geräten, die MATS unterstützen, wenn es eine solche ACPI-Tabelle im BIOS gibt
EnableWriteUnprotector - False
RebuildAppleMemoryMap - True
SyncRuntimePermissions - True
```

```
<key>ForceExitBootServices</key>
<false/>
```

Booten sie mit ExitBootServices bei neuen Speicherkarte. Dies ist der Punkt, an dem Kernel- und Kext-Patches auftreten. Verwenden sie es nur, wenn Sie sicher sind, was Sie tun.

```
<key>ProtectMemoryRegions</key>
<false/>
```

Ändert Attribute einiger Speicher-Regionen, wenn es einen Konflikt zwischen den Konzepten von MacOS und dem BIOS gibt. Dieser Key enthält mehrere Korrekturen, darunter eine von mir persönlich entwickelte, eine, die den CSM-Bereich schützt. Insbesondere, wenn Ihre Grafikkarte nicht über UEFI VBIOS verfügt, sollte dieser Pick aktiviert werden. (Im OsxAptioFix3Drv-Treiber ist es dagegen definitiv eingeschaltet und hat noch nie Beschwerden verursacht. Solange sie jedoch **durch Standard.?**)

<key>ProtectSecureBoot</key>

<false/>

Schützt die Verschlüsselungs-Schlüssel vor dem Überschreiben durch das Betriebssystem. Sichere Technik Boot, das wir irgendwie nicht besonders brauchen. Die Notwendigkeit dieses Picks wird beobachtet an einige Insyde BIOSe. Die anderen brauchen es nicht.

<key>ProtectUefiServices</key>

<false/>

Einige BIOSe wie VMware versuchen, Zeiger auf Runtime-Dienste umzuschreiben. Wir brauchen sie nicht, also schützen wir sie. Jemand namens Rediskin behauptet, es sei für Z390-Chipsatz notwendig.

<key>ProvideConsoleGopEnable</key>

<true/>

Erzeugt ein GOP-Protokoll für den Konsolenmodus, d. h. für Textausgaben, die nicht in Text Modus, wie Sie es aus dem PC-BIOS gewohnt sind, sondern im grafischen Modus, wie es Apple macht. (diese Auswahl ist unbedingt erforderlich) (in der Version 5129 entfernt)

<key>RebuildAppleMemoryMap</key>

<false/>

Erzeugen Sie eine macOS-kompatible Speicherkarte. Die Sache ist die, dass Apple eine etwas andere Vorstellung davon hat, wie und was zu tun ist, als unser UEFI-BIOS. Das Problem ist jedoch, dass unsere die Speicherkarte zu unserer Hardware passt. Deshalb schalten wir die Auswahl ab. Die anderen Keys erledigen den notwendigen Teil dieser Arbeit. Dieser Key scheint EnableWriteUnprotector für Systeme zu ersetzen, die MAT unterstützen.

Außerdem erfordert diese Auswahl die Aktivierung von SyncRuntimePermissions. Sie ist jedoch standardmäßig aktiviert. Siehe EnableWriteUnprotector Key.

<key>SetupVirtualMap</key>

<true/>

Diese Einstellung befasst sich mit der Reihenfolge, in der virtuelle Adressen zugewiesen und verwendet werden. Einige BIOSe, wie z. B. OVMF, unterstützen diesen Key nicht. Hmmm, warum sollte das OcQuirks.efi-Treiber auf einem System mit OVMF?! Er funktioniert auch ohne ihn, genau wie der alte Clover. Schalten wir es also ein.

<key>SignalAppleOS</key>

<false/>

Sagt dem BIOS, dass wir ein MacOS-System laden, obwohl wir Windows laden. Es muss bei einigen Macbooks gemacht werden, aber nicht wir. Wird vielleicht von OpenCore benötigt, aber nicht von Clover.

<key>SyncRuntimePermissions</key>

<true/>

Aktualisiert die Berechtigungsflags im Bereich Runtime. Natürlich brauchen sie das.

Die allgemeine Situation ist, dass für meine nicht mehr ganz so neuen Computer diese speziellen Einstellungen die Vorgaben ausreichend waren. Dass Rediskin eine andere Liste hat, und er behauptet, dass diese Liste entspricht dem Verhalten von AptioMemoryFix. Dies ist jedoch nicht der Fall mit seinem Set. Mein Computer bootet nicht, während mit dem alten AptioMemoryFix alles in Ordnung ist. Also der Standardsatz von Clover unterscheidet sich von den ursprünglichen OcQuirks (RIP). In Clover 5125 sind durch die Aufnahme von Patches aus OpenCore in diesem Abschnitt neue Werte vorhanden.

<key>FuzzyMatch</key>

<false/>

Der Schlüssel gewährleistet die Kompatibilität mit System 10.6 (Snow Leopard), da die Menge des Cache unterschiedlich berechnet wird. Ich weiß nicht, wie es mit OpenCore aussieht, aber Clover wird schon seit langem Schneeleo geladen, zusammen mit dem gleichen Cache.

<key>KernelCache</key>

<string>Auto</string>

Mögliche Werte (Auto, Cacheless, Mkext, Prelinked). Diese Cache-Typen waren vor dem System 10.7 aktuell und wurden üblicherweise mit dem Boot-Argument Kernel=, Mkext= und anderen ausgewählt, die es im alten Clover gab.

<key>AppleXcpmExtraMsrs</key>

<false/>

XCPM-Unterstützung auf redundanten CPUs, etwas, was der KernelXCPM-Patch früher getan hat.

<key>AppleXcpmForceBoost</key>

<false/>

Dieser Patch schreibt 0xFF00 in das Register MSR_IA32_PERF_CONTROL = 0x199, d. h. statt normaler p-Zustände bietet OpenCore an, auf den maximalen Schritt zu verbiegen.

<key>DisableIoMapper</key>

<true/>

Deaktiviert die VT-d-Technologie auf eigene Art und Weise. Es gibt eine Alternative, um die DMAR-Tabelle zu verwerfen. Und dann gibt es noch die Option dart=0.

<key>DisableLinkeditJettison</key>

<true/>

Erlaubt Lilu.kext, schneller zu laufen. Auf Kosten dessen, was unbekannt ist. Und wie keepsyms=1 nicht mehr erforderlich.

<key>DisableRtcChecksum</key>

<false/>

Der Sinn des Patches ist, dass die RTC im Mac anders verwendet wird als im PC, und die Prüfsumme wird schlecht. Wir blockieren einfach seine Zählung. Es gibt ein AppleRTC=YES-Analogon in Clover, und wir tun es.

<key>DummyPowerManagement</key>

<false/>

Blockiert AppleIntelCpuPowerManagement als Alternative zum NullCPUPM Kext. Zur Erinnerung, dieser Kext wird meistens blockiert, wenn 0xE2 gesperrt ist, und es gibt andere Patches dagegen.

<key>ExternalDiskIcons</key>

<false/>

Der altbekannte Aufnäher gegen gelbe Icons ist als Spitzhacke ausgeführt.

<key>IncreasePciBarSize</key>

<false/>

Erhöht den Speicherwert für die Leiste von 1 GB auf 4 GB im Gütesiegel IOPCIFamily. Ich weiß nicht, wer sich diesen Patch ausgedacht hat und für welche Fälle. OpenCore rät davon ab, ihn überhaupt zu verwenden.

<key>PowerTimeoutKernelPanic</key>

<false/>

Dieser Patch, der nur für Catalina ist, verhindert, dass der Kernel in Panik gerät, wenn ein Gerät sehr lange braucht um zu reagieren. Warum stornieren Sie es?! Lassen Sie es in Panik geraten, anstatt stumm zu schweben. Wenn Sie jemals jemals auf eine solche Situation stoßen, versuchen Sie diesen Patch.

```
<key>ThirdPartyDrives</key>
<false/>
```

Ein seit langem bekannter Patch ist in der Clover-Konfiguration zu finden: Enable Trim on Non-Apple. Es geht darum, das Wort Apple in einer Leerzeile zu ersetzen. Es gibt auch einen Kommentar, dass Catalina intern eine Gelegenheit bietet, das Gleiche zu tun.

```
<key>XhciPortLimit</key>
<true/>
```

Es ist ein bekanntes Problem, dass Apple die Anzahl der XHCI-, USB2+USB3-Controller-Ports begrenzt auf die Anzahl von 15, und obwohl die neuen Chipsätze mehr Anschlüsse bieten, hat es Apple nicht eilig, diesen Grenzwert erhöhen. Binäre Patches gibt es schon lange, aber sie sind abhängig von der Version der Systeme. Inwieweit es OpenCore gelungen ist, die Versionsabhängigkeit zu überwinden, weiß ich nicht. Die Praxis zeigt an, dass alle nicht entfernbaren Patches entfernt werden. Ich habe mir eine Legacy_USB.kext erstellt, der die USB2-Ports entfernt, die nicht benötigt wurden, und dadurch Platz für USB3 machte. Dieser Kext funktionierte noch in 10.13, und er arbeitetauch noch in 11.1.

Acpi

Eine Gruppe von Parametern, die die Korrektur von verschiedenen ACPI-Tabellen regeln. Und es ist nicht nur so, dass der Mac seine eigenen Anforderungen hat, es sind einfach unterschiedliche Versionen des ACPI Spezifikationen und elementare Faulheit der Hersteller, und es gibt einfach im BIOS des Motherboards keine Informationen über die installierten Karten und die CPU (können Sie diese nicht dynamisch erkennen? Clover (Kleeblätter) tut es!).

Parameter für die FADT-Tabelle

```
<key>ResetAddress</key>
<string>0x64</string>

<key>ResetValue</key>
<string>0xFE</string>
```

Diese beiden Parameter dienen für einen sehr wertvollen Fix, dem Neustartfix. Diese Werte sollten in der FADT-Tabelle stehen, aber aus irgendeinem Grund sind sie nicht immer da, Die Tabelle selbst ist vielleicht kürzer als nötig, so viel kürzer, dass diese Werte verworfen werden. Standardmäßig geht der bereits in der FACP vorhandene Wert, wenn jedoch nichts, dann wird das Paar 0x64/0xFE verwendet, was einen Neustart über den PS2-Controller bedeutet. Die Praxis hat gezeigt, dass dies nicht bei jedem funktioniert, ein anderes mögliches Wertepaar ist 0x0CF9/0x06, was einen Neustart über den PCI-Bus bedeutet. Dieses Paar wird auch auf dem Original Mac verwendet, aber nicht immer funktioniert es auf Hackintoshes. Der Unterschied ist klar, es gibt auch einen PS2-Controller auf Hackintoshes, der einen Neustart verhindern kann, wenn er nicht zurückgesetzt wird. Eine andere Möglichkeit ist 0x92/0x01, ich weiß es nicht, vielleicht hilft es ja jemandem.

```
<key>AutoMerge</key>
<false/>
```

Kombiniert alle DSDT- und SSDT-Änderungen aus EFI/CLOVER/ACPI/patched mit den vorhandenen ACPI-Dateien.

Wenn sie auf true gesetzt ist, ändert sie die Art und Weise, wie die Dateien in ACPI/patched behandelt werden. Anstatt solche Dateien am Ende der XSDT hinzuzufügen (z. B. indem sie als zusätzliche Tabelle/SSDT behandelt werden), wird, wenn signature, index und OemTableId mit einer vorhandenen OEM-Tabelle übereinstimmen, diese Tabelle ersetzt. Mit dieser Funktion können Sie, wie bei DSDT, einzelne SSDTs (oder andere Tabellen) korrigieren, indem Sie die korrigierte Datei einfach in ACPI/patched ablegen. Sie müssen sich nicht mit DropOem oder DropTables herumschlagen. Und die ursprüngliche Reihenfolge bleibt erhalten. Das Mapping für SSDT basiert auf der Namensgebung, wobei die vom F4-Extraktor im Ladermenü verwendete Namenskonvention zur Identifizierung der SSDT-Position in XSDT verwendet wird. Wenn Ihr ACPI/Origin z. B. SSDT- 6-SaSsd.aml hatte und Sie dies korrigieren wollten, konnten Sie die Datei einfach nach Bedarf korrigieren und sie in ACPI/patched platzieren. Dasselbe gilt, wenn Sie es in ACPI/patched als SSDT-6.aml ablegen. Da einige OEM ACPI-Sets keinen eindeutigen Text im Feld OEM table-id verwenden, verwendet Clover sowohl die OEM table-id als auch die Nummer, die Teil des Dateinamens ist, um das Original in XDST zu finden. Wenn Sie sich an die in ACPI/origin vorgesehenen Namen halten, ist alles in Ordnung. Rehabman in den Revisionen 4265-4346 hinzugefügt

```
<key>HaltEnabler</key>  
<true/>
```

Und dies ist ein Fix, der das Shutdown/Sleep-Problem beim UEFI-Boot löst. Der Fix wird einmalig vor dem Aufruf von boot.efi durchgeführt, ist also nicht 100%ig effektiv garantiert. Trotzdem ist es, zumindest auf Intel-Chipsätzen, recht sicher. Für mich persönlich ist es ein Lebensretter!

```
<key>UseSystemIO</key>  
<false/>
```

Wenn auf true gesetzt, dient der SSDT-Abschnitt dazu, in den generierten _CST-Tabellen zwischen den Registern zu wählen:
Register (FFixedHW,
Register (SystemIO,

```
<key>smartUPS</key>  
<false/>
```

Eigentlich ist dieser Parameter dazu gedacht, ein Profil mit der Leistung=3 in der FADT-Tabelle vorzuschreiben. Die Logik ist wie folgt:
PM=1 - Desktop , Netzversorgung
PM=2 - Notebook, netz- oder batteriebetrieben
PM=3 - Server, versorgt durch SmartUPS, das auch MacOSX kennt.

Die Wahl zwischen 1 und 2 trifft Clover auf der Grundlage einer Analyse des Mobilitätsbits, aber es gibt auch Mobile-Parameter im Abschnitt SMBIOS. Sie könnten z. B. sagen, dass wir einen MacMini haben, und dass er mobil sein soll. Der Wert 3 wird gesetzt, wenn smartUPS=Yes ist.

MADT-Korrektur (APIC)

```
<key>PatchAPIC</key>  
<false/>
```

Auf manchen Rechnern ist es nur möglich, das System mit cpus=1 zu booten, oder mit einem speziellen gepatchten Kernel (Lapic NMI Patch). Eine einfache Analyse zeigte, dass sie die falschen MADT-Tabelle benutzen, d. h., es sind keine NMI-Sektion darin enthalten. Dieser Parameter dient dazu solche Tabellen im laufenden Betrieb anzupassen. Gegen einen gesunden Computer ist nichts einzuwenden, passieren wird. Allerdings habe ich auch keine Berichte gesehen, die irgendjemandem geholfen haben. Ja, es gibt Entwickler, die es brauchen, und die diesen Patch in neuen Versionen von Clover korrigiert haben. Es gibt einen entsprechenden Patch im Abschnitt KernelAndKextPatches, der auch dieses Problem löst, aber mit anderen Mitteln.

Andere ADC-Tabellen:

<key>DropTables</key>

```
<array>
  <dict>
    <key>Signature</key>
    <string>DMAR</string>
  </dict>
  <dict>
    <key>Signature</key>
    <string>MCFG</string>
  </dict>
  <dict>
    <key>Signature</key>
    <string>SSDT</string>
    <key>TableId</key>
    <string>CpuPm</string>
    <key>Length</key>
    <string>0x0fe1</string>
  </dict>
</array>
```

In diesem Array listen wir die Tabellen auf, die wir verwerfen wollen.

DMAR - weil Mac nicht mit der VT-d-Technologie kompatibel ist. Oder besser gesagt, es hat eine andere Tabelle.

MCFG - denn durch die Angabe eines MacBookPro oder MacMini erhalten wir eine brutale Bremsen. Es wurde aber bereits eine bessere Methode entwickelt.

<key>FixMCFG</key>

```
<true/>
```

In diesem Fall wird die Tabelle nicht verworfen, sondern korrigiert. Der Autor des Patches ist wieder vit9696. Dennoch bleibt die Methode zum Verwerfen dieser Tabelle bis auf weiteres auf Lager.

Zurück zur Geschichte von DropTables

SSDTs sind unterschiedlich, und wir geben zusätzlich TableId an, welche wir verwerfen werden, weil wir unsere SSDT-Tabellen nach den Regeln von Apple erstellen werden, und nicht Gigabyte, oder, (**Gott bewahre**), ASUS. Sie können in der Kopfzeile der Tabelle nachsehen, oder im Clovers Boot Logbuch. Hier ist zum Beispiel eine Tabelle, die nicht verworfen werden sollte. DefinitionBlock ("SSDT-0.aml", "SSDT", 1, "SataRe", "**SataTabl**", 0x00001000) Für die gespeicherten Tabellen gilt die Regel für binäre gepatchte DSDT, d.h. diese Tabellen werden ebenfalls geändert, was logisch ist. Wenn alle SSDT-Tabellen aus irgendeinem Grund die gleiche TableID haben, können Sie die Länge der Tabelle, die gelöscht werden soll angeben. Die Länge kann wie oben in Hex eingestellt werden, oder sie kann in <ganzzahlig> als Dezimalzahl angegeben werden.

<key>DisableASPM</key>

```
<false/>
```

Dies betrifft die ACPI-Einstellungen des Systems selbst, so dass Apples ASPM-Management funktioniert nicht wie wir es beabsichtigen, zum Beispiel ein nicht-nativer Chipsatz. In welchen Fällen es anzuwenden ist, und was es bewirkt, weiß ich nicht mehr.

```
<key>SSDT</key>
```

```
<key>DropOem</key>
```

```
<true/>
```

Da wir unsere SSDT-Tabellen erstellen oder dynamisch laden werden, müssen wir unnötige Überschneidungen von Interessen vermeiden. Mit dieser Option können Sie alle nativen Tabellen zu Gunsten neuer Tabellen verwerfen. Oder Sie wollen das Patchen von SSDT-Tabellen kategorisch vermeiden. Sie haben folgende Möglichkeit: Legen Sie die nativen Tabellen mit kleineren Änderungen in den Ordner EFI/OEM/xxx/ACPI/patched/ und lassen Sie die ungepatchten fallen. (igitt!) verwerfen Sie die ungepatchten Tabellen. Besser ist es jedoch, die oben erwähnte Methode des selektive Verwerfens zu verwenden.

```
<key>Generate</key>
```

```
<dict>
```

```
<key>CStates</key>
```

```
<true/>
```

```
<key>PStates</key>
```

```
<true/>
```

```
</dict>
```

Hier definieren wir, dass zwei zusätzliche Tabellen für C-Zustände und für P- erzeugt werden nach den von der Hack-Community entwickelten Regeln. Für C-Zustände, Tabelle mit den Parametern C2, C4, C6, Latenz, die im Abschnitt CPU genannt werden. Es ist auch möglich, Parameter im SSDT-Abschnitt anzugeben, was logisch ist

```
<key>EnableC7</key>
```

```
<true/>
```

```
<key>EnableC6</key>
```

```
<true/>
```

```
<key>EnableC4</key>
```

```
<false/>
```

```
<key>EnableC2 </key>
```

```
<false/>
```

```
<key>C3Latency</key>
```

```
<integer>67</integer>
```

Die Tatsache, dass diese Generierung wirksam geworden ist, wird durch das Kernelprotokoll kontrolliert. Ohne diese Methode schlug die Auswertung ACPI_SMC_PlatformPlugin::pushCPU_CSTData - _CST fehl. Ein gesondertes Wort zur C3Latency. Dieser Wert erscheint bei echten Macs, beim iMac etwa 200, beim MacPro etwa 10. Meiner Meinung nach werden iMacs durch P-Stats geregelt, der MacPro durch C-Stats. Und es hängt auch vom Chipsatz ab, ob Ihr Chipsatz adäquat auf D-State-Befehle von MacOS reagieren wird. Die einfachste Variante ist, diesen Parameter nicht zu schreiben, dann funktioniert alles wie es ist. Für P-Zustände, Tabelle zur Ergänzung des Prozessorteils mit den Methoden _PPC, _PCT und _PSS.

_PCT - Performance Control - Steuerung des speedstep control (der Fahrstufenregelung).

_PPC - Performance Present Capabilities, Diese Funktion liefert eine Zahl, die eine Frequenzgrenze bedeutet. Details unten, im Parameter PLimitDict.

_PSS - Performance Supported States - Ein Set der möglichen Prozessorzustände – P-States. Dieses Array wird aus den Prozessordaten erzeugt, die Clover bereits berechnet hat, und auch unter Berücksichtigung der Parameter des Anwenders:

<key>PLimitDict</key>

<string>1</string>

Die Essenz des Parameters ist sehr einfach - die maximale CPU-Frequenz zu begrenzen. Ein Wert von 0 ist. Betrieb bis zum Maximum, 1 - ein Schritt weniger als das Maximum, 2 - zwei Schritte. Beispiel: Core2Duo T8300 2400MHz läuft mit einer maximalen Frequenz von 2000, wenn er auf zwei Schritte eingestellt wird. Warum sollten sie das machen? Um eine Überhitzung des Laptops zu vermeiden, sind die Fähigkeiten der CPU die Kühlleistung überschreiten begrenzt. Der exakt gleiche Parameter ist in z.B. platform-lists : System/Library/Extensions/IOPlatformPluginFamily.kext/Contents/PlugIns/ACPI_SMC_PlatformPlugin.kext/Contents/Ressourcen/MacBook5_1.plist
Wir werden diese Plists weiter unten besprechen.
Bei einigen Prozessoren, z. B. Core2Quad, wird festgestellt, dass PLimitDict entgegengesetzt funktioniert , und bester Fall =1 ist. Es ist durchaus möglich, dass es sich nur um einen Fehler in der DSDT handelt. Zum Beispiel, weil sie den Darwin-Patch nicht machen wollten

<key>UnderVoltStep</key>

<string>1</string>

Optionaler Parameter zur Reduzierung der Prozessortemperatur durch Absenken der Betriebsspannung. Mögliche Werte sind 0, 1, 2, 3 ... je mehr, desto mehr kühlen wir die CPU ab, bis der Computer sich aufhängt. Hier funktioniert die Überlastungssicherung, Clover wird es Ihnen nicht erlauben, den Wert außerhalb des erlaubten Bereichs zu setzen, bzw. zu schreiben, was Sie wollen, und Nur was erlaubt ist, funktioniert. Die zulässigen Werte können aber auch zu instabiler Betrieb. Die Wirkung dieses Parameters wird tatsächlich beobachtet. Allerdings nur für Penryn-Prozessoren (https://de.wikipedia.org/wiki/Intel_Core_2#Penryn)

<key>NoDynamicExtract</key>

<false/>

Wenn dieses Flag auf true gesetzt ist, wird die dynamische SSDT-Extraktion deaktiviert, wenn F4 im Boot-Menü verwendet wird. Dynamische SSDTs werden nur selten benötigt und führen in der Regel zu Verwirrung (fälschlicherweise werden sie in ACPI/patched eingeordnet). Rehabman in Revision 4359 hinzugefügt.

<key>NoOemTableId</key>

<false/>

Wenn auf true gesetzt, wird die OEM-Tabellen-ID NICHT an das Ende des Dateinamens im ACPI-Tabellen-Dump durch Drücken von F4 in Clover in ACPI/origin angefügt.
Wenn auf false gesetzt, werden Endleerzeichen von SSDT-Namen entfernt, wenn die OEM-Tabellen-ID als Suffix hinzugefügt wird. Rehabman in den Revisionen 4265- 4346 hinzugefügt.

<key>DoubleFirstState</key>

<true/>

Es wurde festgestellt, dass es für einen erfolgreichen Speedstep notwendig ist, den ersten Zustand in der P-States-Tabelle zu duplizieren. Nach der Einführung anderer Parameter ist die Notwendigkeit dafür fraglich geworden. Es gilt auch nur für IvyBridge, der Rest hebt sich bedingungslos auf

<key>MinMultiplier</key>

<integer>7</integer>

Minimaler CPU-Multiplikator. Er selbst meldet 16 und bevorzugt den Betrieb mit Frequenz von 1600, für Speedstep sollten Sie jedoch die Tabellenwerte auf 800 oder sogar 700 stellen. Empirie.

```
<key>MaxMultiplier</key>
```

```
<integer>30</integer>
```

Eingeführt in Analogie zum Minimum, aber es scheint vergeblich zu sein. Sie sollte nicht eingefügt werden. Das ist jedoch der Fall. irgendwie die Anzahl der P-Zustände beeinflusst, so dass Sie experimentieren können, jedoch ohne, daß es dafür keine besondere Notwendigkeit gibt, das zu tun.

```
<key>Generate</key>
```

```
</dict>
```

```
<key>C-Staaten</key>
```

```
<true/>
```

```
<key>PluginTyp</key>
```

```
<false/>
```

```
<key>APLF</key>
```

```
<false/>
```

```
<key>APSN</key>
```

```
<false/>
```

```
<key>PStaaten</key>
```

```
<true/>
```

```
</dict>
```

Im neuen Clover ist diese Gruppe von Parametern in einem Abschnitt zusammengefasst, und PluginType ist nun nur wahr oder falsch, weil es keine anderen Möglichkeiten gibt. APLF- und APSN-Parameter beeinflussen den Speedstep, aber für diejenigen, die wissen, was sie machen. Hinweis: Da APSN/APLF Teil von Generate → PStates sind, gelten sie, wenn Generate → PStates=true, während PluginType unabhängig ist und unabhängig von der Wahl von Generate → PStates gilt.

```
<key>PluginType</key>
```

```
<integer>0</integer>
```

Für IvyBridge-, Haswell- (und höhere?) Prozessoren sollten Sie 1 eintragen, für den Rest 0. Dieser Schlüssel erzeugt zusammen mit dem Schlüssel Generate → PluginType die SSDT-Tabelle, enthält nur PluginType, aber keine P-States, wenn deren Erzeugung deaktiviert ist.

Dieser Schlüssel wird nicht benötigt, er wurde aus Gründen der Abwärtskompatibilität gespeichert.

Großer Abschnitt über DSDT-Setup und Patches.

```
<key>DSDT</key>
```

```
<dict>
```

```
<key>Debug</key>
```

```
<false/>
```

Mit diesem Parameter können wir sehen, was mit dem DSDT passiert, während es gepatcht wird, wenn dadurch das System am Hochfahren gehindert wird. Zuerst wird die Originalversion gespeichert /EFI/CLOVER/ACPI/origin/DSDT-or.acpi, dann gehen Sie die Prozedur aller Patches durch (Übrigens es hinterlässt eine ganze Reihe von Meldungen in debug.log, wenn es auch verbunden ist), und dann speichert es die Datei /EFI/CLOVER/ACPI/origin/DSDT-pa0.acpi, wenn eine solche Datei bereits existiert mit des vorherigen Versuchs wird die nächstnumerierte DSDT-pa1.acpi erstellt, DSDT- pa2.acpi... , überschreiben sie sich nicht gegenseitig. Denken Sie am Ende aller Übungen daran den Ordner zu leeren.

```
<key>Name</key>
```

```
<string>DSDT.acpi</string>
```

```
<key>Debug</key>
```

```
<false/>
```

Mit diesem Parameter können Sie sehen, was mit DSDT passiert, wenn es gepatcht wird und wir das System danach nicht booten können. Zuerst wird die originale /EFI/CLOVER/ACPI/origin/DSDT-or.acpi gespeichert, dann wird die Prozedur für alle Patches durchlaufen (was übrigens ziemlich viele Meldungen in der debug.log hinterlässt, wenn sie auch gemountet ist), und dann wird die /EFI/CLOVER/ACPI/origin/DSDT-pa0.acpi gespeichert, wenn schon eine vom vorherigen Versuch existiert, dann wird eine DSDT-pa1.acpi erstellt, DSDT- pa2.acpi... Sie überschreiben sich nicht gegenseitig. Vergessen Sie nicht, den Ordner am Ende aller Übungen aufzuräumen.

Sie können mehrere Versionen der DSDT-Datei mit beliebigem Namen haben, und Sie können einen nicht existierenden Namen schreiben, z. B. Standard-BIOS.aml, und dieser wird dann als Basis genommen für die DSDT, die im BIOS verfügbar ist. Die Dateiprioritäten sind wie folgt:

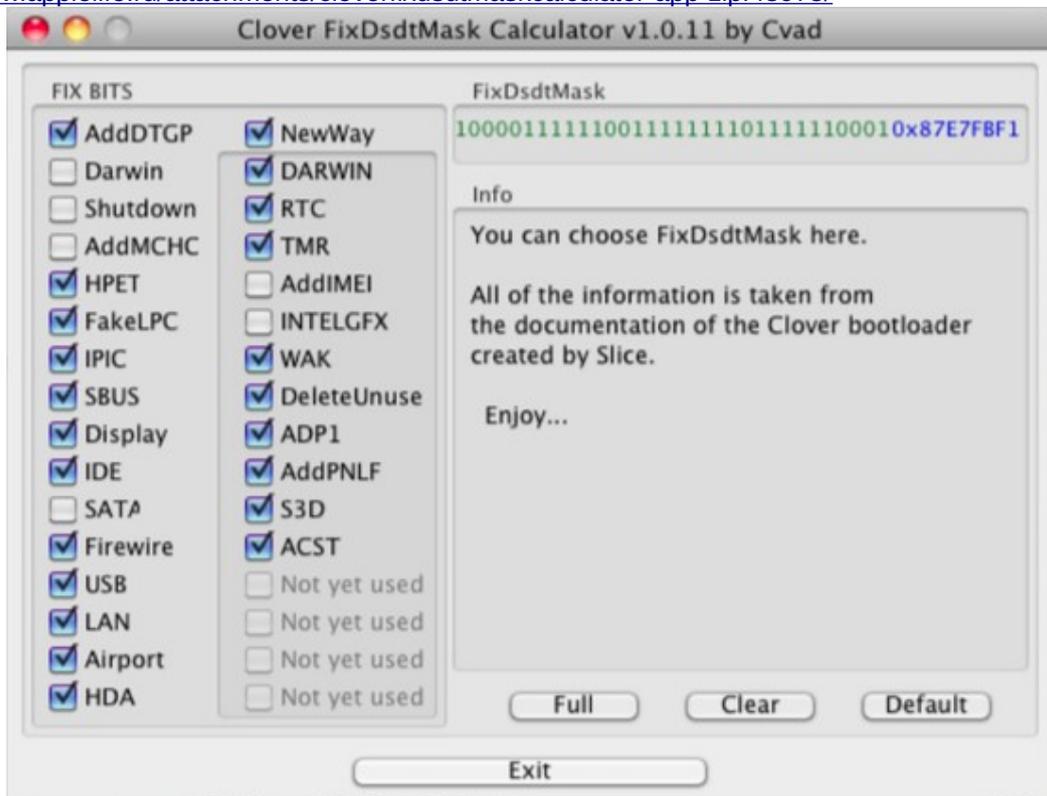
1. Die höchste Priorität hat die Datei DSDT.aml, die im Stammverzeichnis des gebooteten Systems liegt. Die Logik ist, um verschiedene Computer mit demselben Flash-Laufwerk zu booten, von denen jeder seine eigene DSDT braucht.
2. Wenn es keine solche Datei gibt, suchen Sie auf dem Flash-Laufwerk im OEM-Bereich: /EFI/CLOVER/OEM/p8b/ACPI/patched/DSDT.aml
3. Wenn es auch dort nicht zu finden ist, schauen Sie in den gemeinsamen Ordner /EFI/CLOVER/ACPI/patched/DSDT.aml

```
<key>FixMask</key>  
<string>0xFFFFFFFF</string>
```

Dieser Parameter blendet 32 Patches auf einmal für die DSDT-Tabelle aus, und zwar um die Anzahl der Bits in die Maske.

Um zu berechnen, wie sich die Summe der Bits zu einer bestimmten Maske addiert, können sie den Systemrechner nehmen, stellen sie ihn in die Programmiereransicht und schalten Sie auf 16 Zahlen. Und nun geben wir durch Anklicken der Bits 0 bis 31 die gewünschte Maske ein. Es gibt eine offensichtlichere Variante: CloverFixDsdMaskCalculator von Cvad

<http://www.applelife.ru/attachments/cloverfixdsdtmaskcalculator-app-zip.43973/>



Ab der Revision 2184 können (und sollten) Patches nach und nach in den folgenden Abschnitt eingefügt werden

```
<key>Fixes</key>  
<dict>  
  <key>AddDTGP</key>  
  <true/>  
  <key>FixDarwin</key>
```

```
<true/>
<key>FixShutdown</key>
<true/>
<key>AddMCHC</key>
<false/>
<key>FixHPET</key>
<true/>
<key>FakeLPC</key>
<false/>
<key>FixIPIC</key>
<true/>
<key>FixSBUS</key>
<true/>
<key>FixDisplay</key>
<true/>
<key>FixIDE</key>
<false/>
<key>FixSATA</key>
<false/>
<key>FixFirewire</key>
<true/>
<key>FixUSB</key>
<true/>
<key>FixLAN</key>
<true/>
<key>FixAirport</key>
<true/>
<key>FixHDA</key>
<true/>
<key>FixDarwin7</key>
<true/>
<key>FixRTC</key>
<true/>
<key>FixTMR</key>
<true/>
<key>AddIMEI</key>
<true/>
<key>FixIntelGFX</key>
<false/>
<key>FixWAK</key>
<true/>
<key>DeleteUnused</key>
<true/>
<key>FixADP1</key>
<true/>
<key>AddPNLF</key>
<true/>
<key>FixS3D</key>
<true/>
<key>FixACST</key>
<true/>
<key>AddHDMI</key>
<true/>
<key>FixRegionen</key>
<true/>
</dict>
```

Wenn dieser Fixabschnitt vorhanden ist, wird der maskierte Fixschlüssel ignoriert. Aber um Ihnen zu sagen, was diese Korrekturen bedeuten, müssen Sie ein neues Kapitel aufschlagen.

Weitere Keys, um einige Probleme mit dem automatischen Patching zu lösen.

<key>ReuseFFFF</key>

<false/>

In einigen Fällen wird der Versuch, die Anzeige zu patchen, durch das Vorhandensein im Original DSDT-Gerät wie z. B.

```
Device (PEGP)
{
  Name (_ADR, 0xFFFFFFFF)
  Name (_SUN, Eins)
}
```

Es ist möglich, die Adresse auf 0 zu ändern, aber das funktioniert nicht immer. `<true/>` - versuchen Sie, die Adresse zu ändern, `<false/>` - gehen Sie weg und versuchen Sie nicht, sie zu patchen.

Dieser Fix ist im neuen Clover seit 5116 ausgeschlossen:

```
<key>DropOEM_DSM</key>
  <Diktat>.
<key>ATI</key>.
<true/>.
<key>NVidia</key>
<true/>.
<key>IntelGFX</key>
<true/>.
<key>HDA</key>
<true/>.
<key>HDMI</key>
<true/>.
<key>LAN</key>
<true/>.
<key>WIFI</key>
<true/>.
<key>SATA</key>
<true/>.
<key>IDE</key>.
<true/>.
<key>USB</key>
<true/>.
<key>LPC</key>
<false/>.
<key>SmBUS</key>
<false/>.
<key>Feuerdraht</key>
<true/>.
</dict>.
```

In einigen Fällen hat das Gerät, das wir automatisch patchen wollen, bereits OEM-Methode `_DSM`. Sie können nicht duplizieren, daher gibt es zwei Möglichkeiten:
`<true/>` - die ursprüngliche Methode wird verworfen und stattdessen wird unsere Methode erzeugt,
`<false/>` - beim Auftreten der ursprünglichen Methode ziehen wir uns zurück, ohne etwas zu tun. Was ist an der ursprünglichen Methode dran? Kaum das, was wir gerne sehen würden, und kaum das, was wird für OSX benötigt. Normalerweise denken BIOS-Hersteller nur an Windows. Aber wenn Sie denken, dass es etwas Wichtiges in dieser Methode gibt, dann injizieren Sie Ihre Eigenschaften in dieses Gerät mit Strings (siehe Kapitel Geräte->Injizieren). Ich würde also empfehlen, alle diese OEM-DSMs zu verwerfen, außer wenn Sie setzen Sie Ihr benutzerdefiniertes DSDT, und wenden Sie weitere Korrekturen aus dem Bereich automatisch, aber Sie sind nicht bereit, Ihre `_DSM`-Methoden für automatisch zu ersetzen generiert.

Das DSDT meines neuen Computers erwies sich als wahnsinnig schlecht in Bezug auf macOS. Es hat vielen DSM-Methoden, sowohl in der Elternschaft als auch bei den Töchtern. Das Ergebnis ist Panik. DropOEM_DSM ist hier machtlos. Der folgende Patch hat geholfen Patches → find:_DSM, ersetzen: ZDSM. Siehe unten.

```
<key>PNLF_UID</key>
```

```
<string>0x0A</string>
```

Es gibt mehrere Beispiel-Helligkeitskurven/-grafiken im System, und sie haben unterschiedliche UUIDs, aber wenn ein echter Prozessor diese Kurve hatte, bedeutet das nicht, dass Sie die gleiche Helligkeit mit dem gleichen Prozessor haben werden. Es kommt auf das Panel an, nicht auf den Prozessor.

Es wäre besser, ein PNLF-Kalibrierungssystem zu entwickeln, aber das ist eine Spitzenleistung. Im Moment schlagen wir nur vor, mit verschiedenen Werten zu experimentieren und zu sehen, ob es besser wird. In Revision 5103 hinzugefügt.

```
<key>SuspendOverride</key>
```

```
<false/>
```

Der Shutdown-Patch funktioniert nur bei Zustandstyp 5 - Shutdown, wir möchten jedoch diesen Patch auf die Zustände 3 und 4 erweitern, dann setzen Sie SuspendOverride = true. Es half mir beim Sleep-Problem während eines UEFI-Boots. Andernfalls würde der Bildschirm erlöschen und die Beleuchtung und die Ventilatoren laufen weiter.

Fortgeschrittene Hacker können ihre eigenen DSDT-Patches erstellen, indem sie eine Ersetzung auf binärer Ebene verwenden:

```
<key>Patches</key>
```

```
<array>
```

```
<dict>
```

```
<key>Finden</key>
```

```
<data>UFhTWAhfQURSAAhfUFJXEgYC</data>
```

```
<key>Replace</key>
```

```
<data>UFhTWAhfQURSAAhfU1VOCgQIX1BSVxIGAg==</data>
```

```
<key>TgtBridge</key>
```

```
<data>UFhTW</data>
```

```
</dict>
```

```
<dict>
```

```
<key>Comment</key>
```

```
<string>Rename oem _DSM to ZDSM</string>
```

```
<key>Disabled</key>
```

```
<false/>
```

```
<key>Find</key>
```

```
<data>X0RTTQ== </data>
```

```
<key>Replace</key>
```

```
<data>WkRTTQ== </data>
```

```
</dict>
```

```
</array>
```

TgtBridge definiert, dass das Patch nur innerhalb eines einzelnen Blocks funktioniert, der durch unter diesem Namen Gerät (RP02) { . } definiert wird.

Diese Methode ist jedoch ungenau, da Blöcke fragmentiert sein können, es ist besser, sie verwenden die Methode RenameDevices (s. unten)

Die spezifischen Zahlen sind von Ihren Entwürfen, wenn Sie wissen, was zu tun ist. Wenn die Stringlängen nicht übereinstimmen, wird Clover die Längenänderung korrekt berücksichtigen, mit einer Ausnahme: Wenn es innerhalb einer If- oder Else-Anweisung geschieht. Wenn Sie eine solche Änderung wünschen, ersetzen Sie den gesamten Operator. (The length of the strings may not be the same, Clover will correctly consider the length change, with one exception: to prevent it from happening inside the If or Else statement.)

Einige Erklärungen. Die Kommentartaste dient nicht nur dazu, Sie daran zu erinnern, dass hier ein Kommentar geschrieben steht, sie wird auch im Clover-Menü verwendet, um Menüs durch connect/disable dieser Fixes zu compilieren. Den Anfangswert freigegeben oder nicht wird durch die Zeilen Disabled=false definiert. Der Standardwert ist aktiviert. Wenn Sie Patches schreiben, ist es besser, sie zuerst als Disabled=true zu schreiben, und dann im Menü ihre jeweils einen zuzulassen.

Andere ACPI tables

Clover verfügt über einen Dienst zum Laden anderer Tabellen. Insbesondere ist es sehr häufig eine ganze Bibliothek mit verschiedenen SSDT-xxx.aml für verschiedene Geräte und für Speedstep zu erstellen. Ab der Revision 3088 hat sich die Regel zum Laden solcher Tabellen geändert.

Legen Sie die Tabellen in den Ordner
/EFI/CLOVER/OEM/xxx/ACPI/patched/

Wenn es keinen solchen Ordner gibt, wird der öffentliche Ordner als
/EFI/CLOVER/ACPI/patched/ verwendet.

Alle Dateien mit der Erweiterung ".aml", die nicht mit einem Punkt "." beginnen, und nicht die Zeichenfolge "DSDT" in seinem Namen enthält, werden aus diesem Ordner geladen, weil die verschiedenen DSDTs die Optionen mit einem anderen Algorithmus geladen werden. Die Bootreihenfolge ist nicht garantiert. Wenn wir eine streng definierte Reihenfolge haben wollen beim booten, müssen wir es explizit in die Config schreiben:

```
<key>ACPI</key>
<dict>
  <key>SortedOrder</key>
  <array>
    <string>SSDT-3.aml</string>
    <string>SSDT-1.aml</string>
    <string>SSDT-2.aml</string>
  </array>
```

Und wenn ein solches Array vorhanden ist, werden nur diese Tabellen geladen, und zwar genau in dieser Reihenfolge (**lacht** Okay?).

Ein weiteres Problem bei Tabellen, ist der Name. Der OEM wird weder durch die Verwendung der nationalen Alphabet, noch das Fehlen eines Namens verboten, aber für Apple ist dies inakzeptabel. Der Name muss 4 des lateinischen Alphabets sein. Der nächste Fix wird das beheben.

<key>FixHeaders</key>

```
<true/>
```

Genau derselbe Fix war jedoch im Abschnitt DSDT Fixes, da er sich nicht auf DSDT bezieht, wird er in einem separaten Punkt des ACPI-Abschnitts abgelegt.

<key>RenameDevices</key>

```
<dict>
  <key>_SB.PCI0.RP02.PSXS</key>
  <string>ARPT</string>
  <key>_SB.PCI0.EHC1</key>
  <string>EH01</string>
  <key>_SB.PCI0.POP2.PEGP</key>
  <string>GFX0</string>
</dict>
```

Anders als ACPI-Binärpatches, die übrigens nicht nur DSDT, sondern auch SSDT als auch, sowohl nativ als auch bootfähig, dient diese Methode als Ersatz für einen Patch wie Suchen->PX SX, Ersetzen->ARPT. Wenn aber im Abschnitt DSDT->Patches eine solche Ersetzung auf in der Methode RenameDevices beruht, sucht der Algorithmus nur nach diesen Geräten, die auf der besagten Bridge liegen.

Hier ist ein komplexes Beispiel zum Ersetzen von _SB.PCI0.RP02.PSXS → ARPT Scope(_SB)

```
{
```

```

{
    Device (PCI0)
    {
        Device(RP02)
        {
            Device(PSXS) <- hier ändern
            {
                Method(_ON)
                {
                }
            }
            Method(_OFF)
            {
            }
        }
        PSXS._ON() <- hier ändern
    }
    Scope(RP02)
    {
        PSXS._OFF() <- hier ändern
    }
    Device(RP03)
    {
        Device(PSXS) <- hier nicht ändern
        {
        }
    }
    PSXS._ON() <- hier nicht ändern
}

```

DSDT-Korrektur

DSDT - Differentiated System Description Table - die größte und komplexeste ACPI Tabelle. Die Mindestlänge beträgt 36 Byte, die tatsächliche Länge beträgt 20kb oder mehr. Diese Tabelle beschreibt Geräte und deren Zugriffsmethoden. Zugriffsmethoden können arithmetische und logische Ausdrücke sein, und bilden somit ein Programm in einer Programmier-Sprache ähnlich wie in C durch die geschweiften Klammern. Um diese Tabelle zu korrigieren bedeutet, dass Sie etwas über Programmierung wissen müssen. Clover bietet eine Art von Variante der automatische Bearbeitung, aber Sie müssen verstehen, dass es in Clover noch keine künstliche Intelligenz gibt, und die automatische Programmanpassungen sind noch lange nicht perfekt. Ein Programmierer*in wird es besser machen.

Warum muss es repariert werden? Das gesamte DSDT-Patch war von Anfang an darauf ausgerichtet zuallererst das HPET-Gerät - High Precision Events Timer - zu reparieren. Der Punkt ist, dass das OSX-System das Paket AppleIntelCPUPowerManagement enthält, das zur Steuerung der Prozessorleistung (Speedstep des Prozessors) dient, und die strikt verlangt, dass dieses Gerät ein HPET mit IRQ-Interrupts war. Ohne diese Bedingung gerät der Kext in Panik.

Es ist war nur möglich, durch Deaktivieren oder Löschen dieses Kext zu arbeiten. Aber es gibt noch eine andere Möglichkeit. Korrigieren Sie die DSDT und der HPET startet richtig! Jedoch ab MacOS 10.9 haben sich Anforderungen geändert. Blödsinn, der Patch ist weiterhin erforderlich und hat die gleiche Gültigkeit. In neuen Systemen, auf der neuen Hardware hat bereits er seine Relevanz verloren, vor allem aufgrund der Tatsache, dass die Kext-Funktionalität in den Kernel übergegangen ist.

Dies ist Patch **Nr. 1**, eine lebensnotwendige Notwendigkeit. Ist MacOS das einzige, das diesen HPET benötigt? Sicher nicht, aber die BIOS-Hersteller beginnen gerade erst, dies zu erkennen und schreiben korrekten Parametern, ist es immer noch selten, dass ein DSDT ohne Patch funktioniert.

Moment **Nr. 2**. Im DSDT lassen sich einige betriebliche Abhängigkeiten erkennen Systeme, "Windows 98", "Windows 2001", "Windows 2006 (Vista)", "Linux", MacOS hat „Darwin“-Kennung, und der Mit dem Mainboard ausgelieferte DSDT ist in der Regel nicht dafür ausgelegt. Und selbst wenn es konzipiert ist, dann für eine Version wie FreeBSD. MacOSX ist ein ernstzunehmendes ACPI-System, d.h. verwendet DSDT maximal, so wie Windows 2001 es verwendet, aber nicht Linux, nicht Windows 98, und nicht Windows 2006 (Vista, auf Windows-Server 2003 aufbauend). Das Richtige ist, eine Nachahmung von Windows 2001 zu machen.

Auch wenn Sie bereits "Darwin" haben, bringen Sie es als "Windows 2001" zum Laufen. **Viele BIOSe haben OSYS = 0x07D2 als Wert, aber nicht 0x2410, da es im Attribut angegeben ist.** Obwohl es Berichte gibt, dass 7D6 oder 7D9 für einige Konfigurationen irgendwie besser sind. Sie müssen sich den Algorithmus ansehen.

Auf meinem Skylake 2017 stellte sich als bester Wert 0x7DA heraus. Daher habe ich eine zusätzliche DSDT FixDarwin7_10000, der ein Windows von 2009 nachahmt, d. h. Windows 7. Ich habe es entsprechend 0x7D9, und es beeinflusst den USB 3.0-Betrieb, aus irgendeinem Grund in dem BIOS wird es für ein WindowsXP-System deaktiviert Auch hier wird das Patch möglicherweise nicht benötigt, wenn diese Kennung im DSDT nicht weiter verwendet wird. Haben Sie das überprüft?

Moment #3. Der Hersteller des Motherboards, und damit auch dessen BIOS und dessen DSDT, kann nicht vorhersehen, welche CPU installiert wird, welche Grafikkarte und andere PCI Geräte. Diese sollten aber in der DSDT vorgeschrieben werden! Und umgekehrt, sollten von der DSDT, solche Geräte wie Lautsprecher, Diskettenlaufwerk, Parallelport ausgeschlossen werden. Es gibt keine Treiber für sie und es werden keine benötigt. Außerdem ist es oft notwendig, Anschlüsse/Ports hinzuzufügen oder zu entfernen, um einige Geräte, wie z. B. eine Grafikkarte oder ein SATA-Controller nutzen zu können.

Khaki Klee. Version 5.1, Revision 5129
Moskau, 2021

118

DSDT liegt im BIOS und wird im System im AML-Binärcode verwendet. Es existiert der „IASL-Compiler/Decompiler“, der Codes in eine für Menschen verständliche Sprache übersetzt

DSL. Der menschliche Weg der Bearbeitung ist AML->DSL->Bearbeiten->DSL->AML. Und dann gibt es noch

Punkt **Nr. 4**. Die letzte Kompilierung wird aufgrund von Fehlern unmöglich, Syntax und logischen, die ursprünglich im OEM DSDT vorhanden waren. Bei der Bearbeitung ist es erforderlich, diese ebenfalls zu bearbeiten und zu korrigieren. Nun, zur gleichen Zeit, um die semantischen Fehler, die verursacht wurden, zum Beispiel zu korrigieren, und verhindern, das der Computer nicht einschlafen oder nicht aufwachen kann. Oder vielleicht gibt es auch neue Geräte, die beschrieben werden sollen.

(Und im Allgemeinen ist es seltsam, aber Kompilieren/Dekompilieren ist nicht strikt invers. Der Betrieb, hin und her ändert die Tabelle, oder auch allgemein, hin geht, zurück nicht - es ist erforderlich zu intervenieren. Aus meiner Sicht bedeutet dies, dass der Decompiler mit Bugs verwendet wurde, solche Programmierer haben daran gearbeitet. Und bei Nichteinhaltung von Normen ist notwendig, das sie als Warnungen und nicht als Fehler markiert werden. Wenn im ursprünglichen AML etwas schief geht, dann müssen Sie verstehen, dass der Computer funktioniert, also etwas falsch interpretiert werden kann).

Wenn wir all dies getan haben, können wir die korrigierte DSDT in den Bootloader verwenden, indem sie sie in den Ordner /EFI/CLOVER/OEM/xxx/ACPI/patched legen, oder, falls der OEM-Computernamen noch nicht bekannt, in den Ordner /EFI/CLOVER/ACPI/patched, oder **das bootfähige System selbst hat eine eigene Version der DSDT, die im Stammverzeichnis der Systemplatte liegt.**

Woher bekomme ich die Quell-DSDT, die gepatcht werden muss? Es gibt verschiedene Möglichkeiten, sie zu erhalten unter Windows, Linux oder sogar OSX. Wenn Clover es irgendwie geschafft hat zu laufen, bietet sich diese Möglichkeit selbst. Es ist notwendig, das grafische Menü aufzurufen und die Taste F4 zu drücken. Wenn Clover auf einer FAT32-Partition installiert ist, kann es alle OEM ACPI-Tabellen, einschließlich unangetasteter DSDT und FADT sichern.

Seien Sie geduldig, wenn Sie auf einem Flash-Laufwerk speichern und es viele Tabellen gibt, kann der Prozess eine merkliche Zeitspanne in Anspruch nehmen. In der aktuellen Revision extrahiert Clover einen Satz von Tabellen, die zuvor auf andere Weise nicht verfügbar waren, einschließlich AIDA64. Es gibt auch einen Weg, zum Speichern der vorgefertigten DSDT-Version auf der Festplatte. Geben Sie dazu in der Clover-Oberfläche in das Optionsmenü, ändern Sie die DSDT-Maske, verlassen Sie dann das Menü und drücken Sie F5. Clover wird Ihre DSDT speichern, korrigiert auf die aktuelle Maske, mit einem Namen wie DSDT-F597.aml, d.h. gepatcht mit Maske 0xF597. Sie können mehrere Varianten erstellen, um sie später zu vergleichen.

Jetzt können Sie eine DSDT-Datei nehmen und sie bearbeiten... Und für diejenigen, die der Sprache ASL nicht mächtig sind, Clover schlägt vor, einige Korrekturen automatisch durchzuführen. Auch diese Frage beantworte ich gleich.

Frage: "Warum kompiliert DSDT immer noch mit Fehlern, nachdem es von Clover bearbeitet wurde?". Ja, die DSDT ist ein Satz von Beschreibungen und Methoden, von denen wir viele nicht benötigen. Clover berührt sie nicht, auch wenn wir die maximale Maske setzen. Fehler können an diesen Stellen unberührt

abgelegt sein, und sie bleiben dort. Das hindert aber nicht daran, dass trotzdem alles funktioniert, denn die DDT funktioniert nicht als Ganzes, sondern nur als ein Satz von Beschreibungen und Methoden.

Lassen Sie uns die Korrekturen im Detail betrachten. In der Revision 4300+ wurde die Beschreibung der Fixes vereinfacht.

AddDTGP bit(0):

Zur Beschreibung von Geräteeigenschaften, mit Ausnahme der Option DeviceProperties, die oben diskutiert wurden, gibt es eine Variante mit der in DSDT vorgeschriebenen _DSM-Methode.

_DSM - Device Specific Method - eine bekannte Ausprägung dieser Methode, die in MacOSX seit Version 10.5 enthält. Diese Methode ein Array, das das Gerät beschreibt und den Aufruf universelle DTGP-Methode, die für alle Geräte gleich ist.

Diese Korrektur fügt diese Methode einfach hinzu, so dass sie dann auf andere Korrekturen angewendet werden kann. **Sie hat keinen eigenständigen Wert. Ich habe den Ratschlag gesehen, eine 0x31-Maske aufzusetzen, sie sagen, die anderen Korrekturen sind nicht erforderlich. Aber dann brauchen Sie auch (1) nicht!**

FixDarwin bit(1):

Darwin OS imitiert das Windows XP-System. Viele Schlaf- und Helligkeitsprobleme erwachsen aus einer Fehleinschätzung des Systems.

Khaki Klee. Version 5.1, Revision 5129
Moskau, 2021

119

FixDarwin7 bit(16)

Ähnlich, nur vom Windows 7 System nachgeahmt. Ältere DSDTs haben möglicherweise nicht die Möglichkeit, das für ein solches System zu testen. Sie haben Optionen.

FixShutdown bit(2):

Die Funktion _PTS wird um eine Bedingung erweitert: wenn Argument = 5 (aus) ist, sollten keine anderen Aktionen ausgeführt werden. Seltsam, warum ist das so? Dennoch gibt es immer wieder Beweis für die Wirksamkeit dieses Patches für ASUS-Boards, vielleicht auch für andere. In einigen BIOSen hat die DSDT bereits eine solche Prüfung durchlaufen, in diesem Fall sollte eine solche Korrektur deaktiviert werden, wenn sie in der Konfiguration ist. Mit SuspendOverride=true, wird diese Korrektur auf die Argumente 3 und 4 erweitert. Das heißt, das Verlassen zu unterbrechen. Andererseits, wenn HaltEnabler=true ist, ist dieser Patch wahrscheinlich nicht nötig, Ich brauche ihn aber.

AddMCHC Bit(3):

Ein solches Gerät der Klasse 0x060000 ist normalerweise in DSDTs nicht vorhanden, aber für einigen Chipsätzen ist dieses Gerät wartungsfähig und muss daher Vorschrift zur korrekten Verteilung des PCI-Bus-Power-Managements haben. Die Frage nach die Notwendigkeit eines Patches wird experimentell entschieden. Mehr Erfahrung, dieses Gerät wurde benötigt auf einem Motherboard mit einem Z77-Chipsatz, da sonst Core-Panic beim ersten Start auftritt. Umgekehrt, auf der G41M-Chipsatz (ICH7) verursacht dieser Fix Panik. Leider lässt sich keine allgemeine Regel erkennen.

FixHPET bit(4):

Wie gesagt, dies ist eine wichtige und notwendige Korrektur. Die minimal erforderliche Maske eines DSDT-Patches sieht also wie 0x0010 aus.

FakeLPC Bit(5):

Ersetzt die DeviceID des LPC-Controllers, so dass sich der AppleLPC-CheKextack an diesen anhängt. Er wird benötigt, wenn der Chipsatz nicht für OSX vorgesehen ist (z. B. ICH9). Allerdings, Die Liste der nativen Intel- und NForce-Chipsätze ist so groß, dass die Notwendigkeit eines solchen Patches sehr selten ist. Das System prüft, ob die AppleLPC Kext geladen ist, wenn nicht, wird der Patch benötigt. Allerdings ist auch dies keine Tatsache. Manchmal wird der Kext aus dem Speicher entladen, wenn er nicht benötigt wird, obwohl der Chipsatz unterstützt wird.

FixIPIC-Bit (6):

Entfernt den Interrupt vom IPIC-Gerät. Dieser Fix wirkt sich auf die Funktion der Power-Taste aus (Popup-Fenster mit den Optionen Reset, Sleep, Shutdown).

FixSBUS-Bit (7):

Fügt den SMBusController in den device tree (Gerätebaum) ein und entfernt damit die Warnung über seine Abwesenheit aus dem Systemprotokoll. Und schafft auch die richtige Leistungsteuerung, damit beeinflusst es auch den Ruhezustand.

FixDisplay bit(8):

Erzeugt eine Reihe von Videokarten-Patches. Injiziert Eigenschaften, und Geräte selbst, wenn sie nicht vorhanden sind. Injiziert FakeID, wenn bestellt. Fügt benutzerdefinierte Eigenschaften hinzu. Der gleiche Fix fügt ein HDAU-Gerät für die Audioausgabe über HDMI hinzu. Wenn FakeID gesetzt ist, wird die wird es über die _DSM-Methode injiziert. (1974) Patches sind für alle Grafikkarten, nur nicht für Intel-Grafikeinheiten (IGPU). Für eingebettete Intel-Karten ist anders Bit zuständig.

FIX_INTELGFX_100000

Bei Bedarf wird auch ein HDMI-Gerät (HDAU) hinzugefügt.

Khaki Klee. Version 5.1, Revision 5129
Moskau, 2021

120

FixIDE bit(9):

Die AppleIntelPIIXATA-Kext-Panik erschien in System 10.6.1. Zwei Lösungen Probleme - Verwendung des korrigierten Kext oder Fixierung des Geräts im DSDT. Und für von moderneren Systemen? Lassen Sie es sein, wenn es einen solchen Controller gibt.

FixSATA bit(10):

Behebt Probleme mit SATA und beseitigt das Vergilben der Festplattensymbole im System durch Nachahmung von ICH6. Eigentlich eine umstrittene Methode, aber ohne diesen Fix habe ich DVD-Discs, die nicht abspielbar sind, und ein DVD-Laufwerk sollte nicht entfernt werden können, d.h., ersetzen Sie einfach die Icons, ist das keine Option!

Es gibt eine Alternative, die durch Hinzufügen von AppleAHCIport.kext gelöst wird. Siehe das Kapitel zum Patchen der Kexts. Und dementsprechend kann dieser Teil weggelassen werden! (Eines der wenigen Stücke, **die ich empfehle es nicht?**)

FixFirewire bit(11):

Fügt dem Firewire-Controller die Eigenschaft "fwhub" hinzu, wenn Sie einen haben. Wenn nicht, nichts wird nicht passieren. Sie können es nicht beachten, wenn Sie nicht wissen, ob Sie es setzen müssen oder nicht.

FixUSB bit(12):

Der Versuch, die vielen Probleme mit USB zu lösen. Für den XHCI-Controller, wenn bei Verwendung einer nativen oder gepatchten IOUSBFamily ist dieser DSDT-Patch unverzichtbar. Der Apple-Treiber verwendet speziell ACPI, und die Schreibweise in DSDT muss korrekt sein. Es besteht kein Konflikt mit den Strings im DSDT.

FixLAN bit(13):

Injizieren die "eingebauten" Eigenschaft für den Netzwerk-karte (NIC) - notwendig für korrekte Bedienung. Das Kartenmodell wird ebenfalls injected - für die Kosmetik.

FixAirport bit(14):

Ähnlich wie beim LAN wird zusätzlich das Gerät selbst angelegt, wenn es nicht bereits in DSDT vorhanden ist. Bei einigen bekannten Modellen wird die DeviceID ersetzt durch unterstützte. Und der airport (Flughafen) wird ohne weitere Patches aktiviert.

FixHDA bit(15):

Korrigieren der Soundkartenbeschreibung in der DSDT, damit der native Treiber AppleHDA funktioniert. AZAL -> HDEF wird umbenannt, layout-id wird injiziert und Pin-Konfiguration.

FixMutex:

Dieser Patch findet alle Mutex-Objekte und ersetzt den SyncLevel durch 0. Wir verwenden diesen Patch, weil OS X kein richtiges Debugging von Mutex und Interrupts bei jedem Acquire mit einer Mutex, die einen SyncLevel ungleich Null hat. SyncLevel-Mutex-Objekte ungleich Null sind eine der häufigsten Ursachen für den Ausfall der ACPI-Batteriemethode.

Im Lenovo u430 sind die Mutexe beispielsweise wie folgt deklariert:

Mutex (MSMI, 0x07).

Um es mit OS X kompatibel zu machen, müssen Sie es in ändern:

Mutex (MSMI, 0).

Dies ist ein sehr umstrittener Patch. Verwenden Sie es nur, wenn Sie genau wissen, was Sie tun. Rehabman in den Revisionen 4265-4346 hinzugefügt.

Khaki Klee. Version 5.1, Revision 5129
Moskau, 2021

121

FixRTC:

Entfernt einen Interrupt vom _RTC-Gerät. Dies ist notwendig, und es ist sehr seltsam, dass jemand ein solches Patch ablehnt. Wenn der ursprüngliche Interrupt noch nicht vorhanden ist, dann ist es in Ordnung, das dieser Patch nicht ausreichen wird. Es kam jedoch die Frage auf, ob die Länge der Region bearbeitet werden muss. An es gab keinen CMOS-Reset, die Länge sollte auf 2 gesetzt werden, aber der Satz erscheint im Kernel-Log wie... nur eine Bank...

Ich weiß nicht, was an dieser Meldung falsch ist, sie kann behoben werden, indem die Länge auf 8 gesetzt wird. Aber in In diesem Fall besteht die Gefahr eines Bios-Resets nach dem Sleep. Daher ist ein zusätzliche der Key zum Lösen eines solchen Tricks eingeführt worden:

```
<key>ACPI</key>
<dict>
  <key>DSDT</key>
  <dict>
    <key>Rtc8Allowed</key>
    <false/>
  </dict>
```

true - die Regionslänge bleibt bei 8 Byte, falls sie es war, false - wird um 2 Bytes korrigiert, wodurch ein CMOS-Reset zuverlässiger verhindert wird. Wie von vit9696 recherchiert, sollte die Länge der Region noch 8 sein, da sie für der Speicherung des Ruhezustandsschlüssels wichtig ist. Und der Fix selbst ist nützlich. Nun, auf Desktops ist der Ruhezustand nicht unbedingt nötig, also denken Sie vielleicht darüber nach, das CMOS zu dumpen.

FixTMR

Entfernt auf ähnliche Weise einen Interrupt vom _TMR-Timer. **Es ist veraltet und wird von MacOS nicht verwendet.**

AddIMEI

Erforderlicher Patch für SandyBridge und höher, um ein Gerät IMEI in den device tree (Gerätebaum) hinzuzufügen, wenn es dort nicht bereits vorhanden war.

FixIntelGfx

Der Patch für die integrierte Intel-Grafik ist getrennt vom Rest der Grafikkarten, d. h. es ist möglich, eine Injektion für Intel und nicht für Nvidia zu setzen.

FixWAK

Fügt Return zur Methode _WAK hinzu. Es wird unbedingt gebraucht, aber aus irgendeinem Grund ist es in der DSDT oft nicht enthalten. Offenbar haben sich die Autoren an andere Normen gehalten. In jedem Fall ist dieser der Fix vollkommen sicher.

DeleteUnused

Löscht unbenutzte Diskettengeräte. Was meinen Sie, worüber Sie sich Sorgen machen müssen? Tatsächlich werden dadurch noch CRT- und DVI-Geräte entfernt - absolut notwendig Die IntelX3100-Startbedingung auf Dell-Laptops. Ansonsten gibt es einen schwarzen Bildschirm, verifiziert durch Hunderte von Benutzern.

FixADP1

Korrigiert das ADP1-Gerät (power supply (Stromversorgung)), was für einen ordnungsgemäßen sleepe der Laptops notwendig ist, wenn sie an die Steckdose angeschlossen oder wenn sie von ihr getrennt werden.

AddPNLF

Fügt das PNLF-Gerät (Hintergrundbeleuchtung) ein, das für die korrekte Steuerung der Bildschirmhelligkeit erforderlich ist und es hilft seltsamerweise bei der Lösung von Schlafproblemen, auch bei Desktops.

FixS3D

In ähnlicher Weise löst dieser Patch das Schlafproblem.

FixACST

In einigen DSDTs gibt es ein Gerät oder eine Methode oder Variable namens ACST, aber dies ist der von MacOSX 10.8+ verwendete Name für die Steuerung des c-state. (**Völlig logisch?**).

Es führt zu einem Konflikt mit sehr unklarem Verhalten. Dieser Fix benennt alle Vorkommen von solchen Namen und wandelt sie in etwas Sicheres um. Ich kann nicht verstehen, wie Sie diese Serie von Patches ignorieren können! Braucht ihr Jungs nicht einen gut funktionierenden Computer?

AddHDMI

Fügt dem DSDT ein HDAU-Gerät hinzu, das dem HDMI-Ausgang der Graphik-Karte von ATI oder Nvidia. Da die Grafik-Karte separat von der Hauptplatine gekauft wird, ist es klar, dass die native DSDT einfach nicht über ein solches Gerät verfügt. Zusätzlich wird das Gerät mit Eigenschaft hda-gfx=onboard-1 bzw. onboard-2 ausgestattet:

-1 if UseIntelHDMI=false

-2 if there is an Intel port that occupied port 1 (wenn ein Intel-Port vorhanden ist, der Port 1 belegt hat).

FixRegions

Dies ist ein sehr spezielles Patch. Wenn die anderen Patches dazu gedacht waren, folgendes zu beheben BIOS.aml, um eine gute DSDT aus dem Nichts zu erzeugen, dann ist dieser Fix für endgültige Ausrichtung einer gut gemachten benutzerdefinierten DSDT.aml, aber für BIOS.aml ist es nutzlos. So sieht's aus.

Im DSDT gibt es z. B. Regionen, die eigene Adressen haben:

OperationRegion (GNVS, SystemMemory, 0xDE6A5E18, 0x01CD)

Das Problem ist, dass diese Regionsadresse dynamisch vom BIOS erzeugt wird und sie kann unterschiedlich von Last zu Last sein. Zuerst wurde es beim Wechseln des Gesamtspeichers bemerkt, dann beim Ändern der BIOS-Einstellungen, und auf meinem Computer hängt es auch von der Vorgeschichte der Bootens ab, z. B. von der Größe des belegten NVRAM. Es ist klar, dass in der benutzerdefinierten DSDT.aml diese Zahl fest vorgegeben ist und sollte daher nicht zutreffend sein. Die einfachste Beobachtung ist der Schlafmangel. Nach dem Korrigieren der Region erscheint der Schlaf, aber nur bis zum nächsten Neustart (Schicht). Dieser Fix fixiert alle Regionen in der benutzerdefinierten DSDT auf BIOS-DSDT-Werte und damit ist die Maske

```
<key>Fixes</key>
  <dict>
    <key>FixRegionen</key>
    <true/>
  </dict>
```

ist eine ausreichende Maske, wenn Sie eine gut gemachte DSDT mit allen mit den erforderlichen Korrekturen haben.

Es gibt noch einen weiteren Patch, der aber nicht DSDT, sondern alle ACPI-Tabellen im Allgemeinen betrifft, und der hier illegal durchgeführt wird.

FixHeaders

Es prüft die Header nicht nur von DSDT, sondern von allen ACPI-Tabellen im Allgemeinen und löst das Problem der chinesischen Zeichen in Tabellennamen, die MacOS nicht toleriert, und sofort Panik auslöst. Unabhängig davon, ob Sie ein Problem mit den Tabellen haben oder nicht, ist es sicherer, diese Korrektur zu aktivieren.

Patch-Auswahl

Wie entscheiden Sie, welche Patches notwendig, welche harmlos und welche gefährlich sind? Probieren sie es aus. Sie werden den Computer in keiner Weise zerstören. All dies geschieht nur während des Betriebes im Speicher, und das wird nach einem Neustart vergessen. Sie können einen Satz von Korrekturen testen, indem Sie Maske im grafischen Menü bearbeiten und dann das Ergebnisses durch Drücken von F5 - "Save DSDT- xxxx.aml speichern, korrigiert durch aktuelle Maske".

Sie können versuchen, mit der aktuellen Maske zu booten. Um zu vermeiden, dass ein echter, gepatchter DSDT, der bereits im System vorhanden ist, gestört wird, können Sie im Menü DSDT-Name: BIOS.aml

Wenn keine solche Datei gefunden wird, übernimmt das System die OEM-DSDT aus dem BIOS und nimmt die Korrekturen gemäß der angegebenen Maske vor. Wenn er nach dem Neustart des Computers fehlschlägt, gehen die aktuellen Einstellungen verloren und die Standardeinstellungen werden wirksam. Die Maske 0xFFFFFFFF entspricht dem Einschalten aller Fixes, und wenn das OS danach bootet, war die Arbeit der Programmierer nicht umsonst. Sie haben bereits durch die obige Beschreibung verstanden, dass Sie einige Fixes einfach nicht benötigen (z. B. WIFI). Seit der Revision 1992 wurde daran gearbeitet, Doppel-Patch-Panik zu verhindern, also haben Sie keine Angst, zusätzliche Bits zu setzen. Im Moment würde ich nicht empfehlen, zwei Patches zu verwenden: FIX_SATA, Bit 0x0400, besser ist es, den binären Patch zu verwenden, und FIX_SHUTDOWN, Bit 0x04, denn stattdessen funktioniert das Setzen von HaltEnabler=true fast immer korrekt. Auch gefährlich Patch AddMCHC_0008, jemand braucht es, und für jemand anderen ist es kategorisch kontraindiziert.

Um zu sehen, wie sich die DSDT-Patches auf das Ergebnis auswirken, und nehmen wir an, sie könnten nicht booten, dann können Sie den folgenden Schlüssel in den ACPI-Abschnitt der Config schreiben:

```
<key>DSDT</key>
<dict>
  <key>Debug</key>
  <true/>
```

Dann werden vor dem Start des Systems zwei Dateien im Ordner auf der Festplatte gespeichert
/EFI/CLOVER/ACPI/origin/

DSDT-or.aml.

DSDT-pa15.aml

origin ist Ihre DSDT, die entweder von der Festplatte geladen oder aus dem BIOS entnommen wird, bevor Patches eingespielt wurden.

patched - nachdem Patches eingespielt wurden. Da Sie beim Booten gescheitert sind, werden Sie immer mehrere Versuche unternehmen, und die gepatchten Dateien werden fortlaufend nummeriert, ohne dass die alten Informationen gelöscht werden. 15 in diesem Fall - dies ist der 15. Versuch des Startens in diesem Fall, sollte erfolgreich sein und wir müssen schauen, was das Problem beim 14. Versuch war. Ich empfehle jedoch, das doppelte Patchen zu vermeiden. Es kann auch eine Situation entstehen, in der doppelt gepatcht wird, weil das OEM-DSDT bereits eine _DSM-Methode enthält, wenn wir unsere eigene injizieren wollen. Sie müssen also Bits in der DropOEM_DSM-Maske freilegen. Siehe Hardware-Konfiguration → ACPI → DSDT → DropOEM_DSM. Nein, Sie müssen alle OEM-DSMs umbenennen.

Dennoch empfehle ich, ein doppeltes Patch zu vermeiden. Es kann auch eine Situation entstehen, in der Das doppelte Patchen tritt auf, weil die OEM-DSDT bereits die Methode _DSM hat, wenn wollen wir unsere eigene entwickeln. **Sie müssen also die Bits in der Maske setzen DropOEM_DSM. Siehe Hardware-Konfiguration → ACPI → DSDT → DropOEM_DSM.** Nein, Sie müssen alle OEM-DSMs umbenennen.

Manuelles Editieren der DSDT

Voraussetzungen

Sie benötigen einen Computer mit einem bestimmten Betriebssystem, um Textdateien bearbeiten zu können. Oder Sie haben bereits irgendwie macOS auf diesem Computer installiert und möchten nun DSDT erweitern. Für jedes Betriebssystem gibt es einen iasl-Befehlszeilen-Compiler. Einschließlich für Windows läuft iasl.exe auf der Kommandozeile wie unter macOS, hat die gleichen Funktionen und liefert die gleichen Ergebnisse. Es ist nicht komfortabel, Texte in Windows zu bearbeiten, Notepad hat keine Syntaxhervorhebung und Zeilennummerierung, besser ist es, Notepad+ zu bekommen. Es gibt viele Optionen in Mac, und Xcode, und BBEdit, und andere. Sie können die neue Version des Compilers für Mac hier erhalten
Dsd-Optimierung. Neuester Compiler.

Für Windows unter <https://acpica.org/Windows Binary Tools>

Sie sollten auch die ACPI-Sprachbeschreibung ACPI_6_2.pdf vorrätig haben. Es ist besser, eine neue Version zu besorgen, weil Sie mit der DSDT umgehen müssen, das sie aus dem frische BIOS erhalten haben (https://uefi.org/sites/default/files/resources/ACPI_6_2.pdf).

Creating a blank (Erstellen eines Rohlings)

Erstellen einer Voreinstellung

Besorgen sie sich zunächst ein-Flash-Drive (USB-Stick), der bootfähig ist, auch wenn kein System vorhanden ist. Es ist wichtig, dass er im FAT32-Format vorliegt. Dort wird Clover darauf kopiert. Booten sie von diesem Flash-Laufwerk (USB-Stick) den Computer, um ins Clover Interface (Menue) zu kommen. Drücken Sie die Taste "O" (lateinischer Buchstabe O) oder wählen Sie das Symbol im Menü Optionen. Gehen Sie zu ACPI-> dort finden Sie DSDT Name: und geben sie BIOS.aml ein. Dies ist genau die DSDT, die Sie unter Windows über Aida erhalten würden. Gehen Sie im Menü nach unten und wählen sie DSDT-Fixes -> dort können Sie Kästchen ankreuzen, entweder mit der Tastatur oder mit der Maus. Setzen Sie fast alle Kontrollkästchen, die Sie nicht stören. Zum Beispiel Firewire, Airport, IDE werden nicht benötigt, wenn Sie wissen, dass Sie keine solchen Geräte haben. Alles, was Sie nicht kennen, sollten Sie überprüfen. Gehen Sie aus diesem Untermenü zurück, und drücken Sie nacheinander F2, F4, F5.

Jetzt können sie den Computer ausschalten und das Flash-Laufwerk an dem Computer anstecken, mit dem sie die Dateien bearbeiten wollen. Die Dateien, an denen wir interessiert sind, befinden sich auf dem Flash-Laufwerk in den Ordnern:

```
EFI\CLOVER\ACPI\origin  
EFI\CLOVER\misc
```

Kopieren Sie diese Ordner in Ihren Arbeitsordner auf Ihrem Computer, in dem sich iasl-Programm befindet. Entweder mit MacOS oder mit Windows. Die Anweisungen sind die gleichen, abgesehen von Feinheiten wie wie Schrägstriche ([Schräglage\(?\)](#)).

Decompilation

Die resultierende DSDT.aml ist eine Binärdatei, die Sie nicht mit einem Texteditor bearbeiten ([öffnen?](#)) können. Versuchen sie es und schauen Sie sich das an. Von spezialisierten maciASL-Editoren rate ich kategorisch ab, DSDT-Editor und dergleichen, da der Zweck des Themas nicht darin besteht, den guten [Onkel rehab\(?\)](#), der das für sie macht. Es geht darum, mit eigenen Augen zu sehen, was passiert und wie es passiert.

Führen Sie die Befehlszeile aus: cmd.exe unter Windows, Terminal.app unter Mac, whatever unter Linux. Ich weiß es nicht, ich glaube, es heißt "Bash".

```
> cd „ihr ArbeitsOrdner“/origin
```

d. h. in dem gleichen Ordners, in den sie die vom Flash-Laufwerk erhaltenen Dateien kopiert haben. In den gleichen Ordner legen iasl.exe, wenn Sie auf Windows sind, oder installieren Sie iasl auf dem System, wenn Sie auf MacOS sind:

```
$ sudo cp ~/Downloads/iasl /usr/local/bin/
```

Jetzt können wir endlich dekompilein, d. h. die DSDT in eine lesbare Sprache bringen.

```
$ iasl -da SSDT*.aml DSDT-*.aml
```

Die -da Option bewirkt das Wunder, es wird in allen Dateien gleichzeitig nach Zeichen gesucht, so dass es am Ende keine unbekanntenen Zeichen mehr geben sollte. Dies gilt jedoch nur für den Idealfall. In der Praxis funktioniert nicht es, wir erhalten eine Menge Fehler wie AE_ALREADY_EXISTS

~~~~

```
Firmware Error (ACPI): Failure creating named object [IO1B], AE_ALREADY_EXISTS (20200925/ dswload-495) ACPI Error: AE_ALREADY_EXISTS, During name lookup/catalog (20200925/psobject-372) Could not parse external ACPI tables, AE_ALREADY_EXISTS
```

~~~~

Die andere Option erweist sich als erfolgreicher

```
iasl -e SSDT*.aml -d DSDT-*.aml
```

Die Ausgabe ist eine Warnung am Ende:

```
Parsing completed
```

```
Warning - Emitting ASL code "External (BNUM)"
```

```
This is a conflicting declaration with some other declaration within the ASL code
```

```
This external declaration may need to be deleted in order to recompile the dsl . file.
```

Das heißt, in der DSDT.dsl müssen wir die ad entfernen

```
Deklaration External(BNUM, UnknownObj) // Conflicts with a later declaration
```

Because there is an ad (**Da es eine Ankündigung gibt?**)

```
Field (GNVS, AnyAcc, Lock, Preserve)
```

```
{
```

```
... BNUM, 8,
```

Das heißt, der Name wird hier bekannt gegeben, und Sie müssen ihn nicht extern benennen, oder gar einen unbekanntenen Namen nehmen. Aber dann sind da noch die Fehler

```
External (_PR_.CFGD, UnknownObj)
```

Dennoch existiert diese Variable und wird in einer der SSDTs beschrieben. Das Decompilat ist noch unvollständig. Jeder dieser Fälle muss einzeln behandelt werden.

Wir dekompilein die DSDT, die Clover gepatcht hat, und haben somit bereits eine Menge nützlicher Korrekturen. Auf ähnliche Weise können Sie die Originaldatei dekompilein

```
$ iasl -da SSDT*.aml DSDT.aml
```

Und um vergleichen zu können, was war und was nicht war. Und wir haben eine Datei mit dem langen Namen DSDT-1234567.dsl, (wahrscheinlich haben Sie andere Zahlen und Buchstaben). Dies ist die ursprüngliche Voreinstellung, die in DSDT.dsl umbenannt werden sollte, und das Original in DSDT-origin.dsl, bearbeiten und kompilieren in einem unendlichen, iterativen Prozess:

1. Mit einem Texteditor bearbeiten.

2. Kompilieren, DSDT.aml erhalten

Kompilierbefehl

```
$ iasl -ta DSDT.dsl
```

3. Testen Sie, d.h. legen Sie es in den Ordner EFI\CLOVER\ACPI\patched und starten Sie das System.

4. Zurück zu Punkt 1.

Leider kann die Kompilierung mit folgendem Fehler fehlschlagen:

Code:

Input file SSDT-10x.aml, Length 0x37F (895) bytes

ACPI: SSDT 0x0000000000000000 00037F (v02 PmRef Cpu0Cst 00003001 INTL 20120913)

Pass 1 parse of [SSDT]

ACPI Error: [C3ST] Namespace lookup failure, AE_ALREADY_EXISTS (20160729 / dswload-462)

ACPI Exception: AE_ALREADY_EXISTS, During name lookup / catalog (20160729 / pobject-310)

Could not parse external ACPI tables, AE_ALREADY_EXIST

Das bedeutet, dass die Beschreibung des C3ST-Symbols in zwei verschiedenen SSDTs zu finden ist, die letzte davon ist SSDT-10x.

In meinem Fall stellte sich heraus, dass es ähnlich wie SSDT-5x ist, mit dem Unterschied, dass 5x für die Version von ACPI1.0 ist und 10x für die ACPI2.0 Version. Und die Namen sind die gleichen! Ich habe etwas Ähnliches im Kernelprotokoll. Es ist BIOS-Fehler!

Außerdem befinden sich zwei identische Tabellen (**Tische?**) im Laptop. Ich bin dem nachgegangen, sie sind eigentlich beide im BIOS vorhanden, das ist nicht mein Fehler (**Schuld?**).

Was ist zu tun? Entfernen Sie die Zweitbesetzung vor dem Kompilieren, und in der realen Arbeit tun Sie es in der config Clover Drop extra Tabellen (**Zusatztische?**). Im Falle von Duplikaten fliegen beide weg.

Was ist zu reparieren

1. Syntaxfehler durch den Hersteller dieses Computers
2. **Sinnvolle Fehler(?)**
3. Tricks aus dem Internet. Zu diesem Zeitpunkt denkt jeder, es ist die einzige Sache, die getan werden muss. Nein, auch die ersten beiden Punkte sind wichtig.

Syntax-Fehler

Betrachtet man die Datei DSDT.dsl selbst, so sieht man die ersten Probleme in der Datei selbst

```
External (_SB_.PCI0.PEG0.VID_.LCD_, UnknownObj)
```

Es bedeutet, dass es irgendwo im Text einen Verweis auf ein Objekt gibt, das Objekt selbst aber nirgends zu finden ist. Wir sind auf der Suche nach ihr. Es ist befindet sich in SSDT-7. Es stellt sich heraus, dass es nicht von dort exportiert wurde. (Aus diesem Grund ist es dieser Laptop schlägt vor, dass dieses SSDT in ein **generisches DSDT aufgenommen wird?**). Durch einfaches Kopieren des Textes, der SSDT von der ersten geschweiften Klammer bis zur letzten im Ende von DSDT.dsl vor die letzte Klammer.

Zweiter Fehler mit fehlenden Zeichen

```
External (HNOT, MethodObj) // Warning: Unknown method, guessing 1 arguments
```

Eine Suche zeigt, dass die Methode in diesem design code (Konstruktion) erwähnt wird:

Eine Suche zeigt, dass die Methode in der folgenden Konstruktion erwähnt wird

Code:

```
If (CondRefOf (HNOT))
{
    HNOT (Arg0)
}
Else
{
    Notify (GFX0, 0x80) // Status Change
}
```

Dann ist alles in Ordnung. Wenn die Methode undefiniert ist, wird sie nicht ausgeführt. Es ist mir nicht klar, wie wird es dann kompiliert? Besser, Sie löschen es, lassen nur Notify...

Ausführen der Kompilierung der Datei, die ich gerade erhalten habe (ich habe DSDT-1F2C3B4D.dsl in die einfachere DSDT1.dsl umbenannt, als ersten Versuch)

```
$ iasl -ta DSDT1.dsl
```

1 Fehler, 14 Warnungen, 91 Anmerkungen, 109 Optimierungen

Der Fehler muss behoben werden, sonst gibt es kein Ergebnis.

Sie müssen auch die Warnungen beheben und darüber nachdenken, womit der Compiler kämpft.

Das ist immer noch eine gute Situation - ich bin beim ersten Kompilieren auf hunderte von Fehlern gestoßen.

In diesem Fall ist der Fehler unkritisch, in der Zeile

```
Name (_HID, "*pnp0c14")
```

Zeichenkettenformat ist ungültig, es wird nach Vorschrift (wissenschaftlich) wie folgt korrigiert :

```
Name (_HID, EisaId ("PNP0C14") /* Windows Management Instrumentation Device */)
```

Es muss korrigiert werden, sonst wird es nicht kompiliert, aber es hat keinen Einfluss auf den Betrieb, es ist nur für Windows von Bedeutung.

Warnungen sind eigentlich kritischer (**Die Lackierungen sind wirklich kritischer**), hier sind Beispiele für Korrekturen.

- das war: - CreateDWordField (BUF0, _SB.PCI0._Y0F._LEN, MSLN) // _LEN: Länge

+ was wurde geändert: CreateQWordField (BUF0, _SB.PCI0._Y0F._LEN, MSLN) // _LEN: Länge

Der Hinweis stand im Kompilierungsprotokoll:

ResourceTag larger than field (size mismatch tag 64bit, Field 32 bit

Die Regel ist einfach: Tag ist das, was Sie brauchen, Field ist das, was Sie reparieren müssen:

```
tag=1 => CreateBitField
```

```
tag=8 => CreateByteField
```

```
tag=16 => CreateWordField
```

```
tag=32 => CreateDWordField
```

```
tag=64 => CreateQWordField
```

Es muss korrigiert werden, damit im realen Betrieb der gesendete Wert nicht über den Nachbarwert hinauschießt das nächste Feld. Es gibt keine Möglichkeit, diese Störung abzufangen.

Diese Art der Warnung:

```
Not all control path return a value
```

Es ist ein Fehler der Logik, die Methode muss etwas zurückgeben, aber es stellt sich heraus, dass in einigen Fällen nichts zurück gegeben wird. MacOS wird natürlich nicht darüber stolpern, aber erwarten Sie auch keine Angemessenheit. Und was soll ich dort schreiben? Return oder Return(Local0)? Es reicht, wenn der Compiler sich entspannt, aber im Allgemeinen sollten Sie sich die Logik ansehen.

Ähnliche Störung

Reserved method should not return a value

Der Code lautet

Code:

```
Method (_SRS, 1, Serialized) // _SRS: Set Resource
Settings
{
    Return (BUF2) / * \ _SB_.PCI0.A_CC.BUF2 * /
}
```

Öffnen Sie das oben erwähnte Pdf, ADCPI-Spezifikation, suchen Sie nach der _SRS-Methode, und lesen Sie, was sie tun soll. Die Rückgabe von Werten ist dort nicht vorgesehen.

Lassen Sie es uns also auf folgende Weise umarbeiten

Code:

```
Method (_SRS, 1, Serialized) // _SRS: Set Resource
Settings
{
    BUF2 = Arg0
}
```

So it seems more logical.

Well, well-known warning

Name (_T_0, Zero)

Use of compiler reserved name _T_0

Der neue iasl-Compiler kommt mit diesem Konstrukt sehr gut zurecht, und Sie brauchen nichts zu tun. Der alte Compiler erforderte die Umbenennung in TT_0

Conceptual errors (Sinnesfehler?)

Sorry, aber man muss zumindest ein bisschen Programmierer sein, um sie zu erkennen. Sie können von nirgendwo Hinweise erwarten.

In der DSDT-Quelle können wir sehen

Extern (CFGD, IntObj)

Diese Variable finden wir in der SSDT CpuPm, und es ist Zeit, sich daran zu erinnern, dass wir diese Tabelle haben zusammen mit dieser Variable verworfen haben!

Wir sollten es in DSDT kopieren. Sowie andere, die sich als nützlich erweisen könnten.

Code:

- + Name (CFGD, 0x0066F6FF)
- + Name (PDC0, 0x80000000)
- + Name (PDC1, 0x80000000)
- + Name (PDC2, 0x80000000)
- + Name (PDC3, 0x80000000)
- + Name (PDC4, 0x80000000)
- + Name (PDC5, 0x80000000)
- + Name (PDC6, 0x80000000)
- + Name (PDC7, 0x80000000)
- + Name (SDTL, Zero)

Ein häufiger Fehler ist die OEM-Methode `_DSM`.

Es ist nicht der Fehler des Herstellers, es ist die Tatsache, dass er für Windows geschrieben hat, und wir wollen aber einen Hackintosh betreiben.

Beispiel

Code:

```
- Method (_DSM, 4, Serialized) // _DSM: Device Specific Method
  {
-   Name (_T_0, Null) // _T_x:
      Emitted by ASL-Compiler
      If (Arg0 == ToUID ("a5fc708f-8775-4ba6-bd0c-ba90a1ec72f8"))
      {
          While (One)
          {
```

Sehen Sie die UUID? Sie stammt aus der Windows-Registrierung, bei dem MacOS gibt es so etwas nicht, und es läuft auch nichts damit. Das wäre nur die halbe Miete, aber wenn es für die Windows in Ordnung ist, `_DSM` sowohl für das Gerät als auch für den seine Bridge zu haben, verursacht es einen Absturz in MacOS. Lasst uns alle anderen Leute `_DSM töten!(?)`. Der einfache Weg:

Code:

```
- Methode (_DSM, 4, Serialized) // _DSM: Device Specific Method
+ Methode (ZDSM, 4, Serialized)
```

Die Methode selbst hat überlebt, aber es wird sich niemand mehr an sie wenden, und es wird keine Panik entstehen.

Hier ist ein kuriozes Stück

Code:

```
OperationRegion (DXHC, SystemMemory, 0xFED1F418, 0x04
  XHCD, 1
}
If ((OSYS < 0x07D6) && (OSYS > 0x03E8))
{
  XHCD = One
  Notify (XHC, Zero) // Bus Check
}
```

Nachdem ich mir die Adresse genauer angesehen habe, ist mir klar geworden, dass es sich um ein Function Disable Bit handelt.

Der Sinn der Operation ist es, USB3 für Systeme unterhalb von Windows Vista zu sperren. Ich denke, dieser Teil sollte ganz herausgeschnitten werden.

Das ist der Schwachsinn der Autoren.

Code:

```
If ((OSYS > 0x07D0) || (OSYS < 0x07D6))
```

Bereiche addieren sich und überschneiden sich mit beliebigen Werten

So ist eher, wie es sein sollte.

Code:

```
If ((OSYS > 0x07D0) && (OSYS < 0x07D6))
```

Dann überschneiden sich die Bereiche. Aber dann müssen wir sehen, was in der If und was in der Else Anweisung ist. Es stellte sich heraus, dass es dann richtiger ist, also die ursprüngliche Konstruktion also das ursprüngliche Design

```

{
    Notify (PCI0, Arg1) }
Else
{
    Notify (GFX0, Arg1) }
}

```

Ich habe es nur auf einen operator (**Bediener?**) reduziert.

Notify (PCI0, Arg1)

Der Punkt ist, dass dieses If prüft, ob das System WindowsXP ist, oder Else was es dann für die Systeme wie Windows7,8,10 macht.

Was ist der Unterschied? Optimus arbeitet in dem neuen Systemen. Unter **MacOS** benötigen wir eine first-class notification, für den gesamten Bus. Und in ähnlicher Weise müssen wir an anderer Stelle in der DSDT verfahren.

Ein kleiner Krimi zum Thema, vielleicht kommt ja jemand auf die richtige Idee.

Also, die Herausforderung. Der Laptop schläft und wacht auf, wenn er nicht eingesteckt ist. Aber dann ist die Batterie leer. Wenn er jedoch eingesteckt ist, wacht er sofort auf, eine Sekunde nachdem er eingeschlafen ist. Was ist denn los?

Genau bei der DSDT!

Also, auf der Suche nach dem, was den Unterschied zwischen **ACPI-Status in the outlet (im Sockel?)** und nicht macht und welches Gerät das verursacht.

Code:

```

Device(AC)
{
    Name (_HID, "ACPI0003"/* Power Source Device */) // _HID: Hardware-ID

```

HID: Hardware ID

Der Decompiler hat uns freundlicherweise mitgeteilt, dass dies die Stromversorgung ist.

Die Methode _PSR bestimmt, ob dieses Gerät eingeschaltet ist oder ruht.

Code:

```

Method(_PSR, 0, NotSerialized)    // _PSR: Power Source
{
    Lokal0 = EVG5 ()
    Lokal0 &= One
    If (Lokal0 != PWRS)
    {
        PWRS = Lokal0
        PNOT ()
    }

    Return (Lokal0)
}

```

Sie können es im Buch ACPIspec.pdf nachlesen, beliebige Jahresausgabe:

Ein Integer, der den Status der Stromquelle enthält

0 - Off-line (nicht am Netz)

1 - On-line

Das heißt, in meinem Fall gibt eine fest verdrahtete Methode ECG5() eine Zahl aus, in der das Bit 0 bedeutet eingesteckt, und speichert es in der PWRS-Variable, wenn der Wert dort anders ist.

Suchen Sie nun im gesamten Code, wo PWRS erwähnt wird, und finden Sie es in der Methode _PTS (Prepare To Sleep), der für das Einschlafen verantwortlich ist.

Code:

```
Method (_PTS, 1, NotSerialized)    // _PTS: Prepare To Sleep
{
    P80D = zero
    P8XH (zero, Arg0)
    PTS (Arg0)
    If (AOAC & one) {}
    If (Arg0 == 0x03)
    {
        If (PWRS == zero)
        {
            \_SB.PCI0.XHC.PMEB = zero
            \_SB.PCI0.EHC1.PMEB = zero
            \_SB.PCI0.EHC2.PMEB = zero
            If (\_SB.PCI0.XHC.PMST == one)
            {
                \_SB.PCI0.XHC.PMST = one
            }

            If (\_SB.PCI0.EHC1.PMST == one)
            {
                \_SB.PCI0.EHC1.PMST = one
            }

            If (\_SB.PCI0.EHC2.PMST == one)
            {
                \_SB.PCI0.EHC2.PMST = one
            }
        }
    }
}
```

Interessant hier! Voraussetzung: Wenn das Gerät nicht in die Steckdose gesteckt ist, dann tun Sie etwas in SSB2 und SSB3, und wenn sie es tun, müssen Sie nichts tun. Im Protokoll des unerwarteten Aufwachens sehen wir XHC, EHC2.

Die Ersteller dieser DSDT glauben also, dass ein Laptop nur eingesteckt bedient werden kann. Ist es nicht mit der Stromquelle verbunden, sollte es nicht arbeiten.

Durch Entfernen dieser Bedingung If (PWRS == Zero) habe ich die Lösung für mein Problem erhalten. Ich habe es jetzt so, daß der Laptop schläft und wacht auf, und das Netzteil stört ihn nicht.

Viel Erfolg bei der Erstellung einer minimal korrekten DSDT!

Native Speedstep

Korrekter ist es, CPU Power and Speed Management (CPU PSM) zu sagen, auf Englisch heißt es EIST - Enhanced Intel Speedstep Technology, daher das russische Wort "Speedstep".

Eigentlich geht es in diesem Thema nicht so sehr um den Bootloader, sondern um die Konfiguration von Hackintosh im Allgemeinen, aber da Clover einige Schritte durchführt, werden wir dies in einem eigenen Kapitel beschreiben.

Clover macht nicht alles, was er machen muss, es ist auch etwas Handarbeit erforderlich. Was ist überhaupt der Zweck davon? Der Punkt ist folgender: Der Prozessor läuft im Leerlauf mit minimaler Frequenz bei minimaler Spannung, unter Last wachsen Drehzahl und Spannung. (Warum die Spannung? Und weil die Impulsfront steiler wird, und damit Pegel schneller, geht es schneller von Zustand 0 zu Zustand 1).

UPiP kann auf zwei Arten durchgeführt werden: durch ein spezialisiertes Dienstprogramm, wie z. B. <http://www.coolbook.se/CoolBook.html> (nicht für moderne Systeme), oder GenericCPUPM, oder durch das Verstehen des nativen Speedstep, dank der Tatsache, dass MacOS dies kann.

Die folgenden Schritte sind erforderlich:

1. Es ist notwendig, HPET im SDT zu fixieren, was erfolgreich durch von Clover mit 0x0010-Maske geschieht.
2. Es muss ein korrekter Prozessorteil vorhanden sein, was von Clover mit GeneratePStates=Yes Key (und zusätzlich DropSsdT).
3. Als Beispiel für Ihr SMBIOS muss ein MacModell ausgewählt werden, für das die EIST-Technologie ist vorgesehen für. Es stellt sich heraus, dass dies nicht für alle Modelle gilt. Zum Beispiel wird Speedstep nicht für MacBook1.1 funktionieren, aber für MacBook5.1 schon.
Punkt 3 kann wie folgt umgedacht werden: Lassen Sie das Modell in der Konfiguration dem realen Modell ähnlicher sein, aber wir werden seine Plattform-Pliste so festlegen, dass die Speedstep auftaucht.
Für jedes Modell gibt es eine eigene Plist, siehe hier

System/Library/Extensions/IOPlatformPluginFamily.kext/Contents/PlugIns/

ACPI_SMC_PlatformPlugin.kext/Contents/Resources/MacBook5_1.plist

Sehen Sie die Gemeinsamkeiten und Unterschiede der verschiedenen Listen, und korrigieren Sie Ihre in die richtige Richtung.

ConfigArray

```
<key>ConfigArray</key>
<array>
  <dict>
    <key>WWEN</key>
    <true/>
    <key>model</key>
    <string>MacBook4,1</string>
    <key>Restart-Aktion</key>
    <dict>
      <key>cpu-p-state</key>
      <integer>0</integer>
    </dict>
  </dict>
</array>
```

Dieser Neustart-Aktionsschlüssel bedeutet, in welchen P-State die CPU beim Neustart fallen soll. Nur mit diesem Key funktioniert der Ruhezustand und das Herunterfahren des Rechners!

CtrlLoopArray

```
<key>CtrlLoopArray</key>
<array>
  <dict>
    <key>Description</key>
    <string>SMC_CPU_Control_Loop</string>
    ...
    <key>PLimitDict</key>
    <dict>
      <key>MacBook4,1</key>
      <integer>0</integer>
    </dict>
  </dict>
```

Dieser PLimitDict-Schlüssel wurde bereits bei der Erzeugung von P-Zuständen erwähnt. Wiederholen wir es: Diese Einschränkung bewirkt die maximale Geschwindigkeit des Prozessors. 0 - die Geschwindigkeit ist maximal, 1- eine Stufe niedriger als das Maximum. Fehlt dieser Key hier jedoch, bleibt der Prozessor bei der Mindestfrequenz.

CStateDict

```
<key>CStateDict</key>
<dict>
  <key>MacBook4,1</key>
  <string>CSD3</string>
  <key>CSD3</key>
  <dict>
    <key>C6</key>
    <dict>
      <key>Einschalten</key>
      <true/>
    </dict>
  </dict>
</dict>
```

Die Praxis zeigt, dass es besser ist, den gesamten Abschnitt zu entfernen, damit die Steuerung Spannungsversorgung genau nach PState, nicht nach Cstate funktioniert. Obwohl, es von der eigenen Meinung abhängt, vielleicht diese Option durchzuarbeiten.

Das Symptom ist, dass der Prozessor mit maximaler Frequenz arbeitet und nicht abfällt. Nach dem Entfernen des Abschnitts beginnt in der Häufigkeit zu variieren.

Mit Skylake-Prozessoren gibt es eine neue Methodik zur Verwaltung der CPU-Frequenz, in ihrem eigenen Namen, heißt es SpeedShift. Ich habe es noch nicht herausgefunden, wie es funktioniert, vorläufige Ergebnisse:

Im CPU-Bereich schreiben wir zwei Keys (der Erfinder ist goodwin_c)

```
<key>CPU</key>
<dict>
  <key>HWPEnable</key>
  <true/>
  <key>HWPValue</key>
  <string>0x30002a01</string>
</dict>
```

Die erste Taste setzt eine 1 in das MSR-Register 0x770.

Die zweite Taste schreibt den angegebenen Wert in das MSR-Register 0x774.

Sleep problem (Das Problem mit dem Sleep (Schlaf))

Was ist mit dem Schlafproblem? Wenn alle oben genannten Schritte durchgeführt wurden, wird der Computer ins Bett gehen und wie ein gehorsames Kind aufwachen. Das Wichtigste, was Sie dazu tun müssen hat Clover bereits getan: korrigierte FADT und FACS. Es bleibt nur noch die DSDT zu korrigieren, besorgen Sie sich einen guten Speedstep Kext,, und Sie werden glücklich sein.

Jedes Gerät kann eine gute Nachtruhe stören, auch ein stromloser PCI Gerät, oder teilweise arbeitende Geräte, zum Beispiel AppleHDA. Der Schlaf wird stark durch den NullCPUPM.kext gestört. Möglicherweise benötigen Sie keinen Speedstep, aber Sie sollten den HPET-Patch auf diese Weise erstellen, um den nativen AppleCPUPM auszuführen zu können, und dann wird der NullCPUPM.kex nicht benötigt. Und für diejenigen, die keinen Prozessor haben, der es erlaubt, AppleCPUPM zu verwenden, können Sie SleepEnabler ausprobieren - manchmal hilft das, oder einen gepatchten Kernel.

DSDT hat eine Gruppe von _GPE-Methoden mit Benachrichtigungen pro Gerät, die aus ihrem Schlaf geweckt werden müssen. Der Computer selbst ist wach, aber es kann sich herausstellen, dass er Video/Netzwerk/Ton/Maus vergessen hat aufzuwecken. Siehe DSDT, lernen Sie die Theorie, wie es geht.

[Außerdem gab es ein Problem mit dem Ruhezustand beim UEFI-Boot auf einem 10.8-System.](#)

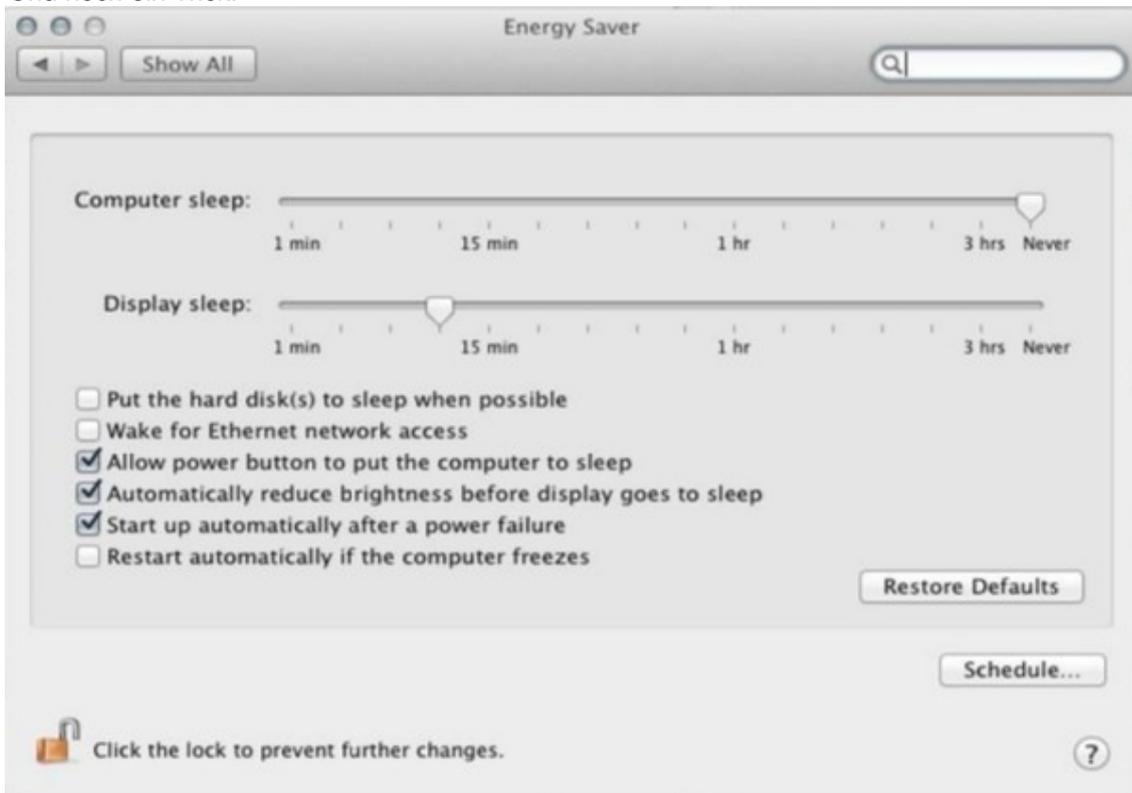
[Das Problem wurde in der Revision 1942 behoben. Änderung im OsxAptioFixDxe-Treiber: sleep/wakeup funktioniert auch in 10.8, sogar mit CsmVideoDxe.](#) Heutzutage ist dafür ProtectMemoryRegions zuständig.

Nächster Trick für UEFI-Boot

```
<key>ACPI</key>
<dict>
  <key>HaltEnabler</key>
  <true/>
</dict>
```

Dies korrigiert den von UEFI-bios fehlinitialisierten Zustand des Chipsatzes. Für Legacy-Downloads macht CloverEFI alles richtig, ich habe dieses Problem dort nicht gesehen.
Symptome – der Computer geht nicht richtig in den Schlaf, Bildschirm geht aus, aber der Lüfter nicht.

Und noch ein Trick.



Ohne das Kontrollkästchen "Automatisch starten..." hatte ich keine Möglichkeit, nach einem Nickerchen aufzuwachen.

Hibernate

Man nennt es auch Tiefschlaf, aber im Allgemeinen ist es eher der klinische Tod. Der Punkt ist, dass das System scheinbar in den Ruhezustand versetzt wird, aber es speichert seinen Zustand in dem sleepimage auf der Festplatte, und schaltet den Computer einfach aus, damit er beim Wiedereinschalten nur seinen Zustand wieder herstellt und er ist aufgewacht. Für Laptops ist dies entscheidend. Wenn im normalen Ruhezustand der Computer nicht vollständig heruntergefahren wird und weiterhin Elektrizität verwendet, wenn auch weniger als im Arbeitszustand, aber immer noch spürbar für die Batterie wird sie während einer gewissen Ruhezeit vollständig entladen sein. Während des Hibernate-Ruhezustand wird die Batterie nicht verwendet, und nur durch seine Leckage entladen wird. Da der Zustand des Computers im Sleepimage gespeichert wird, gehen dennoch keine Daten verloren.

Hibernate funktionierte vor langer Zeit mit Chameleon, aber nur bis zur Version 10.7.2 (glaube ich), dann funktionierte diese Methodik nicht mehr, aufgrund einiger Änderungen im System. Mit Clover wurde es geschafft, einen Hibernate zu machen, aber unter folgenden Bedingungen:

- Laden Sie entweder CloverEFI (Legacy), InsydeEFI oder Phoenix 2.3.1. Mit der aktuellen Revision 2915+ wurde der Treiber OsxAptioFix2Drv eingeführt, mit dem hibernate mit AMI UEFI auf einem 10.9.1+ System funktioniert. Aber das System 10.7.5 bootet überhaupt nicht mit dieser Option. Ach, und was soll's!

- Das System ist entweder 10.7.5 oder 10.9.1+. Andere Systeme sind noch nicht aufgewacht. - Mode 21 oder, noch besser, 29 oder sogar 57, obwohl Apple auf 25 beharrt.

```
sudo pmset -a hibernatemode 29
```

Ab Captain kann man das nicht mehr de 29 benutzen. Der Trick ist folgendermaßen:

- kopieren Sie /Library/Preferences/com.apple.PowerManagement.511CE201-1000-4000- 9999-120361221216.plist auf dem Desktop.

Es gibt mehrere solcher Dateien, wählen Sie die gewünschte aus, die mit der UUID in den Systeminformationen übereinstimmt. Sie bearbeiten diese Datei, zum Einfügen von 29

```
<key>Hibernate-Mode</key>  
<integer>29</integer>
```

Und kopieren Sie die bearbeitete Datei mit Hilfe des Terminals zurück

```
sudo cp ~/Desktop/ com.apple.PowerManagement.511CE201-1000-4000-9999- 120361221216.plist  
/Library/Preferences/
```

Danach ist ein Neustart erforderlich, erst dann wird die Änderung wirksam.

- Wenn Sie jedoch ein echtes NVRAM laufen haben, können Sie Mod 25 ausführen, dann in der Konfiguration

```
<key>Boot</key>  
<dict>  
  <key>StrictHibernate</key><0>  
</dict>
```

- Für System 10.13 und höher müssen Sie Folgendes aktivieren:

```
<key>RtcHibernateAware</key>  
<true/>
```

weil der Chiffrierschlüssel in das CMOS geschrieben werden kann, und weil Clover zusätzliche NVRAM-Variablen erzeugen sollte. (über letzteres werden sie noch nachdenken müssen).

Es funktioniert wie folgt:

1. Stellen Sie Mod 29 (oder 25) ein, falls nicht bereits eingestellt. Es ist nicht notwendig, dies zu wiederholen.
2. Versetzen Sie den Computer in den Ruhezustand, entweder über das Menü oder durch Schließen des Deckels oder durch Drücken der Powerknopf, wenn dies so konfiguriert ist. Nach einer Minute wird der Computer vollständig heruntergefahren sein.
3. Um aufzuwachen, schalten Sie es einfach wie gewohnt ein. Sehen Sie den BIOS-Startbildschirm, geben Sie das Menü "Clover". Und hier sehen wir, dass unser System markiert ist (im Winterschlaf)



ei anderen Systemen ist dies nicht der Fall. Wenn dieses Symbol angeklickt wird, wird das System mit dem markierten Bild in ein paar Sekunden gebootet. Sie können den Fortschritt am unteren Rand sehen, und das System schaltet sich ein. Es ist viel schneller als ein normaler Systemstart, insbesondere bei Laptops, und vor allem bei eine große Anzahl von offenen Anwendungen.

Es ist zu beachten, dass, wenn das Dateisystem des Datenträgers geändert wurde, nachdem sie z. B. von einem System, das von einer zweiten Partition geladen wurde, darauf zugreifen und es verändern, gibt es eine ernsthafte Gefahr der Beschädigung des Dateisystems, da im schlafenden System ein Cache mit anderen Strukturen ist. Für das System 10.9 wird diese Schwierigkeit automatisch überwunden, indem der Vergleich Änderungsdaten stattfindet. Dies funktioniert nicht im System 10.7.5, vergewissern Sie sich manuell, dass alles korrekt ist.

Sie können das Aufwachen aus dem Bild abbrechen, indem Sie die Leertaste auf diesem Symbol drücken, und indem Sie "Ruhezustand abbrechen" wählen.

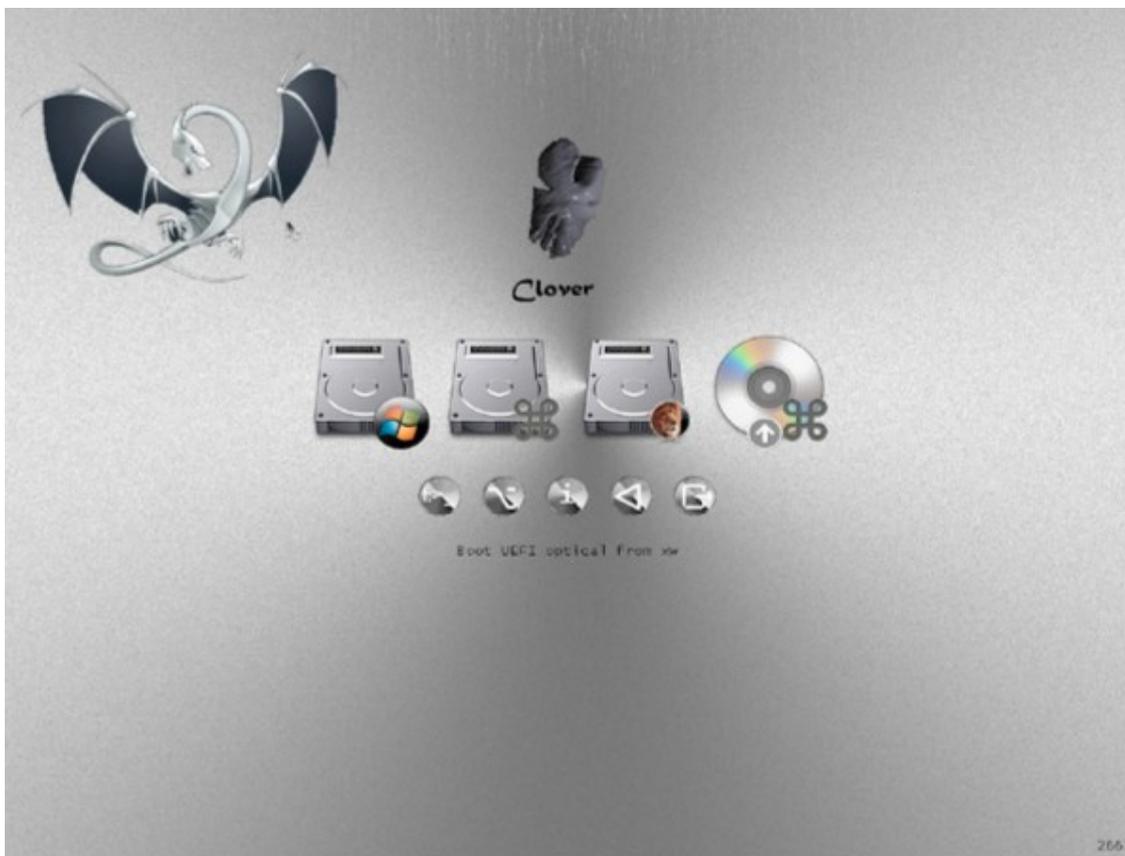
Wenn das System immer noch der Meinung ist, dass es aufwachen sollte, muss es in der Konfiguration einstellen:

```
<key>Boot</key>
<dict>
  <key>NeverHibernate</key>
  <true/>
</dict>
```

So verwenden Sie Clover

Die erste Begegnung

Booten Sie zunächst in die Clover-GUI und versuchen Sie, sich damit vertraut zu machen. Drücken Sie verschiedene Tasten, bewegen Sie die Maus herum.



Die obere Reihe der Schaltflächen sind die vorgesehenen Betriebssysteme, die gebootet werden können. In diesem Bild gibt es zwei davon, Lion und Windows, wie Sie auf den Bildern sehen können. Realistisch gesehen, erinnere ich sie daran, dass Clover kein Bootloader für das Betriebssystem ist, sondern ein Manager ihrer eigenen Bootloader. Speziell für den Mac ist der Bootloader `/System/Library/CoreServices/boot.efi`. Und In diesem Fall für Windows, `/EFI/microsoft/boot/bootmgfw.efi`.

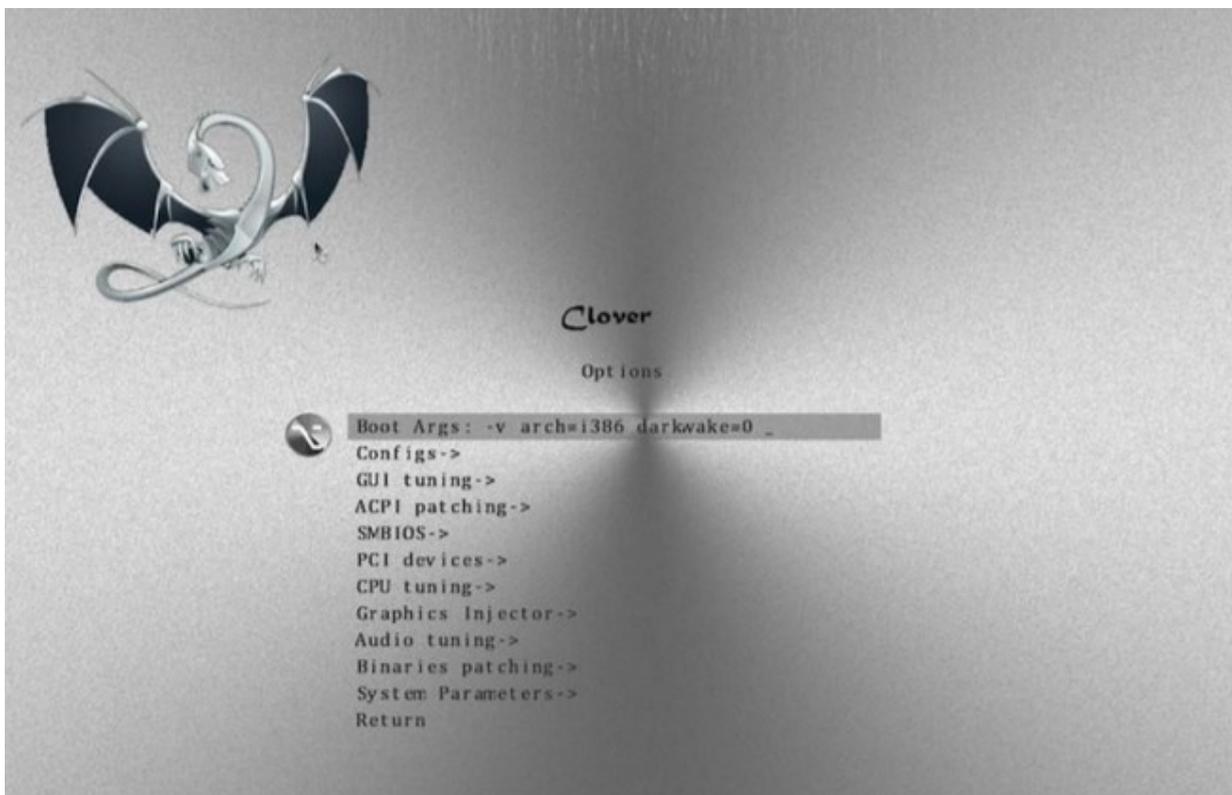
Untere Reihe der Schaltflächen - zusätzliche Funktionen: Befehlszeile (Shell), Menü Optionen, Informationen über den Lader und die Umgebung, starten Sie den Clover neu und beenden Sie ihn. Ausstieg wo? Zurück zur EFI-Umgebung, UEFI-BIOS bzw. im CloverEFI.

Es ist sehr nützlich, für die erste Einführung F1 zu drücken (wer hätte das gedacht?!). Wenn in der Konfiguration eingestellt

```
<key>GUI</key>  
<dict>  
  <key>Language</key>  
  <string>ru:0</string>
```

wird die Hilfe auf Russisch sein. Die Befehlszeile ist so etwas wie DOS, mit der Möglichkeit Dateien zu kopieren und zu löschen. Wie und wofür, das würde den Rahmen dieses Buches sprengen. **Dies ist Shell.efi mit seiner Hilfe(?)**.

In den Optionsmenüs können Sie einige Einstellungen ändern, die sich auf Ihren Fortschritt des Boot-Prozesses auswirken.

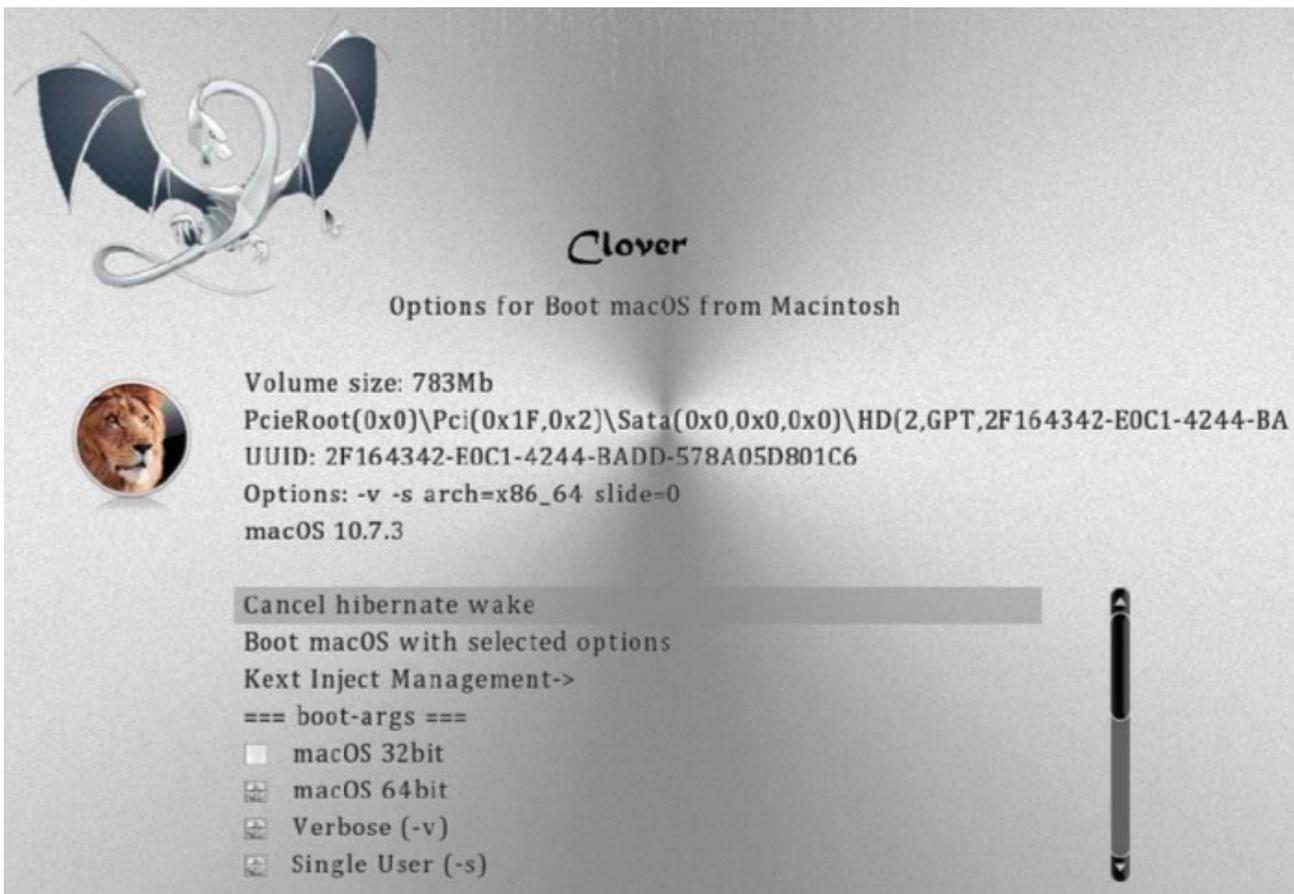


Einige von ihnen sind in der Datei config.plist angegeben, aber es könnte durchaus sein, dass dort falsche Einstellungen steht, und um diese Datei noch nicht zu bearbeiten, können die Einstellungen bereits hier im Clover-Menue korrigieren (bei einer unvollständigen Clover (Kleeblatt) config.plist).

Was genau zu ändern ist und wofür, das hängt von der Aufgabe ab, was sie genau erreichen wollen. Es gibt sehr nervige Anfragen wie "Ich habe eine xd Western Digital und Corsair-Speicher, helfen Sie mir beim Einrichten der config".

Config wird nicht durch die Hardware konfiguriert, sondern durch das Ergebnis. Wenn Sie nicht sofort booten, versuchen Sie festzustellen, was das Problem sein könnte, und beheben Sie es in diesem Menü.

Mehrere spezielle Boot-Methoden können durch Drücken der Leertaste beim entsprechenden Boot-Icon erreicht werden . (ein erneutes ENTER bewirkt – Systemstart, drücken der SPACE-Taste - Aufruf des Zusatzmenüs für das Booten



Insbesondere ist dies die einzige Möglichkeit, einen unerwünschten Hibernate (Überwinterung) zu verhindern.

Warum ist Clover so langsam beim Starten?

Einige können nicht einmal den Start abwarten und berichten, dass Clover nicht funktioniert. Schauen wir uns diese Frage einmal genauer an.

1. Das debug.log wird gesetzt:

```
<key>Boot</key>
<dict>
  <key>Debug</key>
  <true/>
```

Ja, es ist eine sehr wertvolle Information, um heraus zu bekommen, was nicht funktioniert und warum. Aber bedenken Sie, dass es den Startvorgang wirklich verlangsamt. Wenn Clover auf dem USB installiert ist (um es sozusagen auszuprobieren, und wir nicht wissen, ob Clover überhaupt arbeiten kann, oder?!), dann kann ein Debug-Startup 10 Minuten dauern. Ja, wirklich. Dies liegt daran, dass das Debug-Protokoll öffnet, schließt und es wird auf dem Flash-Laufwerk in jeder Zeile überschrieben. Damit ist sichergestellt, dass Sie Informationen über den Erfolg der Inbetriebnahme auch bei einem heftigen Bootproblem bekommen. Wenn sie also Debuggen wollen, warten sie! Oder setzen sie den Key auf <false/> für den ersten Versuch.

2. **Es gibt zu viele Festplatten, Partitionen und Dateien** auf ihrem System. Clover muss, um das Startmenü zu erstellen, alle Festplatten, alle Partitionen und alle Dateien auf ihnen durchforsten, um herauszufinden, welche Systeme zur Ausführung angeboten werden können. Nun, dann warten sie oder sie brechen den Scanvorgang ab,

```
<key>GUI</key>
<dict>
  <key>Scan</key>
  <dict>
    <key>Entries</key>
    <false/>
    <key>Legacy</key>
    <false/>
    <key>Tool</key>
    <false/>
  </dict>
</dict>
```

und stellen Sie ein Menü manuell zusammen

```
<key>GUI</key>
<dict>
  <key>Custom</key>
  <dict>
    <key>Entries</key>
    <array>
      <dict>
```

Zugegeben, es bedarf einiger geistiger Anstrengung, um herauszufinden, was man dort schreiben soll. Entweder Belassen Sie den Bootloader-Scan vorerst, um wenigstens irgendwo zu booten.

```
<key>Scan</key>
<dict>
  <key>Entries</key>
  <true/>
```

3. **Eine riesige Windows- oder Linux-Partition**, und vielleicht auch noch mehrere. Die Windows-Partition wird mit dem NTFS-Treiber gescannt. Und in dieser Partition gibt es normalerweise eine Million Dateien, und unter ihnen suchen wir nach bootmgr.efi. Ich würde empfehlen, Windows so zu installieren, dass sich diese Datei auf der EFI-Partition befindet, und den NTFS.efi-Treiber komplett zu entfernt und somit keine Windows-Partitionen gescannt werden kann. Ähnlich verhält es sich mit der Linux-Partition und dem Treiber VBoxExt2.efi.

4. **Zu viele Treiber im Ordner** /EFI/CLOVER/drivers/

Ich sehe eine Situation voraus, in der Leute anfangen, ihre eigenen Treiber dieses Typs zu produzieren und bereit sind, sie auszuprobieren. Überlegen Sie an dieser Stelle, ob Sie die zusätzlichen Treiber benötigen Tastatur, Maus, LowMemoryFix... Der Punkt ist, dass das Lesen all diese Dateien Zeit in Anspruch nehmen kann, wenn Sie von wenn Sie vom USB-Stick starten.

5. **Nicht unterstützte Maus.**

Leider werden nicht alle Mäuse von dem uns vorliegenden EFI-Treiber unterstützt. Schlecht kann es sein, dass sich die Maus auf dem Bildschirm nicht korrekt verhält, oder dass sie stark bremst. Verbieten Sie es zur Inspektion, oder sogar dauerhaft, wenn es falsch ist

```
<key>Mouse</key>
<dict>
  <key>Aktiviert</key>
  <false/>
  <key>Geschwindigkeit</key>
  <integer>0</integer>
</dict>
```

6. **Langsamer HFS+-Treiber.**

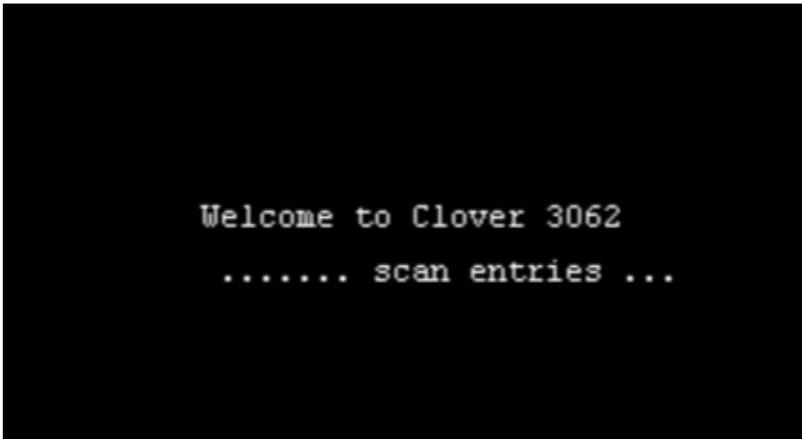
Der offizielle Clover kommt mit dem Treiber VBoxHFS.efi, der sauber lizenziert ist und Links versteht, aber langsamer ist als Apples HFSplus.efi. Laden Sie diesen inoffiziellen, aber schnellen Treiber von irgendwo herunter und legen Sie ihn in /EFI/CLOVER/drivers/UEFI/ ab. Dies gilt auch für das Booten von Legacy-Geräten (.drivers/BIOS/).

7. Es wurde ein ungeheuer schönes Designthema gewählt. Je reichhaltiger die Farben und Animationen des Themes sind, desto länger dauert es, bis es geladen ist. Wählen Sie eines der eingebauten Themen, sie sind die Schnellsten.

```
<key>GUI</key>
<dict>
  <key>Theme</key>
  <string>embedded</string>
```

8. Verwenden Sie die neuesten Clover-Version.

Egal, was die guten Ratgeber flüstern, die neue Version ist besser als die alte, und sie behebt Insbesondere Bugs, die das Einfrieren des Clover verursachten. Und ab der Revision 3063 erschien Meldungen direkt auf dem Bildschirm über den Bootvorgang.



Die Beschriftung ist unscharf, weil die guten Schriften noch nicht geladen wurden, und das hat einige wenig schmeichelhafte Antworten auf die Frage "Wie entferne ich Bildunterschriften?" gegeben. Setzen Sie `timeout=0`, und es wird keine Schrift zu sehen sein. Aber sie sind sehr nützlich für den Anfang, viele neuen Benutzer haben erkannt, dass in Clover meistens alles funktioniert, nur langsam. Dafür ist dieses Kapitel gedacht. Oder sie machen es so:

```
<key>Boot</key>
<dict>
  <key>NoEarlyProgress</key>
  <true/>
```

Als nächstes wollen wir uns einige Techniken, spezielle Patches und Arbeitsmethoden ansehen, die nach dem Prinzip des umgekehrten Wörterbuchs zusammengesetzt sind. Es gibt ein Problem → hier ist die Lösung.

Analysieren Sie `Debug.log/Preboot.log`

Warum? Clover gibt Ihnen Informationen über die Hardware von Eigenschaften, die Sie normalerweise nicht zu sehen bekommen und es sie wird auch darüber informieren, wie es die Konfiguration handhabt, bevor Sie das System starten.

Wie man diese Log-Dateien bekommt? Lassen Sie mich gleich sagen, dass es dasselbe ist, mit dem Unterschied des Erfolgs und Geschwindigkeit. Um das `preboot.log` zu erhalten, gehen Sie in die Clover-Shell (GUI) und drücken Sie die F2-Taste. Die Datei `preboot.log` wird im Ordner `Clover EFI/CLOVER/misc/File` gespeichert. Die Datei endet mit der Phrase "Enter GUI", was durchaus Sinn macht. Wenn Sie jedoch das System booten, wird die Datei enden mit dem Start des Systems selbst, das ist alles, was hier möglich ist.

Wenn Clover sich aufhängt, hilft debug.log. Um ihn zu erhalten, setzen Sie ihn in der Konfiguration

```
<key>Boot</key>
<dict>
  <key>Debug</key>
  <true/>
```

In diesem Fall wird die Datei Zeile für Zeile erstellt, und es ist garantiert, dass Sie einen Punkt darin finden wo der Bootvorgang stoppt.

Beispiel:

```
222:890 0:001   DefaultIndex=-1 and MainMenu.EntryCount=7 222:892
0:001   SetScreenResolution: 1366x768 - already set 0:100 0:100
MemLog inited, TSC freq: 2591578780
0:100   0:000   CPU was calibrated with ACPI PM Timer
```

Hier sehen wir, dass nach der Meldung über die Änderung der Bildschirmauflösung ein neues Protokoll begonnen hat. Das heißt, dass sollte der Fehler im Clover-Text nach dieser Meldung und vor der nächsten Meldung gesucht werden, die nicht da war. Melden Sie diese Zeilen dem Entwickler, er wird danach suchen. Lassen Sie uns nun analysieren, was wir im Protokoll sehen.

```
MemLog inited, TSC freq: 2591583140
```

Das hier ist die anfängliche Taktrate des Prozessors. Standardmäßig startet der Prozessor mit einem Kern bei maximaler Nicht-Turbo-Frequenz. Wenn das nicht der Fall ist, sollten bei ihnen sofort die Alarm-Glocken läuten! Der Bootprozess wird nicht funktionieren. Solche Situationen sind selten, und wir behandeln jede einzelne davon durch die speziellen Änderungen der Clover-Konfiguration. (Die Figur ist nicht rund, und das zu Recht!) Es wird für die eingebauten Dienstprogramme, bei einfacher Bildschirmauflösung reichen.

Now is 14.7.2019, 8:47:7 (GMT)

Startdatum und -zeit sind in GMT angegeben. Die Ortszeit in Moskau wird +3 sein. Unterscheidung des alten Protokolls von dem neuen sind damit möglich.

```
0:100 0:000 Starting Clover revision: 5120 (master, commit dddceaae3) on American Megatrends EFI
```

Hier und weiter unten finden wir heraus, welche Version des Clover der Benutzer verwendet und ob er UEFI boot hat, wie in diesem Fall, oder legacy (das wäre dann Clover EFI).

```
=== [ Get Smbios ] =====
```

Als nächstes folgt die Dekodierung der Speichermodule, wie sie vom BIOS erkannt werden. In der Regel korrekt, also fügen wir es in die Konfiguration ein:

```
<key>SMBIOS</key>
<dict>
  <key>Trust</key>
  <true/>
```

Und wenn Sie die Werte nicht mögen, dann setzen sie false ein. Clover ermittelt dann selbständig dadurch, das SPD oder XMP gelesen wird. In SMBIOS werden die Werte auch aus SPD-Lesevorgängen entnommen, durch Algorithmen des BIOS. Wir haben gute Algorithmen, aber die sind universell, und das BIOS ist spezifisch für dieses Mainboard, daher ist nicht sicher, was genauer ist.

```
<key>Boot</key>
<dict>
  <key>XMPDetection</key>
  <string>-1</string>
```

Siehe das Kapitel zur Konfiguration.

```
Running on: 'Latitude E6430' with board '0H3MT5'
```

Und das ist die Information, wie Ihr Computer genannt wird.

```
=== [ GetCPUProperties ] =====
```

```
Hier sind Informationen über den Prozessor, die CPUID und der eigene Name
BrandString = Intel(R) Core(TM) i5-3320M CPU @ 2.60GHz
```

Und andere wichtige Eigenschaften:

MSR 0xE2 before Patch 1E008404

Sehen Sie die Acht? Es ist eine spleton? (**Blockchain?**), eine schreckliche Sünde für einen Mac. Wenn Sie können, Flashen Sie das BIOS auf ein ungesperrtes. Wenn nicht, ist es durchaus möglich, mit Patches zu leben:

```
<key>KernelAndKextPatches</key>
<dict>
  <key>KernelPm</key>
  <true/>
  <key>AppleIntelCPUPM</key>
  <true/>
</dict>
```

Clover macht dies theoretisch automatisch, aber es ist besser, wenn Sie es manuell tun.

Turbo: 31/31/31/33

Diese Zahlen bedeuten, dass mit einem Kern eine Frequenz von 3300MHz erreicht werden kann, mit zwei Kernen oder mehr nur 3100. Da der Mac alle Kerne aktiviert hat, ist es nicht nötig, die Behauptung aufzustellen, dass der Prozessor des Macs nicht auf die maximal mögliche Frequenz von 3300 übertaktet wird.

Ich habe jedoch eine weitere Zeile

```
The CPU not supported turbo
```

Es ist sehr wahrscheinlich, dass das BIOS die Verwendung von Turbofrequenzen deaktiviert hat. Wir müssen uns die Einstellungen des BIOS ansehen.

=== [GetDevices] ===

In diesem Abschnitt erfahren Sie, welche PCI (PCIe)-Geräte sich in dem Computer befinden, zusammen mit ihren Adressen und DeviceID/VendorID. Sehr hilfreich zum Herausfinden, ob es möglich ist, Treiber für dieses Gerät zu finden

```
0:100 0:000 PCI (00|00:00.00) : 8086 0154 class=060000 MCH
0:100 0:000 PCI (00|00:01.00) : 8086 0151 class=060400
0:100 0:000 PCI (00|00:02.00) : 8086 0166 class=030000 VideoCard
0:100 0:000 - GFX: Modell=Intel HD Graphics 4000 (Intel)
0:100 0:000 PCI (00|00:14.00) : 8086 1E31 class=0C0330 USB3.0
0:100 0:000 PCI (00|00:16.00) : 8086 1E3A class=078000 IMEI
0:100 0:000 PCI (00|00:16.01) : FFFF FFFF class=FFFFFFF
0:100 0:000 PCI (00|00:16.03) : 8086 1E3D class=070002 SerialPort
0:100 0:000 PCI (00|00:19.00) : 8086 1502 class=020000 LAN
0:100 0:000 - LAN: 0 Hersteller=Intel
0:100 0:000 PCI (00|00:1A.00) : 8086 1E2D class=0C0320 USB2.0
0:100 0:000 PCI (00|00:1B.00) : 8086 1E20 class=040300 HDA
0:100 0:000 PCI (00|00:1C.00) : 8086 1E10 class=060400
0:100 0:000 PCI (00|00:1C.01) : 8086 1E12 class=060400
0:100 0:000 PCI (00|03:00.00) : 14E4 4353 class=028000 WiFi
0:100 0:000 - WiFi: Hersteller=Broadcom
0:100 0:000 PCI (00|00:1C.02) : 8086 1E14 class=060400
0:100 0:000 PCI (00|00:1C.03) : 8086 1E16 class=060400
0:100 0:000 PCI (00|00:1C.05) : 8086 1E1A class=060400
0:100 0:000 PCI (00|0C:00.00) : 1217 8221 class=080501 SD-Leser
0:100 0:000 PCI (00|00:1D.00) : 8086 1E26 class=0C0320 USB2.0
0:100 0:000 PCI (00|00:1F.00) : 8086 1E55 class=060100 LPC
0:100 0:000 PCI (00|00:1F.02) : 8086 1E03 class=010601 SATA AHCI
0:100 0:000 PCI (00|00:1F.03) : 8086 1E22 class=0C0500 SMBUS
0:100 0:000 PCI (00|00:1F.06) : FFFF FFFF class=FFFFFFF
```

Einige Geräte sind auch auskommentiert.

FFFF bedeutet, dass das Gerät nicht aktiv ist, obwohl es irgendwie vorhanden ist.

Geräte der Klasse sind Video, Audio, USB2.0, USB3.0, LAN, WiFi und so weiter. Alle Transkripte sind aus PCI-Spezifikationen bekannt. Ich habe die Klassenentschlüsselungen in der rechten Spalte hinzugefügt, im Protokoll sind sie nicht vorhanden. Die Klasse 060400 ist eine Brücke, ein Steckplatz, in den ein Gerät eingesetzt werden kann. Hier sind keine USB-Geräte vorhanden. Clover ist noch nicht reif genug, um sie auch zu scannen. Und hier ist noch etwas, aus dem Protokoll eines anderen.

```
- GFX: Modell=GeForce GTX 760 Familie CE (Fermi)
```

Was sehen wir?! 760 muss Kepler sein! Warum sagt Clover, sie sei Fermi?

Clover hat Recht, Sie können ihn nicht täuschen, indem Sie das BIOS flashen, Informationen zur Videokartenfamilie wird nicht aus dem BIOS entnommen, sondern aus dem Betrieb des Videokerns.

```
0:128 0:027 EFI\CLOVER#\#.plist nicht mit Namen aus LoadOptions geladen: Nicht gefunden 0:143 0:015
EFI\CLOVER\config.plist loaded:
Suces
```

Diese Funktion von Clover wird von kaum jemandem genutzt. Die Sache ist die, dass im BIOS gibt es eine Stelle, an der der Name der config.plist-Datei geflasht werden kann, und so können Sie direkt im BIOS Wählen Sie, mit welcher Konfiguration Clover laden sollen. In diesem Fall wurde kein solcher Name gefunden, und daher wird die Standard config.plist geladen. (Und im Allgemeinen können Sie verschiedene config.plist haben, wenn Sie sich an die Anweisungen von Dmazar(2014) erinnern (überaus praktisch um verschiedene config.plist zu testen, sie müssen natürlich unterschiedlich benannt werden!)

Es gibt eine weitere Liste im Protokoll unten

```
0:346 0:003 === [ Gefundene Konfigurationsplists ] =====
0:403 0:057 - config0.plist
0:403 0:000 - config1.plist
0:403 0:000 - config.plist
```

Dies ist für das Clover-Menü, um die Konfiguration von dort aus zu ändern. Aber die Abschnitte Boot, GUI und KernelAndKextPatches wird nicht mehr geändert. Das ist hier zu spät!

```
0:143 0:000 === [ GetListOfThemes ] =====
0:162 0:018 - [00]: embedded
0:176 0:014 - [00]: random
0:177 0:001 - [00]: Metall
0:188 0:010 - [01]: BGM
0:223 0:035 - [02]: CÄSIUM
0:285 0:061 - [03]: METALL@2X
0:298 0:012 - [04]: Clovy
0:343 0:044 - [05]: BOOTCAMP
```

Es ist verständlich, eine Liste mit allem, was installiert ist. Es muss nur sichergestellt werden, dass in der Konfiguration ein Thema gewählt werden, das wirklich auch in Clover enthalten ist.

KextsToPatch: 13 requested

KernelToPatch: 1 requested

Eine kleine Liste, was für die Kext und Kernel-Patches konfiguriert ist. Wir betrachten eine andere Konfigurieren, und kritisieren, warum er bestimmte Patches gemacht hat.

```
0:749 0:002 === [ LoadDrivers ] =====
0:861 0:111 Laden von ApfsDriverLoader.efi status=Success
0:883 0:021 - driver needs connecting
0:885 0:002 Loading AudioDxe.efi status=Success
0:893 0:007 - driver needs connecting
0:895 0:002 Loading DataHubDxe.efi status=Success
0:917 0:022 Loading EnglishDxe.efi status=Success
0:926 0:009 Loading Fat.efi status=Success
0:935 0:008 - driver needs connecting
0:937 0:002 Loading FSInject.efi status=Success
0:944 0:007 Loading OsxAptioFix3Drv.efi status=Success
0:952 0:007 Loading SMCHelper.efi status=Success
0:959 0:007 Loading VBoxHfs.efi status=Success
0:966 0:007 - driver needs connecting
0:968 0:002 4 driver needs connecting ...
0:970 0:002 PlatformDriverOverrideProtocol not fund. Installing... Succes
0:974 0:004 APFS-driver loaded
0:978 0:003 Searching for invalid Disklo BY_DRIVER connects: not found, all ok
```

Und der nächste Kritikpunkt ist, warum der Anwender diese Treiber herunterlädt, und warum er nicht andere nimmt.

SetScreenResolution: 1366x768 – already set

Ich habe diese Auflösung für meinen Bildschirm eingestellt. Und sie ist erfolgreich eingestellt. Für diejenigen, die das nicht mögen, siehe das Kapitel Konfiguration.

```
4:481 0:002 === [ GetMacAddress ] =====  
4:561 0:080 MAC-Adresse von LAN #0= D4:BE:D9:6C:86:CD:
```

Clover kann die MAC-Adresse von fast jeder Netzwerkkarte lesen. Wir verwenden diese Informationen, um Ihren ROM-Variablenwert einzustellen. Bei einigen UEFI-BIOS funktioniert es nicht, suchen wir nach anderen Wegen.

```
=== [ ScanSPD ] ===
```

Überprüfen von Speichermodulen, wenn wir dem BIOS nicht trauen.

```
=== [ GetAcpiTablesList ] ===
```

Eine Liste der ACPI-Tabellen, die sich im BIOS befinden. Nützlich, wenn Sie etwas verwerfen wollen (drop) (Tropfen)

```
- [06]: SSDT CpuPm len=2850
```

Es hat sowohl die ID = CpuPm als auch die Länge 2850. Sie können verwerfen entweder auf diesen Weg oder auf anderen, je nach Einzigartigkeit.

```
=== [ GetUserSettings ] ===
```

einige selektive Informationen darüber, was genau in der Config eingestellt ist, besonders nützlich, wenn man die Protokolle für Lesen der anderer Leute überprüft.

```
=== [ ScanVolumes ] ===
```

Eine Liste von Volumes mit ihren Adressen und UUIDs. Ein Volume ist entweder eine Partition oder eine ganze Festplatte.

Es ist nützlich diesen Bericht sich anzusehen, wenn jemand meint : "Clover kann meine Festplatte nicht finden !".

```
=== [ InitTheme ] ===
```

Weitere Informationen zur erfolgreichen Erstellung einer grafischen Oberfläche mit dem gewählten Thema. Ich für meinen Teil sehe

```
OSIcon os_mav not parsed
```

d.h. mein gewähltes Theme hat kein Mavericks-Icon, es wird nur das normale Mac-Icon verwendet. Hier finden Sie auch Informationen über den Startsound, der vom Thema abhängt.

```
6:511 0:002 === [ Dump SMC keys from NVRAM ] =====  
6:570 0:059 found AppleSMC protocol  
6:584 0:014 Registered 17 SMC keys
```

In den meisten Fällen spielen die SMC-Keys am Anfang keine Rolle. Sie sind unbedingt erforderlich für FileVault2 und für Hibernation. Sie werden vom SMChelper.efi-Treiber und von Clovers Infrastruktur bereitgestellt. (Es gibt auch eine Option von VirtualSMC mit eigener Infrastruktur). Für eine Systeminstallation scheint es nicht erforderlich zu sein, aber... wer weiß!

```
=== [ ScanLoader ] ===
```

Und hier ist eine Liste, was von wo heruntergeladen werden kann. Und es gibt auch Informationen, wenn sich das System in einem Ruhezustand (hibernation) befindet.

```
=== [ GetEfiBootDeviceFromNvram ] ===
```

Der Erfolg dieser Operation bestimmt, ob Clover per Timeout automatisch startet. Siehe Anweisungen mit dem entsprechenden Kapitel.

So sieht Erfolg aus

```
Boot redirected to Entry 3. 'Boot macOS from HighHD'
```

Das heißt, ich habe das System Timeout beim Booten vom HighHD-Laufwerk.

```
=== [ StartLoader ] ===
```

Startet das System um zu booten

```
GetOSVersion: 10.13.6 (17G7024)
```

Um zu sehen, welches System der Benutzer lädt.

Als nächstes folgt die Information, was genau Clover macht, bevor das ausgewählte System geladen wird, welche Patches, welche Eigenschaften erzeugt werden, welche Prüfsummen geladen werden, und die letzte Zeile, entsperren Sie ggf. USB2.0.

USB EHCI Ownership for device 1E26 value=1000001

In Version 5120 wird das Protokoll mit Werten aus den Kernel-Patch-Prozeduren und den Kext fortgesetzt. Dies ist für Entwickler bestimmt.

Ausführen von OSX auf nicht unterstützter Hardware

Eigentlich geht es im ganzen Buch um DAS. Ich erzähle Ihnen hier einen Teil davon, basierend auf der Frage. **Nicht unterstütztes BIOS.** Darauf können Sie wetten! Es ist der Hackintosh, über den wir reden. Und zunächst einmal sind es die Daten im DMI, die den Namen des Herstellers enthalten (es muss Apple.inc sein.), die Modell- und Seriennummer, deren Zahlen und Buchstaben nicht zufällig sind, sondern etwas bedeuten, insbesondere das Modell und das Herstellungsdatum. In der einfachsten Version, seit den Tagen von Netcas, Modell, das jeder verwendet, um MacPro3,1, und eine bestimmte Serie, eine für jeden, das funktioniert. Jetzt, nachdem Clover die Hardware (das Bügeleisen) analysiert hat, bietet er Dutzende von Optionen an, die praktikabel sind. Trotzdem ist es empfehlenswert, die Seriennummern zu generieren, oder vielleicht ein anderes Modell zu nehmen als Standardmodell.

Nicht unterstützter Prozessor. Ja, verschiedene Versionen von MacOS unterstützen unterschiedliche CPU-Kits, und Ihre CPU wird möglicherweise nicht unterstützt.

Hier ist eine solche Tabelle für die alten Systeme:

CPU-Name	CPUID	10.4	10.5.8	10.6.3	10.6.8	10.7.2	10.7.5	10.8.5	10.9.5
Yonah	0x0006E6	1	1	1	1	1	1	0	0
Conroe	0x0006F2	1	1	1	1	1	1	1	1
Penryn	0x010676	0	1	1	1	1	1	1	1
Nehalem	0x0106A2	0	1	1	1	1	1	1	1
Atom	0x0106C2	0	0	0	0	0	0	0	0
XeonMP	0x0106D0	0	0	0	0	1	0	0	0
0									0
Linnfeld	0x0106E0	0	0	1	1	1	1	1	1
Havendale	0x0106F0	0	0	1	1	1	1	1	1
Clarkdale	0x020650	0	0	0	0	1	1	1	1
AtomSandy	0x020660	0	0	0	0	0	0	0	
Lincroft	0x020670	0	0	0	0	0	0	0	
SandyBridge	0x0206A0	0	0	0	0	1	1	1	1
Westmere	0x0206C0	0	0	0	0	1	1	1	1
Jaketown	0x0206D0	0	0	0	0	1	1	1	1
NehalemEx	0x0206E0	0	0	1	1	1	1	1	1
WestmereEx	0x0206F0	0	0	0	0	1	1	1	1
Atom2000	0x030660	0	0	0	0	0	0	0	
IvyBridge	0x0306A0	0	0	0	0	0	0	1	1
Haswell	0x0306C0	0	0	0	0	0	1	1	
IvyBridgeE5	0x0306E0	0	0	0	0	0	0	1	
HaswellMB	0x0306F0	0	0	0	0	0	1	1	
HaswellULT	0x040650	0	0	0	0	0	0	1	1
CrystalWell	0x040660	0	0	0	0	0	0	1	1
usw.									

Das heißt, die Unterstützung für Yonah und XeonMP wird eingestellt; je neuer der Prozessor, desto neueres System ist erforderlich; Atom wurde nie unterstützt, obwohl er wie ein normaler Intel-Prozessor aussieht.

Die Tabelle ist veraltet, schauen Sie sich die XNU-Quellen an. Skylake wird z.B. ab 10.11.6 unterstützt...

Wenn Sie das System auf einem nicht unterstützten Prozessor laufen lassen, erhalten Sie einen Core Panic. **Der Patch KernelCpu=true wird verwendet, um dies zu verhindern. Es ersetzt einfach den Panikaufruf durch leeren Operator, und alles funktioniert weiter. Wie korrekt ist es?** Nun, zumindest funktioniert es! Bei der neuen Revisionen von Clover, habe ich den Patch FakeCPUID=0x010676 gemacht. Oder andere Zahlen, die für Ihr System geeignet sind und in der Nähe Ihres Prozessors (etwa die gleiche Generationen, z. B. bei Atom ist es wert, Penryn zu ersetzen, oder sogar Conroy). Substitution erfolgt im Kernel auf der Ebene des Aufrufs der Prozedur get_cpu_info() und hat somit Einfluss auf die Caches, die Informationen für die CPU bereit halten, anstatt die CPUID selbst aufzurufen. So funktioniert zum Beispiel AppleIntelCPUPowerManagement.kext darauf, und dieser Patch funktioniert für ihn.

Beispiel:

```
<key>KernelAndKextPatches</key>
<dict>
  <key>FakeCPUID</key>
  <string>0x010676</string>
```

Nicht unterstützte Grafikkarte.

Intel. Unterstützt: GMA950, X3100, HD3000 und höher. Leider helfen keine Ersetzungen. Es gibt einen unterschiedlichen Satz von Patches für jede Option, und wenn die Grafikkarte unterschiedlich ist. Im besten Fall haben Sie ein Bild, bei dem Sie die Auflösung aber nicht ändern können, und keine Grafikkbeschleunigung. Es ist möglich, im Prinzip damit zu leben, aber ich bin nicht damit zufrieden, das die Bildschirmfarbe nicht zu kalibrieren ist. Denn selbst mit Fotos auf einem Computer wie diesem, kann ich nicht arbeiten, es ist unerträglich. HD4000 und höher werden im System ab 10.14 unterstützt. Aber HD4000 unterstützt nicht 10 Bit/Farbe, jedoch bei den Skylake HD530 ist es bereits dabei!

Nvidia. Die 7300-7600-Karten werden nur bis zum System 10.7.5 im 32-Bit-Modus unterstützt. Es ist wahrscheinlich sinnlos, über ältere Karten zu sprechen. Es gibt einige Fragen zu den Fermi-Karten der Serie 4xx/5xx. Diese werden teilweise unterstützt, und zwar nur bis zum System 10.11.6. Im Falle von Nvidia prüfen sie auch den AppleGPUPowerManagement-Test, er enthält möglicherweise auch Ihre ID oder die einer ähnlichen Karte.

Für Systeme ab 10.12 funktioniert nur **Kepler**, GK-Familie nativ. Für neuere Karten benötigen Sie den WEB-Treiber, den es nur vor System 10.13.6 gibt. Das sind GTX 6xx-7xx Karten, aber nicht alle, darunter sind auch Fermi, dann ist Schluss.

Nvidia benötigt einen MacModel-Trick oder eine BoardID-Substitution, um zu funktionieren. Für Systeme ab 10.14 gibt es keine Web-Treiber, d.h. Maxwells und Pascals fallen komplett aus dem Rahmen. Auf der anderen Seite lässt jemand alte Tesla's in Mojave laufen, anscheinend hat Apple ein Schlupfloch für die alten Rechner gelassen.

ATI/AMD. Das ist eine lange Geschichte. Und darüber, wie ich zu einer Radeon9000IGP gekommen bin, und über dong cxt für X1500, und über Callisto cxt, und über komplizierte Rezepte zum Patchen von Steckern für moderne Karten, können sie in diesem Buch im Details nachlesen. Es wurde schon viel für Radeons getan, suchen, lesen, nicht dumm sein! Auf Systemen ab 10.13 funktionieren die Radeons der 6000er Serie aber darunter nicht. Sie haben keine Metal-Unterstützung, und daher sind die Treiber im System unvollständig oder gar nicht enthalten.

Nun hier gibt es WhateverGreen.kext von vit9696, der - "Treiber aller Grafikkarten, setzen Sie einfach die neueste Version ein, sowie die neueste Version von Lilu, und machen Sie sich weiter keine Gedanken. In Clover müssen Sie alles deaktivieren, was mit Grafikkarten zu tun hat".

Aber ich spiele nicht auf diese Weise! Wer es nicht herausfinden will, soll es bitte so machen. Für andere nehmen wir Schritt für Schritt auseinander, was es braucht, um eine Grafikkarte aufzuziehen.

Die 5700-Karten funktionieren erst ab Catalina. Für Mojave z.B. ist die Liste der möglichen Karten 550-590 und Vega.

Die 5700-Karten funktionieren erst ab Catalina. Für Mojave, zum Beispiel, ist die Liste der mögliche Karten: 550-590 und Vega.

Soundkarte. Professionelle Karten haben in der Regel Treiber für Mac. HDA-Standard-Chipsatz-Codex werden alle mit VoodooHDA Kext unterstützt. Der originale AppleHDA-Codex-Treiber wird von keinem auf dem Markt befindlichen Codex unterstützt. Es gab mal ALC885, aber das ist heutzutage nicht mehr üblich. Aber die Hacker haben eine Patching-Technik entwickelt für AppleHDA, so dass so gut wie jeder Realtec-Chip unterstützt wird, den Sie benötigen (d.h. ALCxxx). Clover hilft bei der Korrektur von DSDT für diesen Kext und schlägt Möglichkeiten zum Patchen vor. Lesen Sie in den Foren, was zu patchen ist und wie man es patcht. HDMI Audio arbeitet mit den in Clover bereitgestellten DSDT-Patches, funktionieren jedoch bei einigen Karten von AMD nicht. Aber in Systemen 10.13+ gibt es einen neuen Treiber AppleGFXHDA.kext, studieren Sie ihn!

Netzwerkkarte. Erstens unterstützen die Treiber von Apple eine Reihe von Chips.- Zweitens haben die Programmierer gelernt, wie man einen Kext für Netzwerkkarten schreibt, und es gibt Treiber für die meisten bekannten Karten. In manchen Fällen reicht es aus, eine FakeID für die Karte zu erstellen, um es auf die Liste der von nativen Treibern unterstützten zu bekommen, aber in den meisten Fällen benötigen Sie einen separaten Kext.

WiFi. An dieser Stelle wird es wirklich traurig. Einige Broadcom, Atheros und Ralink werden unterstützt. In den Foren finden Sie Informationen zu den einzelnen Modellen. **Intel-Chips werden im allgemeinen auf keinen Fall unterstützt** (aber s. u.). Clover kann bei FakeID helfen, zum Beispiel bei meiner Variante der Substitution von Broadcom 4315 damit sie durch 4312 unterstützt wird. Sowie Atheros mit angrenzenden Nummern.

Seit 2020 hat es einen Treiber für einige Intel-WiFi-Karten online. Die Chancen stehen gut!

Kexte blockieren (Block cupcake (Cact-Schloss?))

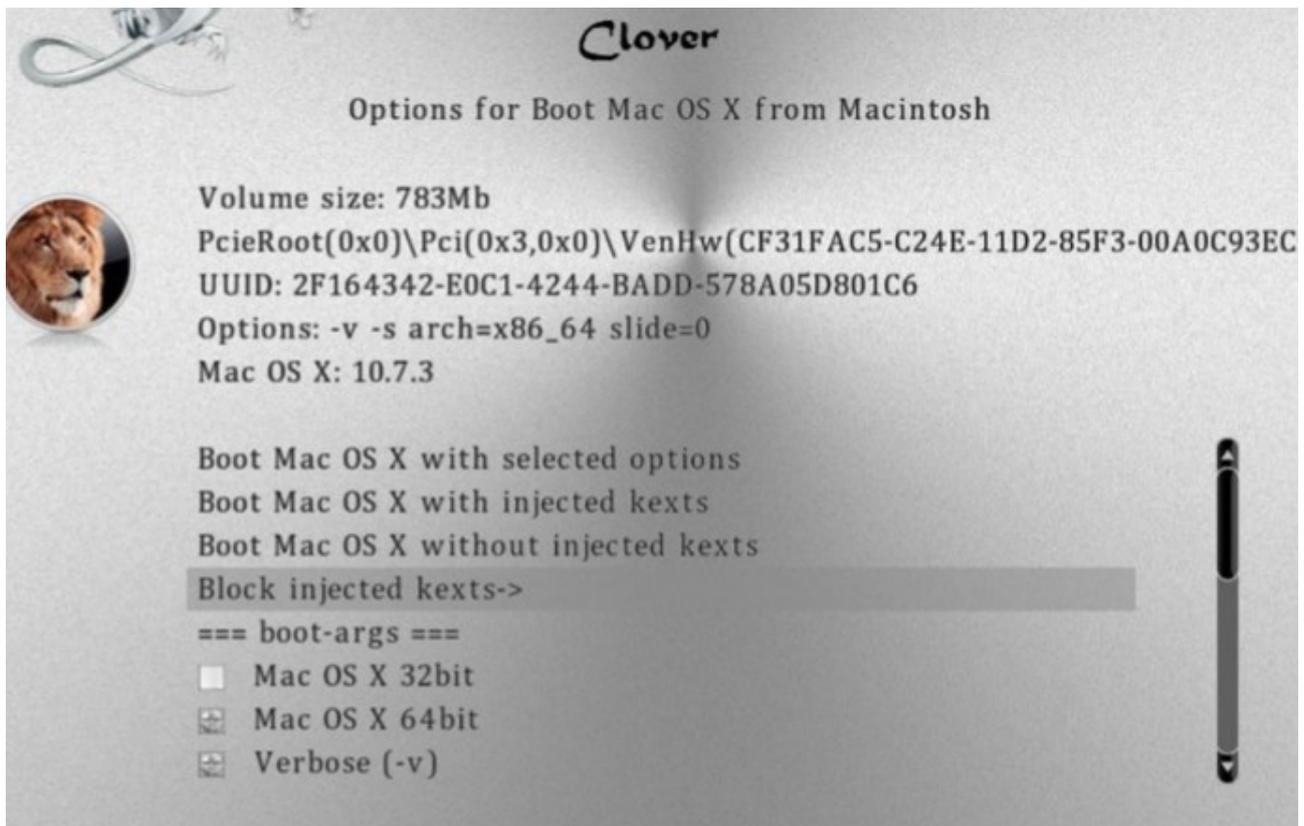
Beispiel: Ich installiere zufällig Geenna.kext in den SLE-Systemordner. Auf dem Bildschirm kommt es zu panic, nach dem Neustart wird zuerst dieser Kext geladen, und dann panic. So. Was machen wir jetzt? Es muss deinstalliert werden, aber es befindet sich noch kein anderes System auf diesem Computer. Für diese Situation wurde in Clover eine zusätzliche Funktion eingeführt: Im Optionsmenü in der dritten Zeile Block kext: Geenna eingeben und das System sicher im single user mode (Einzelbenutzermodus) booten (im Clover-Menü auf die Leerzeichen-Taste drücken und auswählen). Der Kext hat keine Zeit zum Laden, weil er blockiert ist. Wenn das System gestartet ist, im Terminal folgendes ausführen:

```
fsck -fy
mount -uw /
rm -r -v /S*/L*/Ex*/Geenna.kext
```

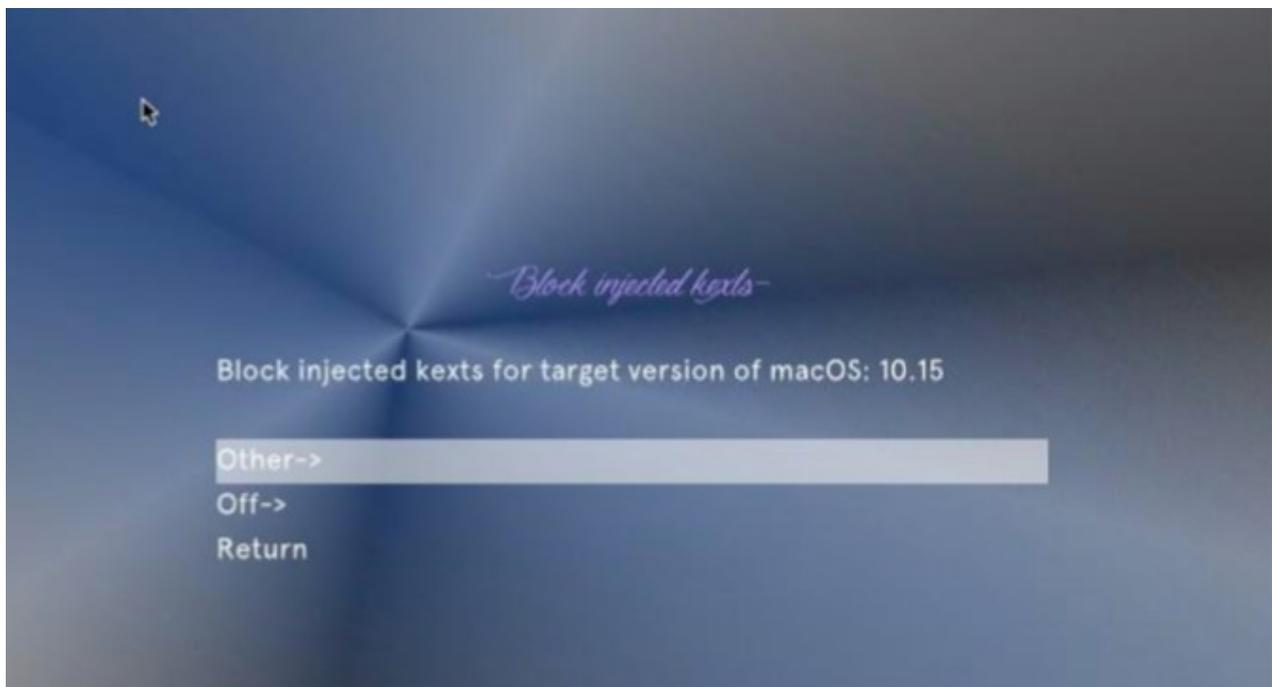
Dann können sie einen Neustart machen.

Hier ist ein Neustart erforderlich, da der Kernel sonst im nächsten Schritt ohnehin diesen Kext lädt, und es wird wieder Panik geben.

Aber es gibt eine neue Technik im neuen Clover: wir legen alle unsere Kext in einem Ordner ab EFI/CLOVER/kexts/Other. Und im Clover-Menü Details (Leertaste drücken) finden wir ein Untermenü von lock/unlock des Kexts.



Ab Revision 5052 ist es nun möglich, nicht benötigte Kexte im Ordner Off zu speichern, und Sie können sie bei Bedarf im gleichen Menü verbinden



Gleichzeitig sind die Kontrollkästchen im Ordner "Other" standardmäßig aktiviert, im Ordner "Off" sind sie jedoch deaktiviert, sie sind aber für Clover für die Konnektivität sichtbar.

Steckplatzname (AAPL,Steckplatz-Name)

Meistens handelt es sich um Kosmetika, obwohl behauptet wird, dass es in einigen Fällen ein Muss ist. Woher bekommt das System den Namen des Steckplatzes? Das alte System versuchte es mit _DSM zu injizieren, Eigenschaft "AAPL,Slot-Name", aber es ist die absolut falsche Methode, weil sie die Folge von statt der Ursache ist. Diese Eigenschaft wird vom AppleSMBIOS-System Kext auf Basis von ACPI-Eigenschaften _SUN und DMI-Tabellen bereitgestellt. Das heißt, _SUN gibt eine ID im Bereich 0-255 an, mit der ist der SMBIOS-Tabellentyp 9 mit der entsprechenden ID, aus der der Steckplatzname und andere seiner Eigenschaften abgeleitet werden.

Siehe Kapitel zum Bestücken der Konfiguration, Abschnitt SMBIOS->Slots

HDMI-Audio

Alles, was Sie für AppleHDA benötigen, wird von Toleda untersucht, aber nicht jeder möchte sich auf die Suche nach seine Erklärungen in englischer Sprache begeben. Ich habe mit seiner Beteiligung DSDT-Patches erstellt, mit denen sie an sein Ergebnis heran kommen.

Grundsätzlich gibt es zwei Optionen für HDMI-Geräte:

1. Auf einer externen ATI- oder NVidia-Grafikkarte. Im System ist es als Soundgerät der Klasse HDA = 0x0403 aufgeführt und wird vom gleichen Soundtreiber bedient. Es ist nur notwendig, dass sowohl Grafikkarte als auch HDMI die gleiche Eigenschaft "hda-gfx=onboard-1" haben. Oder vielleicht auch nicht. Nicht unterstütztes Gerät.

2. Die Intel Embedded-Karte hat einen HDMI-Anschluss, aber es gibt kein solches Gerät, es verwendet den Ton vom Chipsatz HDA. In diesem Fall müssen Sie in der Konfiguration vorgeben

```
<key>Devices</key>
<dict>
  <key>UseIntelHDMI</key>
  <true/>
```

In diesem Fall wird der Sound von ATI oder NVidia zu "onboard-2". Für DSDT notwendige Korrekturen: FixDisplay_0100, FixHDA_8000, AddHDMI_8000000

3. Option, dass die Einbettung nur für IQSV verwendet wird. Dann bei (iMac18,3).

HDEF-Gerät hat die Eigenschaft No-hda-gfx

Es gibt nichts in IGPU (die Inline für IQSV)

GFX0 (das ist Radeon) hat hda-gfx="onboard-1"

Und in HDAU ist es das Gleiche.

Clover macht das noch nicht automatisch, verwenden Sie das Array Eigenschaften.

Beachten Sie, dass alle diese zusätzlichen Eigenschaften nur für AppleHDA benötigt werden. Der VoodooHDA-Treiber benötigt keine externen Hinweise.

Computer Starupsound

Dies ist eine Erfindung von Goldfish64. Er schrieb einen EFI-Treiber für HDA-Sound, und machte Dienstprogramme für Sound-Tuning, Testen und Erstellen von Sound-Codec-Dumps.

<https://github.com/Goldfish64/AudioPkg>

Aber es fügte den Ton im Moment des Starts von boot.efi ein, indem es dessen Aufruf durch den Bootloader abfing. Das heißt, die Idee, dass es nicht nur mit Clover, sondern mit jedem EFI-Bootloader funktionieren würde, ohne seine internen Codes. Und ich bin mehr daran interessiert, dass der Sound funktioniert, bevor ich die Clover-Oberfläche betrete, oder sogar während ich durch sein Menü gehe. Die Lizenz ist offen, so dass ich es nach Ihren Wünschen umgestaltet habe. Alle Einstellungen und Tests können von Clover selbst durchgeführt werden, mit seinem grafischen Menü, und ich habe den Treiber selbst in das Clover-Repository aufgenommen, damit er

nicht verloren geht, und damit es in Zukunft verbessert werden kann, ohne die Autoren zu fragen, die heute vielleicht nicht, aber morgen verschwinden könnten.

1. Damit der Ton funktioniert, müssen sie also Clover Revision 4871+ verwenden. Es funktionierte in früheren Revisionen, aber mit Fehlern.

2. Legen Sie den Treiber AudioDxe.efi in den Ordner EFI/CLOVER/drivers/BIOS oder drivers/UEFI beziehungsweise, oder beides. Dieser mit Clover mitgelieferte Treiber unterscheidet sich bereits von des Autors, noch nicht im Prinzip, aber ich würde genau meine Version empfehlen.

3. Legen Sie die Sounddateien mit den Namen sound.wav und sound_night.wav in die von Ihnen verwendeten Themen. Der Startton ist also abhängig vom gewählten Thema. sound_night.wav funktioniert nicht obligatorisch, wenn es nicht vorhanden ist, wird sound.wav auch nachts abgespielt. Diese Dateien müssen RIFF/WAV-Format, 2 Kanäle, 16bit little endian unsigned int, Abtastrate kann 8, 44.1, 48kHz sein, die Dateigröße hängt davon ab. Das Audio selbst unterstützt 44 und 48kHz, oder sogar mehr. Aus Gründen der Kompaktheit habe ich auch 8kHz zugelassen, und Clover wandelt solche Datei bei 48kHz zum speichern. Die Qualität ist zwangsläufig geringer, aber für ein Gehäuse wie dieses ist es nicht wirklich benötigt wird. Aber es war möglich, den Sound dieses Formats direkt in Clover zu packen, und es wird für Ausgangstest abgespielt.

4. Gehen Sie zur Clover-Oberfläche, Startup sound output → , und testen Sie, welche der Ausgänge werden die Wiedergabe bringen.



In der ersten Zeile stellen Sie die Lautstärke von 0 bis 100 ein. Dies sind Prozentsätze, nicht mehr als 100 geschieht. Ein Wert von 0 bedeutet, dass kein Ton abgespielt wird. Das heißt, es ist nicht etwas, das schreit mit geschlossenem Maul, was bedeutet, dass es keinen Versuch gibt, zu miauen. Übrigens, ich weiß nicht, ob die Skala linear oder logarithmisch.

Die folgenden Zeilen sind aus dem Chipmodell und dessen Ausgabe kombiniert. Wenn Sie mehr als eine Soundkarte haben, wie so oft, gibt es neben der eingebauten Soundkarte auch HDMI, dann sehen Sie alles in dieser Liste, mit allen ihren Ausgängen. Auswählen, F7 drücken zum testen (anhören). Nach dem Verlassen dieses Menü, wird die gewählte Einstellung im NVRAM gespeichert, auch die, die in

Clover.SoundVolume, Clover.SoundDevice, Clover.SoundIndex Variablen. Hier habe ich im Gegensatz zu der Version von Goldfish es so gelöst, daß Clover die Einstellungen in einem privaten Bereich speichert mit seiner UUID, was unmöglich für emulierten Speicher, für Legacy-Clover funktioniert. Meine Einstellungen werden sichtbar sein von Systeme, können aus dem System entfernt oder geändert werden, und das Clover-Präfix stellt sicher, dass es keinen Konflikt mit den Interessen von Apple gibt. An diesem Punkt wird Clover goldfish (Fisch)einstellungen lesen, wenn Sie noch keine eigenen haben, sondern nur Ihre eigenen in Zukunft verwenden werden.

Beim nächsten Neustart hören Sie den Ton, bevor die Clover-Shell geladen wird, aber nach scan entries . Diese Einstellungen werden nicht in der Config vorgenommen, das ist nicht nötig. In jedem Fall müssen Sie zuerst testen, was bedeutet, dass die Einstellungen bereits in das NVRAM geschrieben wurden. Aber es gibt eine Einstellung in der Konfiguration, die von der Testphase übrig geblieben ist: **PlayAsync=true.**

Wenn False gewählt wurde, wird, nichts funktionieren, während der Ton abgespielt wird. Dies scheint bei der echten der Fall zu sein. Wenn true, spielt der Ton im Hintergrund, ohne alles andere zu stören. Ich habe eine lange Tondatei, und hörte Musik. Die GUI von Clover tauchte auf, Musik spielt, ich wählte das Laden in Verbose-Form. Die Musik spielt. Ich drückte auf "System booten", und siehe da. Meldungen: boot.efi funktioniert, Musik spielt, Kernel gestartet, Musik läuft weiter zu spielen! Und erst nach ein paar geladenen Kexten wurde es still, es muss durch VoodooHDA passieren, wodurch der Soundchip neu initialisiert wurde. Im geladenen System war kein Problem festgestellt. In der Revision 4862 war es nicht möglich, asynchrones Audio zu verwenden, es hing. In der Revision 4871 ist das Problem gelöst und asynchrones Audio kann nun asynchron standardmäßig verwendet werden.

NVRAM, iMessage, Multiboot

Die Frage der Verwendung von nichtflüchtigem NVRAM-Speicher durch das System mit der GetVariable()- und SetVariable()-Funktionen tauchte bereits im Jahr 2010 auf:

<http://www.projectosx.com/forum/index.php?showtopic=1504>

Dann habe ich versucht, das Arbeiten damit in Chameleon in meinem eigenen System zu implementieren, aber nein Ich habe dabei keine Unterstützung bekommen. Keiner wollte es, obwohl mein Argument über den Test Das Panel "Bootable Disk" erhalten blieb. Dann erklärten mir die Gurus, dass dies im DUET-Lader sei. Als ich also das Clover-Projekt auf Basis des DUET startete, habe ich zunächst diese Funktionalität haben wollen.

Chameleon hat diese Funktionen, aber sie sind sehr einfach gemacht "Return Unsupported", also damit das System, das mit Chameleon läuft, nicht in Panik gerät und einfach nicht mehr antwortet wenn diese Funktionen aufgerufen wird. Dies funktionierte vorerst, mit Ausnahme des StartupDisk-Panels. Aber hier ist es der iMessage-Dienst, der sich bereits geweigert hat, auf diese Weise zu arbeiten. Keine Substitution und Emulation hat funktioniert. Ich verneige mich vor Meklort, der die Art und Weise, eine solche Funktionalität in Chameleon zu realisieren innerhalb eines Monats, unter Verwendung des Moduls FakeNVRAM.dylib und eine Mutter(?).

Was ist mit NVRAM-Betriebsfähigkeit gemeint? Wenn das System eine Variable bis zum nächsten Neustart speichern soll, schreibt es das in das NVRAM mit der Funktion SetVariable(...). Wir können unsere Variablen auch speichern mit nvrnm-Dienstprogramme:

```
sudo nvrnm MyVar=qu-qa-re-ku
```

nach einem Neustart, muss diese Variable im System bekannt sein, mit Hilfe des Befehl aus dem auslesen

```
nvrnm MyVar
```

Wie funktioniert dieser Service bei Clover?

1. EmuVariableDxe-Funktionen werden für das Laden von Altlasten verwendet. Dies ist natürlich nicht ein echter nichtflüchtiger Speicher, da der Legacy Clover Computer, die weder über einen solchen Speicher verfügen, noch über ein eigenes EFI mit den notwendigen

Dienstleistungen. Dieser Treiber schreibt Variablen einfach in den Speicher, aber dieser Speicher ist verfügbar für Verwendung von MacOSX in seiner nativen Oberfläche. Beim Herunterfahren des Systems wird das Skript rc.shutdown.local aufgerufen, das all diesen Speicher in der Datei nvram.plist in Wurzel des Systemlaufwerks speichert. Clover durchsucht diese Datei beim Starten und trägt alle Variablen von dort zurück in das RAM als emuliertes NVRAM. Die Methode ist unvollständig, denn nur die AppleBootGuid-Variablen werden auf diese Weise gespeichert, aber es ist ausreichend um das Startup-Laufwerk auszuwählen.

2. Für UEFI-Boot setzen wir auf unseren eigenen Dienst VariableDxe ein, der wird im OEM-UEFI bereitgestellt. In Revision 2837 hat Dmazar die Handhabung dieses Dienstes korrigiert, so dass es bei den meisten Anwendern jetzt auf ihre Weise funktioniert. Für diejenigen, die nicht funktionieren, wird die EmuVariableUEFI als Emulationstreiber bereitgestellt, der ähnlich wie Legacy-Treiber, auch Skripte und die Datei nvram.plist benötigt. Das sind die neuen Zeiten!

Wieder hat vit9696 den OsxAptioFix-Treiber korrigiert, so dass das Hardware-NVRAM funktioniert, aber auf den neueren 360, 390 Chipsätzen funktioniert das auch nicht. Die Änderung wird durch den Treiber OsxAptioFix3Dxe bereit gestellt, und vit9696 selbst bietet eine erweiterte Version von AptioMemoryFix an, die jetzt in das Clover-Repository aufgenommen wurde.

EmuVariable ist in beiden Fällen keine vollwertige Emulation, for example, panic persists.log (Panic.log wird gespeichert **oder doch nicht?**), einfach weil das Skript keine Zeit zur Ausführung hat. Das tut es nicht. auch nicht die Variable boot0082, die für den Ruhezustand benötigt wird, aber wir haben dieses Problem auf andere Weise umgangen. Aber das Vorhandensein von panic.log, ein langjähriger Traum von Hackintoshern, bleibt das Vorrecht von Clover mit echtem NVRAM. Und wieder erfordert der Winterschlaf in Mod 25 das Speichern des Chiffrierschlüssels online, d.h. nur mit echtem NVRAM.

iMessage ist Apples eigenes Instant-Messaging-System. Seit Dezember 2012 Jahren haben sich die Regeln für die Registrierung und Nutzung geändert, und alle hackintoshes wurden außen vor gelassen. Es hätte mit Clover funktioniert, wenn wir nicht im September den iCloud-Service aussortiert hätten. Wir haben einen Fehler bei der Anzahl der Ziffern gemacht, wir hätten 17 lassen sollen, aber wir haben 12 gelassen. Der Fehler wurde erst im Januar verstanden, und so merkten auch die Chamäleone, was los war, nur hatten sie keine NVRAM, ohne das dies alles nicht möglich war. Um sich nämlich erfolgreich für iMessage registrieren zu können, muss die ROM- und MLB-Variablen in das NVRAM geschrieben werden, die es für jeden Computer gibt, und der Computer wird durch seine HardwareUID identifiziert, die dementsprechend auch eindeutig sein muss. Für ganz Dumme habe ich die Generierung dieser Eigenschaften auf der Basis von DMI-Daten, als auch Empfehlungen zum Schreiben der entsprechenden Werte in die config.plist, für diejenigen, die ein wenig intelligenter sind. Gleichzeitig stellte sich heraus, dass der iMessage-Dienst kostenpflichtig ist, und der Benutzer muss seinen Appstore-Account registrieren, über den Apple \$1 berechnet, um zu überprüfen, ob das Bankkonto gültig ist. Es folgt außerdem die Notwendigkeit der Einzigartigkeit des Kontos. Es besteht keine Notwendigkeit, das ROM, die MLB und die UUID eines anderen Herstellers zu verwenden, geschweige denn die Bankkarte eines anderen. Wenn alles in Ordnung ist, hat ROM 12 Ziffern, MLB hat 17 Ziffern, die UUID ist ungleich Null, und sie ist eindeutig, das Konto ist mit einem gültigen Konto mit einem Guthaben verknüpft, für den funktioniert iMessage. Und hören Sie nicht auf irgendwelche Spekulationen über en0, Partitionsformatierung und dergleichen. Ich habe alle Bedingungen aufgelistet.

Bootable Disk ist ein Service, mit dem Sie in der Systemsteuerung angeben können, mit welchem System wir neu starten wollen, drücken Sie Neustart und lassen sie den Computer arbeiten.

Der Computer macht alles von selbst. Für diesen Dienst muss das Laufwerk im GPT-Format partitioniert sein. Dies ermöglicht um z. B. zwischen 10.9 und 10.7 zu wechseln. **Denken Sie an die allgemeine Regel: dynamische Daten haben Vorrang vor statischen Daten.** Daten aus dem NVRAM haben Vorrang vor Daten aus der config.plist.



Verwendung mehrerer Konfigurationen

Mögliches Problem: Sie haben mehrere Systeme, aber diese Systeme müssen mit einem anderen Satz von Patches booten, die in der Konfiguration vorgeschrieben sind, z. B. Definition des Framebuffer für die Radeon im neuen System ist nicht der gleiche wie im alten System. Aber wie macht man das, wenn die Konfiguration in Clover (im Kleeblatt) die gleiche ist? Ab Revision 3266 ist dies möglich.

Hier ist die Konfiguration.

```

<key>GUI</key>
<dict>
  <key>Custom</key>
  <dict>
    <key>Entries</key>
    <array>
      <dict>
        <key>FullTitle</key>
        <string>Lion special</string>
        <key>Settings</key>
        <string>config-spezial</string>
        <key>Volume</key>
        <string>EE9CCC69-EE7F-358F-B120-BCD07AD78282</string>
        <key>Subentries</key>
        <array>
          <dict>
            <key>FullTitle</key>
            <string>Boot Lion with own settings</string>
            <key>CommonSettings</key>
            <false/>
          </dict>
          <dict>
            <key>FullTitle</key>
            <string>Boot Lion with common settings</string>
            <key>CommonSettings</key>
            <true/>
          </dict>
        </array>
      </dict>
    </array>
  </dict>
</dict>
<dict>
  <key>FullTitle</key>
  <string>Lion default</string>
  <key>Volume</key>
  <string>EE9CCC69-EE7F-358F-B120-BCD07AD78282</string>
  <key>Typ</key>
  <string>OSX</string>
</dict>
</array>
</dict>

```

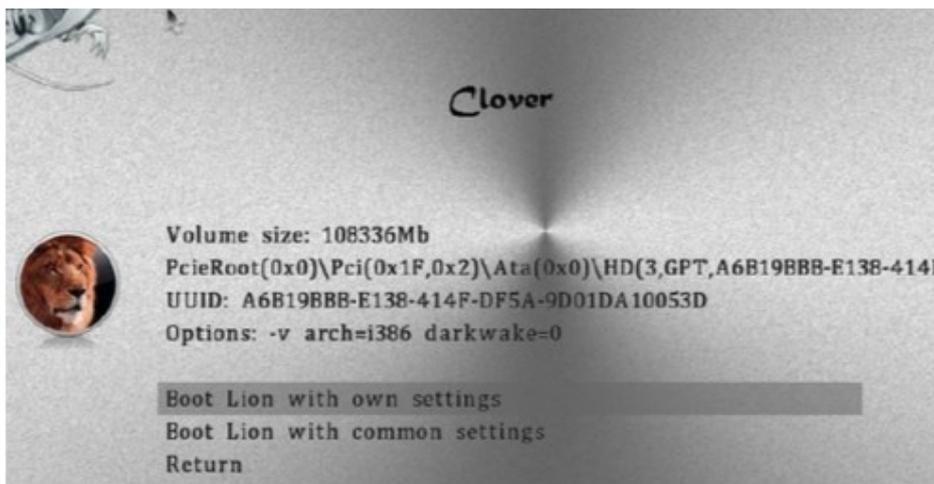
Hier wird folgendes beschrieben: Wir haben eigene Hauptmenüpunkte (Einträge) mit dem zweiten Punkt können Sie normalerweise die Last laden. Mit dem zweiten Punkt können Sie, wie üblich, die

System mit der allgemeinen config.plist, unter Berücksichtigung der im Optionsmenü des Clover vorgenommenen Änderungen.

Der erste Punkt erstellt ein neues Symbol für dasselbe System, aber es wird mit einem anderen durch config-special.plist, wie im Schlüssel Settings angegeben.

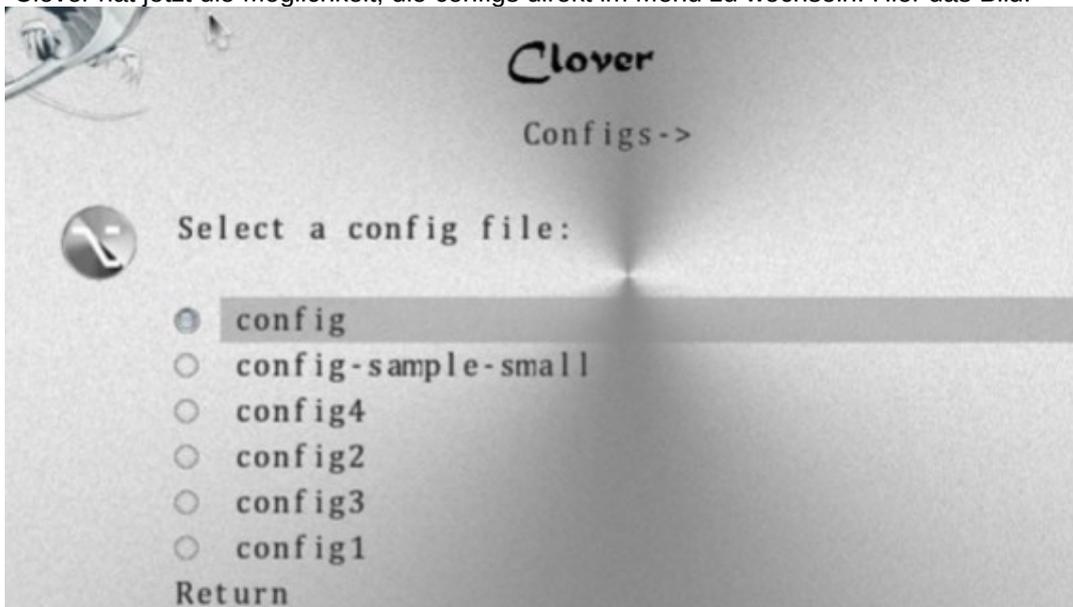


Aber das ist noch nicht alles. Durch Drücken der Leertaste gelangen wir in das Startmenü, und hier finden wir unsere Eingaben, als SubEntries geschrieben



Vergleichen Sie die Konfiguration mit den obigen Bildern, um zu klären, was vor sich geht. Die zweite Option in diesem Menü bedeutet, dass die spezielle Konfiguration nicht verwendet werden soll, aber die allgemeine. Es ist klar, dass, da die spezielle Konfiguration sich anschließt, nachdem Clover läuft, die Boot- und GUI-Partitionen werden darin nicht mehr benötigt, diese können nur noch in der allgemeinen Konfiguration sein. Ich, persönlich, möchte diese Konfigurationen nutzt, um neue Konfigurationen zu testen, um den Kapitän zu betreiben, nachdem eine bewährte funktioniert. Die funktionierende ist speziell, und die allgemeine Konfiguration ist experimentell, da die allgemeine Konfiguration über das Menü geändert werden kann, während die spezielle Konfiguration so verwendet wird, wie sie ist.

Clover hat jetzt die Möglichkeit, die configs direkt im Menü zu wechseln. Hier das Bild:



Eine Einschränkung dieser Methode ist, dass der Boot-Bereich nicht geändert wird. Es ändert auch nicht die KernelAndKextPatches. Die neue Konfiguration wird wirksam, nachdem Sie das Hauptmenü verlassen haben.

Es ist verwirrend, aber warum sollte man die Konfiguration überhaupt ändern, wenn nicht für KextPatches? Leider wenn etwas nicht funktioniert, liegt der Grund irgendwo tief in den Algorithmen von Clover. Auf der anderen Seite, schreiben Sie alle Patches in einer Konfiguration, der Hauptkonfiguration, und in der Clover-Oberfläche können Sie sie durch an- oder abhaken auswählen. Der einzige Vorteil der verschiedenen Konfigurationen sind unterschiedliche SMBIOS-Abschnitte.

How to make boot.efi not too spamming on the screen?

Gehen sie in dieses Menü:



und schreiben Sie dort log=0. Andere Werte sind möglich, untersucht von vit9696

log=value, output direction (Ausgaberichtung)

1 - AppleLoggingConOutOrErrSet/AppleLoggingConOutOrErrPrint (klassisches ConOut (Konsolenausgabe) oder StdErr (Standarderror bei Fehler) bei Ausfall)

2 -AppleLoggingStdErrSet/AppleLoggingStdErrPrint (**StdErr oder serial?**)

4 -AppleLoggingFileSet/AppleLoggingFilePrint (BOOTER.LOG/BOOTER.OLD-Datei auf der EFI Partition)

debug=value

1 - ermöglicht es, etwas in den BOOTER.LOG zu schreiben (gestrippter Code impliziert, dass es einen Absturz geben haben könnte)

2 - aktiviert die Perf-Protokollierung nach /efi/debug-log im **device drei(?)**

4 - aktiviert den Timestemp Ausgabe für **gestylte printf-Aufrufe (?)**

level=value - error rate (**Fehlerlevel?**)

kc-read-size=value - Log-Größe

Für unsere Zwecke ist log=0, ausreichend, die Voreinstellung in Clover.

FAQ

Häufig gestellte Fragen.

Frage (Fragesteller): Ich möchte Clover ausprobieren, wo soll ich anfangen?

O. (Antwort:) Mit dem Lesen dieses Buches.

Zusatz. Es ist seltsam, dies in ein Buch zu schreiben, aber vielleicht werden diese FAQs außerhalb dieser Seiten landen.

Fragesteller: Welche Version von Clover passt am besten zu meiner Hardware?

Antwort: Der letzte. Das steht nicht einmal zur Diskussion.

ZZY. Hier ist ein Buglog über einige Überarbeitungen, was endlich in der jeweiligen Version behoben ist:

3514: DDR4-Unterstützung

3471: Globaler Fehler bei Verwendung von va_args

3362: SMBYOS-Fehler bei duplizierten Zeilen im Original

3358: Die Berechnung der Anzahl der Kerne für viele Xeons wurde korrigiert

3336: Fehler mit Regionfix behoben

3333: neue Prozessoren hinzugefügt

3259: Kernel-Patch, um das Hochladen von cKexts auf ElCapitan zu ermöglichen

3168: Config-Änderung im Menü wurde von Clover nicht akzeptiert

3164: Der IDE-Treiber arbeitet nun korrekt im UDMA-Modus.

3162: XHCI-Treiber korrigiert, um Legacy- und Enable-Ports zu deaktivieren.

3157: Erhöhte AHCI-Geschwindigkeit um ein Vielfaches.

3154: Schleifenbildung auf dem InfoPlist-Patch verhindern.

3147: Fehlerkorrekturen im BiosBlockIO-Treiber.

3144: Fehler beim Lesen von Config und Setup aus dem Menü behoben, Abhängigkeiten.

3138: Fehler beim Windows-Start.

3128: Fehler beim Lesen von SPD.

3121: Starten von exFAT-Partition.

3116: eine Reihe von Korrekturen im Installationsprogramm.

3100: Patch Yosi Kernel-Fähigkeit.

3090: VBoxHFS.efi-Fehler liest die falsche Datei.

3086: Fehler beim Lesen von Mac-Adressen hängt auf neuem Chipsatz.

3074: Die USB-Legacy-Unterstützung hängt. Anfangs bekannt, aber Lösungswege variieren Die Entwickler waren direkt gegensätzlich und inkompatibel.

3060: AHCI-Treiber. Der Patch kam von seinen Schöpfern, Intel.

3057: Adresstabellenüberschneidung und EBDA verursachen Startfehler bei einigen BIOSen.

3053: Prozeduren, die VA_ARG-Makros verwenden, müssen EFI-API haben, sonst ist es möglich Bugs im Betrieb. Und wirklich beobachtet! Patch von Intel.

3041: Neue Grafikkarteninjektion hinzugefügt.

3036: Der Kernel-Patch für 10.10 wurde von Rehabman korrigiert.

3035: Der DSDT-Patch wurde korrigiert, der zum Einfrieren führte.

Und so weiter... Dabei sind die Korrekturen an den Kompilierungs- und Installationsprozessen, an der Kosmetik und Design sowie die Unterstützung für neue Hardware und neue Betriebssysteme.

Fragesteller: Es funktioniert nicht.

Antwort: Sie sind ein Narr.

Nun, was gibt es sonst noch zu sagen?

Fragesteller: Ich habe Clover installiert, aber ich erhalte einen schwarzen Bildschirm.

O. Das Booten des Betriebssystems erfolgt in acht Stufen (siehe Seite 6). Könnten Sie bitte genau angeben, in welchem Stadium der Stopp auftritt. Und stellen Sie sicher, dass Sie in Ihrem Bericht angeben "Installiert mit der Auswahl dieser Optionen durch das Installationsprogramm". Dann werden wir ein Gespräch führen.

Die häufigsten Fehler sind:

- CsmVideoDxe funktioniert nicht mit einigen BIOSen, deinstallieren Sie es;
- Es gibt Zeiten, in denen der Patch VBIOS=Yes einen schwarzen Bildschirm verursacht, versuchen Sie, es zu deaktivieren,
- steht Boot->Debug=true. Alles funktioniert, aber langsam, nicht genug Geduld zum Warten.

Um besser diagnostizieren zu können, was vor sich geht, setzen Sie:

```
<key>Boot</key>
<dict>
  <key>Debug</key>
  <true/>
```

in die Datei config.plist. Das Laden wird sehr langsam sein, da bei jedem Schritt /EFI/CLOVER/misc/debug.log aktualisiert wird, aber nach dem endgültigen Aufhängen wird man Informationen darüber erhalten, was genau passiert ist. Realistischerweise kann es beim Booten von einem USB-Stick etwa zehn Minuten zu gehen, bevor die GUI aufrufen wird. Ab der Revision 3063 ist der Bildschirm nicht mehr schwarz, wenn die CloverGUI zu booten begann, sehen Sie eine Beschriftung auf dem Bildschirm, die Sie darüber informiert, was genau vor sich geht.

Fragesteller: Ich sehe eine 6_ auf dem Bildschirm und sonst passiert nichts.

Antwort: Dies ist der schwerwiegendste Fall von Hardware-Inkompatibilität. Es ist jetzt nicht üblich, außer bei AMD-Prozessoren. Nur ein Programmierer kann das Problem diagnostizieren und Debug-Meldungen in die Clover-Codes einfügen und einen Neustart nach dem anderen durchführen, bis das Problem herausgefunden ist. Leider kann man dem normalen Anwender hier nichts weiterhelfen. Lesen Sie das Kapitel über den langsamen Clover, vielleicht warten?

Es sei denn, Sie spielen mit den BIOS-Einstellungen herum, dann hilft es manchmal. Versuchen Sie, boot7 (Clover BiosBlockIO) anstelle der Boot-Datei zu verwenden. Oder installieren Sie den Sektor boot1 neu.

Fragesteller: Es wird nur das textuelle Gegenstück des BIOS mit fünf Punkten geladen, der oberste ist Weiter>

Antwort: Das bedeutet, dass die Boot-Datei erfolgreich gebootet hat und läuft, aber nicht die CloverX64.efi. Entweder sieht Clover diese Partition nicht, oder es sieht das Gerät überhaupt nicht - wir müssen es herausfinden weiter, indem Sie die Optionen in diesem Menü durchblättern. Es kann z. B. eine fehlende Datei sein. HFSPlus.efi, und Sie haben Clover auf einer HFS+-Partition installiert. Das ist eigentlich ziemlich seltsam, warum macht man UEFI-Boot von einer **HFS+-Partition(?)**.

Fragesteller: Ich habe Clover auf einem Flash-Laufwerk (USB-Stick) installiert, davon gebootet und kann meine HDD nicht sehen.

Antwort: Zunächst muss die Festplatte in den Sata0-Anschluss eingesetzt werden. (Es gab einen solchen Fehler in Intels SATA-Treiber, er ist jetzt behoben.) Zweitens verstehe ich, wenn Sie einen gut funktionierenden Hum, Chimera, HPC, kurz gesagt, BBH (Letter X Booter) haben, Sie wollen es nicht zerstören, sondern Clover ausprobieren, dann scheint eine solche Aktion natürlich. Es gibt jedoch Möglichkeiten, Clover auf der Festplatte zu installieren, die den alten Bootloader nicht zerstören, und in diesem Fall ist der von Ihnen geäußerte Fehler verschwunden.

Versuchen Sie auch die Datei boot7, wenn Sie einen seltsamen SATA/SAS/RAID-Controller haben. Beim UEFI-Boot kann dies auch bedeuten, dass die Dateien PartitionDxe.efi und HFSPlus.efi fehlen.

Fragesteller: Beim UEFI-Laden sehe ich keinen macOS-Abschnitt, nur Legacy.

Antwort: Das bedeutet, dass im Ordner /EFI/CLOVER/drivers/UEFI die Datei HFSPlus.efi oder deren rechtliches Äquivalent zu VboxHFS.efi nicht vorhanden sind.

Fragesteller: Beim Booten mit UEFI sieht Windows wie Legacy aus, obwohl es EFI ist.

Antwort: Das Gleiche, fehlender NTFS.efi-Treiber. Zusatz. Diese beiden Treiber sind aus lizenzrechtlichen Gründen nicht im Repository enthalten, sie müssen diese Dateien irgendwo im Netz zu finden. Es gibt jetzt eine legale Entsprechung. GrubNTFS.efi. Verfügbar im Clover-Installationsprogramm.

Fragesteller: Ich habe die native Auflösung für meinen Monitor im Bootloader eingestellt, aber auf dem der Bildschirm erscheint ein schwarzen Rahmen.

Antwort: Es gibt keine Möglichkeit, es zu reparieren. Den Entwicklern von Clover ist jedenfalls nichts eingefallen, und es gibt niemanden, der diese Frage beantworten kann. Es gibt eine Möglichkeit: Wenn es ein UEFI-BIOS gibt, dann müssen Sie UEFI-Boot auswählen, und das UEFI-Video-BIOS ihrer Grafikkarte flashen. Nehmen Sie im BIOS die Einstellungen vor:

- OS: Windows 8 WHQL
- CSM: Never
- Full screen logo: Disabled

Für ältere Systeme können Sie nichts tun. Mögen Sie den schwermütigen Rahmen nicht, stellen sie eine geringere Auflösung ein.

Fragesteller: Das Betriebssystem bleibt beim Startversuch auf einem schwarzen Bildschirm hängen

Antwort: Dies ist der Zeitpunkt, an dem das DSDT-Patch mit Ihrer Maske erfolgt. Ja, idealerweise sollte es keine Hänger geben. Aber das Problem ist, dass so viele BIOS-Hersteller sich nicht an die Standards halten, sie wissen nicht, wie man programmiert, und sie sind nicht bereit, ihre DSDT auf die Bedürfnisse von OSX anzupassen. Es ist sehr einfach, sicherzustellen, dass der Vorgang „Dekompilieren - Neukompilieren“ nicht passiert - **die DSDT ist schief (?)**. Clover würde gerne dies alles reparieren, aber leider ist die Menge der fehlerhaften Einträge in die DSDTs noch nicht einmal überprüfbar. Daher müssen Sie eine solche Maske auswählen

Fixen sie die DSDT (das DDT) so, dass der Bootloader nicht hängt, und dann so, dass das Betriebssystem nicht hängt, und idealerweise so, dass es auch funktioniert. Dies ist realistisch. Entweder geben Sie Autopatch an (Maske = 0) oder überarbeiten sie die DSDT manuell. Siehe das Kapitel zur **Fehlersuche in dsdt**. Ich empfehle auch sehr die Verwendung von der neuesten Version von Clover, denn solche Fehler werden von Zeit zu Zeit gefunden und können dann korrigiert werden.

Alternativ setzen Sie KernelPM=true.

Fragesteller: Der Kernel beginnt zu booten, gerät aber nach der zehnten Zeile Unable To find driver for this in Panik der ACPI-Plattform.

Dies ist ein fehlender oder falscher DSDT. Wenn der Autopatch fehlschlägt, fügen Sie einen manuell erstellten DSDT hinzu. Achten Sie auf die Autopatch-Optionen sowie auf die Keys ReuseFFFF und DropOEM_DSM.

Fragesteller: Das System beginnt zu booten, friert aber beim Warten auf das Root-Gerät ein....

Antwort: Neben dem üblichen Rat, AHCI im BIOS zu aktivieren, oder, wenn es keinen solchen Treiber gibt, den richtigen Treiber (ich meine kext) für Ihren IDE-Controller zu finden, gibt es auch den Rat, mit der Taste **WithKexts zu booten** (in neueren Versionen von NoCaches), dann wird das Booten langsamer und der Controller hat Zeit zu starten. Dieser Fehler kann übrigens nur auftreten, wenn sich der Clover und das System auf unterschiedlichen Geräten befinden.

Fragesteller: Das System bootet, bis die Meldung: Waiting for DSMOS erscheint...

Antwort: FakeSMC fehlt. Vielleicht hatten Sie mit Chameleon dieses Kext in Extra liegen, und Clover kann diesen Ordner nicht sehen. Der Ordner /EFI/CLOVER/kexts/10.x oder OTHER. Vergessen Sie auch nicht den Key „InjectKexts“. Er ist standardmäßig deaktiviert! In der zweiten Phase der Installation kennt Clover die Version des Systems nicht (sie wurde noch nicht ermittelt), legen Sie also FakeSMC in den Ordner /EFI/CLOVER/kexts/Other/

In neuen Versionen hat der Schlüssel InjectKexts den Wert "Detect", der automatisch diese Situation zu handhaben versucht, überprüfen Sie, was in Ihrer Config.plist geschrieben steht.

Fragesteller: Das System passiert diese Meldung, aber es ändert sich nichts weiter, obwohl die Festplatte (although Winchester buzzes ?) ein Geräusch macht, als ob das System hochfährt (passiert nicht bei SSD auf die HD-LED achten!).

Antwort: Dies ist eine typische Situation, in der sich die Grafikkarte nicht einschaltet. Versuchen Sie GraphicInjector=Yes einzustellen. Bei der zweiten Variante laufen die Radeons auf der "nativen" Konfiguration, oder umgekehrt GraphicInjector =No die es Ihnen sogar erlaubt, bis auf wenige Ausnahmen im System zu arbeiten, Der DVD-Player zum Beispiel funktioniert nicht. Um das Radeon vollständig einzuschalten, müssen sie außerdem die Anschlüsse korrigieren. In anderen Fällen können Sie versuchen, das System mit der Taste x- zu booten., und melden Sie sich am Desktop im VESA-Modus an. Das ist nicht großartig, aber es wird Ihnen die Möglichkeit geben, das sie es reparieren können. Eine andere Variante der Bremse an dieser Stelle wird beobachtet, wenn Sie das MacMini-Modell oder MacBookPro auswählen. Das Problem wird durch Setzen des Schlüssels DropMCFG=Yes oder FixMCFG gelöst

Fragesteller: Das System fährt hoch und zeigt die Meldung an: [Bluetooth-Controller

Antwort: Das ist das Gleiche. Siehe vorherigen Punkt. Bluetooth hat damit nichts zu tun.

Fragesteller: System hat gebootet, alles ist in Ordnung, aber der System Profiler hat Fehler...

Antwort: Im Allgemeinen ist es nur Kosmetik, es beeinträchtigt die Funktionalität nicht. Über PCI-Karten. Siehe das Kapitel über AAPL, Steckplatzname, Über den Speicher. Es gibt zwei Speicher-Frequenz-Werte, den Soll- und den Ist-wert, und sie sind oft nicht übereinstimmend. Welche zeige ich im Profiler an? Ich legte die erste und sie schrien, dass es falsch sei. Legen Sie die zweite ein, diese wurde still, andere Benutzer schrien, dass es falsch sei

Siehe Seite 47 - wie Sie Ihre Speicherwerte in die Config schreiben.

Fazit

Clover ist natürlich noch lange nicht perfekt, aber der Prozess der Programmverbesserung ist nie vollständig. Es wird neue Revisionen geben, es wird neue Funktionen geben, aber im Moment ist es so. Clovers größte Schwäche ist, dass er versucht, universell zu sein. Der Programmierer kann aus den Quellen eine eigene Version erstellen, die für seine Hardware zutreffen. Für den Rest gibt es eine Konfiguration mit Hunderten von Einstellungen, und das ist zu kompliziert für

den Durchschnittsverband, trotz des Vorhandenseins von Automatismen, Anweisungen, Beschreibungen und Massen von Ratschlägen von Kenner. Chameleon arbeitet auf Kosten der BIOS-Treiber und hat daher eine bessere Chance, auf beliebiger Hardware laufen, aber niemand führt Statistiken über den Prozentsatz der Systeme, die mit Clover besser funktionieren.

Die Entwicklung von Clover ist abgeschlossen (**oder doch nicht?**), aber das Projekt ist nicht tot, es läuft weiter und wird sich weiterentwickeln.

Über Chameleon.

Großen Respekt an alle Macher dieses Projekts, die den Mac auf einem regulären P.C. möglich gemacht haben. Clover hat sich viele Technologien davon übernommen, denn es wurde für die gleichen Ziele geschaffen (Videokarteninjektion, efi-Strings, atpi-Patch, ssdt-Generator, smbios-Patch, aber das alles auf einer ganz anderen Ebene).

Ich gehörte auch zu den Entwicklern von Chameleon und bot meine Patches/Verbesserungen an, aber die Projektadministratoren ignorierten mich. Es gibt eine Menge Fehler und einfach Bugs, die nie behoben werden. <http://www.projectosx.com/forum/index.php?showtopic=1106> Wenn Chameleon nicht funktioniert, reden sie nicht darüber, sie ignorieren es einfach.

Der erste Schlag kam im Frühjahr 2011, als das System 10.7 veröffentlicht wurde, und Chameleon es nicht booten konnte. Dann entdeckte Gyk, dass das System von XPC gebootet werden konnte. Das ist ein EFI-Lader (<https://xpcboot.weebly.com/>). Dies war der Startschuss für das Clover-Projekt, einen Open-Source-EFI-Bootloader. Der Grund für das Scheitern von Chameleon lag in der BootArgs-Struktur, die sich im neuen System geändert hatte, sowie in den Legacy-Interrupts. Hut ab vor netkas und cparm, die einen Weg gefunden haben, Chameleon so zu reparieren, dass es das neue System lädt.

Der zweite Schlag kam im Januar 2013, als iMessage erforderte, ROM- und MLB-Variablen im NVRAM zu haben. Clover hat dies bereits im September überwunden, aber mit einem kleinen Fehler in der Länge der Leitung, der erst im Januar korrigiert wurde. Zu dieser Zeit hat Clover iMessage zum Laufen gebracht, und das war für Chameleon unmöglich zu replizieren.

Das Funktionsprinzip ist ganz anders. Meclort und Cosmo1 haben einen Monat gebraucht, um diese Schranke zu überwinden. Seit diesem Winter überstieg die Anzahl der Clover-Benutzer zum ersten Mal die der Chameleon-Benutzer. Aber Chameleon ist wieder voll im Einsatz, und es gibt weiterhin begeisterte Anhänger. "Mit einem Chameleon funktioniert alles!".

Die Chameleonianer ignorierten den dritten Schlag, wie "nein und nicht". Im Januar 2014 des Jahres haben wir Winterschlaf gehalten - Tiefschlaf. Bei Chamäleon funktionierte es nur bis zum System 10.7 aus irgendeinem Grund. Es gab niemanden, der das Warum und Wie untersuchte. Meclort hat sich aus dem Geschäft zurückgezogen, Die übrigen Entwickler im Team können nur neue Grafikkartennamen beisteuern. Clover erwies sich als der einzige Lader, der mit dem der Ruhezustand zumindest im 10.9-System funktionierte.

Ich kann Sie auch daran erinnern, dass mit Chameleon die Floating-Probleme nicht gelöst sind Regionen, den Namen des Slots und eine Menge nicht besonders notwendiger Trivialitäten. Darüber hinaus ist Chameleon voller Fehler, die einfach nicht zu beheben sind.

Die letzte Beule kam im Juni 2014. Apple hat das System 10.10 Yosemite veröffentlicht, das von Clover heruntergeladen werden kann, und die notwendigen Patches wurden bereits vorgenommen, beginnend mit der Revision 2695. Aber für Chameleon sieht es so aus, als ob das Ende gekommen ist... Rückblickend auf die Geschichte, wissen Sie, kann man das nicht mit Sicherheit sagen, alles auf dieser Welt ist möglich, es ist möglich, dass einige der Entwickler alle diese Schranken überwinden werden, und einige Fans werden bei Chameleon bleiben.

Wir wünschen Ihnen einen schönen Aufenthalt!

Zusatz: Ja, auch dieses Problem haben sie gelöst, Chameleon lädt jetzt Yosemite, aber aus irgendeinem Grund gab es Probleme mit 10.9.4, Probleme mit NVRAM und damit mit iMessage. Und nach dem Stand der Forumsaktivität, Chameleon/Chimera haben nur diejenigen, die irgendwie einmal das System benutzt haben, und und nichts daran ändern.

Ein weiterer Schlag, das Aufkommen des apfs-Dateisystems. Für Clover gibt es eine native Apple apfs.efi, aber es funktioniert nur in einer EFI-Umgebung, und in Chameleon nicht. Also wieder, zwei Jahre später Es gibt einen Programmierer, der einen Legacy-APFS-Treiber für Chameleon erstellt hat. Viel Erfolg! Chimera ist ein abgespeckter Brunch von Chameleon, mit eigenem Thema und einem "anderen Video-Injektor". Das heißt, Sie müssen einen DSDT-Patch oder einen Kext vom Typ "natite" anwenden, um die Videokarte zum Laufen zu bringen.

Revoboot ist ein abgespecktes Chameleon, in dem Sie Ihre DSDT kompilieren müssen. Dann es gibt einen Neustart, den sich jeder selbst zusammenstellen muss. Nach Angaben der Ersteller (Master Chief und seine "Tochter" Revogirl) erlaubt es, die Bootzeit für das Lesen von DSDT Dateien zu reduzieren. So ein Quatsch! Die anderen Verbesserungen sind noch fragwürdiger. Zum gegenwärtigen Zeitpunkt. unterstützt von PikeR. Alpha (dem "Sohn" des Häuptlings, dem Bruder dieses Mädchens), der es u.a. schaffte, den Yosemite- zu booten. Für sich selbst kann er es natürlich auch schaffen (**kann das jemand bestätigen?**). Aber für andere Anwender gibt es hier nichts zu bieten.

Ja, Chameleon hat ein Recht auf Leben, weil es ein reines Legacy – Bootloader ist., und er kann dort funktionieren, wo Clover ein Legacy-Boot-Problem hat, alte Computer, Linkshänder-Chipsatz, so etwas in der Art. Ich verstehe den Zusammenhang mit der AMD-CPU nicht wirklich. Bei manchen Leuten funktioniert es, während andere es nicht einmal versuchen, sondern einfach fertige Lösungen mit Chamäleon nehmen. Scheint auch mit Clover zu funktionieren, nur dass niemand nachforscht. Wie auch immer, ich bin es leid, über Chameleon zu reden. Ich behaupte schon seit Jahren, dass Clover besser ist, jemand, der nie überzeugt wurde, kümmert sich nicht darum. Das Thema ist geschlossen.

Andere EFI-Lader.

XRS-Bootloader wurde 2009 angekündigt, ein Team fand sich zusammen und erstellte sogar eine Website Projekt (<https://xpcboot.weebly.com/>). Was mit ihnen passiert ist, weiß ich nicht. Die letzte Meldung lautet: "wegen Spammern". wir werden das Projekt nicht öffnen". Welche Spammer und warum sie sich eingemischt haben, verstehe ich nicht. Das Projekt wurde eingefroren, das Team verstreut. Was bleibt, ist iPhoneTom, der Gründer selbst, der in keiner Weise kooperieren und die Quellen nicht öffnen würde.

Die Sternstunde des Projekts schlug, als Gyk im Frühjahr 2011 10.7 mit XPC installierte, was mit Chameleon unmöglich war, wie ich oben sagte. Tom kam zurück ins Leben, aber er erlaubte nur den Testern, ihre Berichte und Anfragen an das IRS zu übermitteln. Ich habe XPC ausprobiert und festgestellt, dass es nicht auf jedem Computer funktionierte, also begann ich mein Projekt, das war der Start von Clover. Im Herbst 2011 nutzt also die Mehrheit der Anwender Chameleon, die dieses Problem überwand und kräftig zu wachsen begann. Einige haben XPC ausprobiert, und "Anstatt Scheiße zu machen, solltest du Tom lieber mit seinem Lader. Er ist ein ziemlich adäquater Typ, und er hört auf Kritik." Ich hingegen bin Programmierer, ich Ich kann selbständig arbeiten, anstatt im IRC zu sitzen und darauf zu warten, dass der gute Mann etwas repariert. Und ich habe alleine angefangen, einen Bootloader auf Basis des DUET zu erstellen, und im ersten Monat habe ich einige Ergebnisse, die waren besser als XPC. Sei es drum, ich habe mein Know-how nicht an Tom weitergegeben. Ein kleiner anfänglichen Vorteil war die Unterstützung von russische Benutzern, von denen es mehr gibt als alle anderen.

Clover Version 1 benutzte die Schnittstelle von Ninzi, das sie von der frühen Version von XPC "geklaut" hat. In einer solchen Situation war es unmöglich, Clover zu entwickeln, und Anfang 2012 Jahr, als ich alle notwendigen Technologien verstanden hatte, begann ich mit der Entwicklung der Clover-Version 2 basierend auf dem rEFIt-Projekt, mit offenem Quellcode. Ich möchte darauf hinweisen, dass XPC auch vom rEFIt-Projekt kommt, also geht es mehr um Tom, welches Recht hat er, den Quellcode nicht zu veröffentlichen, wenn er selbst Open-Source Quellen verwendet. Jetzt ist Clover lizenziert und hat die Stufe erreicht, auf der war es möglich, über Wettbewerb zu sprechen. Frühjahr 2012. "XPC hat bisher die Funktionalität, die niemandem gelungen ist." Allerdings hatte er ein ungelöstes Systemtypisches Problem, das im Falle eines Laptops den Ruhezustand beeinträchtigte. Und auch die Board-Id, die störte bei einigen Konfigurationen die Installation des Systems 10.7+. Und bei Clover habe ich bei dieser

kein Problem, da ich ursprünglich andere Patches gewählt habe, auf andere Ideen, und das war überhaupt nicht offensichtlich, wenn man sich meine Quellen ansieht. Ich wusste es, aber ich beschloss, es niemandem zu erklären. Die Anwender brauchen es nicht. Er arbeitet in und mit Clover, also werden Sie Clover es benutzen, wenn sie verwenden.

So wurde das bareBoot-Projekt geboren. Autor SunKi, ein eifriger Anhänger von XPC und Toms bester Assistent bei dem Projekt, beschloss, der Sache auf den Grund zu gehen. Er interessierte sich wiederholt dafür, warum und wie in Clover die Dinge gemacht wurden, aber ich machte nie einen Vorschlag, um Clover zu verbessern und um Clovers Fortschritt zu fördern. Nachdem er erkannt hatte, dass ich nicht gehen werde, verriet er seine Geheimnisse und öffnete sein Projekt, z.B. "Ich möchte Dateien zusammenführen, CloverEFI+Patches in einer Datei, und als GUI den vorhandenen SetupBrowser verwenden, mit Änderungen zum Booten mehrerer Systeme, so dass ein Textmenü erhalten wird, in dem Sie das System auswählen können, das nach dem Herunterfahren bootet soll. Ich stimme zu, die Arbeit wurde gemacht, und sie war nicht klein.

Zu diesem Zeitpunkt hatte Dmazar jedoch bereits einen UEFI-Upload durchgeführt, und die Fusion CloverEFI+GUI erwies sich als nicht realisierbar. Bareboot ist ausschließlich entwickelt worden, um Legacy booten zu können. SunKi hatte jedoch nicht das Ziel, einen attraktiven Downloader zu machen, sein Ziel war es die Technologie von Clover zu entschlüsseln. Er begann mit reinem Duet und fügte dann Patches aus Clover Schritt für Schritt, um zu prüfen, was was beeinflusst (und Sie hätten auch mit einem fertigen Clover beginnen können).

Aber auch Clover steht nicht still. Wir haben es, bereits mit Dmazar, schnell verbessert und die Codes umgewandelt, so dass es nicht einfach war, den Überblick zu behalten, noch war es einfach, zu vergleichen, was war und was wurde. Und SunKi konnte nicht herausfinden, wie der Systemtyp in Clover hergestellt wurde.

Zu der Zeit, in der Tom das Projekt aufgab, fand der Barboot keine Trümpfe (there were no trump cards), um Benutzer anzuziehen. Keine Grafik? Nun, wir machten in Clover eine rein textbasiertes Interface, falls jemand allergisch gegen Grafiken ist. Ladegeschwindigkeit? Lass uns konkurrieren.

A

Inzwischen gab es neue Funktionen in Clover, die nicht einfach in bareboot eingebaut werden konnten, insbesondere Patches von DSDTs, Kext und Kernels, ganz zu schweigen von dem UEFI-Loading. Die Anwender zucken nur mit den Schultern: "Wozu brauchen wir überhaupt noch einen Bareboot?"

In der Zwischenzeit gab es ein weiteres bemerkenswertes Ereignis in der Welt des Hackintosh. Eine gewisse QUO-Firma hat ein Mainboard auf Basis des Gigabyte Z77 mit Modifikationen hergestellt für bessere Kompatibilität mit Hackintosh. Aber am wichtigsten ist, dass sie vorgeschlagen haben, dass der Bootloader des Mac direkt in das BIOS geflasht werden sollte. Einer der Begründer dieses Bootloaders, The KiNG, der ständig zum Thema Clover anwesend war, und sich fleißig erkundigte, was und wie, aber auch nichts von Ich habe mich nicht in Clover hineinversetzt. Und hier sehen wir einen bestimmten **Ozmosis**-Boot-Lader, der in das BIOS integriert wird, und er enthält Module, die aus Clover übernommen wurden. Flashen Sie dort im BIOS und einige abgezogene Option von FakeSMC. Und so ist es möglich, reines OSX auf diesem Motherboard zu betreiben, ohne keine einzige Hacker-Datei, keine Bootloader, keine unnötigen Kext. Die Wahrheit ist, meiner Meinung nach, dass all dies ist nur möglich, wenn Sie nichts aktualisieren. Wenn Sie das System aktualisieren, müssen Sie auch das BIOS aktualisieren zum aktuellen System reflashen, und im Allgemeinen können Sie damit u. U. ein Brikett herstellen. Über die Aufrüstung der Fälschung und der Sensoren zu ist eine große Frage. Und natürlich ist dieser Bootloader nicht für andere Motherboards ausgelegt

King hat kürzlich auch einen anderen Satz fallen lassen: "Oz ist für Laptops inakzeptabel". Und ich vermute, dass es nicht nur um die Tatsache geht, dass ein Ziegelstein durch erneutes Brennen des BIOS entsteht. Realistischerweise ist Oz speziell auf das Gigabyte Z77-Board abgestimmt, und der Betrieb auf anderer Hardware ist fraglich. Viel Spaß beim Segeln!

Mit der Veröffentlichung des EICapitan-Systems, noch Beta, erlebte auch der Bootloader Ozmosis einen Schock. In diesem System sind Kexte von außen, z. B. aus dem BIOS, nicht ladbar. Wir haben dieses Problem für Clover gelöst (danke Solstice), aber es liegt im Clover-Körper selbst. Es gibt niemanden, der den Ozmosis-Körper repariert, die Quellen sind geschlossen. Bleibt bei den alten Systemen, Kameraden!

King hat vor kurzem auch einen anderen Satz fallen lassen: "Oz ist für Laptops inakzeptabel". Und ich vermute, es besteht nicht nur die Gefahr, dass ein Ziegelstein durch erneutes Brennen des BIOS entsteht. Realistisch gesehen hat sich Oz nur für das Gigabyte Z77-Board eingependelt, und der Betrieb auf anderer Hardware ist in Frage gestellt. Viel Spaß beim Segeln!