

Autor Slice



Original official document -

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

[Bootloader/releases/download/5128/Clover_Khaki_Color_rus_5128.pdf](https://github.com/CloverHackyColor/Clover-Bootloader/releases/download/5128/Clover_Khaki_Color_rus_5128.pdf)

Übersetzt mit DeepL
Version 0.1
21.012021

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 NSTALLATIONSPROGRAMMS	31
DEN BOOTLOADER MANUELL..... INSTALLIEREN	34
OSX.....	34
Linux.....	35
Windows.....	36
EMPFOHLENE INSTALLATIONSOPTIONEN.....	36
DURCHSETZUNG	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>MaxMultipliiier</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
B. 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
B. 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

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 installierte 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 beschere 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. Im Einzelnen. Oxalis, und es ist in Blumenläden sowohl dreiblättrig als auch vierblättrig erhältlich :)

In der russischen Version nennen wir den Lader "Clover". Das heißt, nicht Clover.
Die Projektentwicklung findet in den Foren statt



1 Ich frage mich, wie man Klee zufällig finden kann? Durch regelmäßiges Zwicken 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 alles, was er bis dahin zu verstehen vermochte, wich der Entwicklung aus und verschwand bald ganz.

Y 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 neues 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 Chamäleon 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 Das 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, Mieke 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 geschrieben ist Codes, 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 ist. 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 CPU, es sind keine CPU-Daten in den internen Tabellen vorhanden, oder eine Art Allgemeingültigkeit, die in einem bestimmten Fall 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 gestart 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 wie in Option B.

Option B. Computer basierend auf UEFI-BIOS (neues Schema, UEFI-Laden)
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 vorgeschrieben 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 PKR-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.

PKR-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. Es hat 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 folgendermaßen aus:

EFI:

```
BOOT:
  BOOTX64.efi
CLOVER:
  ACPI:
    WINDOWS:
      SLIC.aml
    origin :
      gepatcht:
        DSDT.aml.
        SSDT-1.aml
  CLOVERX64.efi
Themen:
  schwarz_grün:
    BoG_LucidaKonsole_10W_NA.png
  Icons:
    func_about.png
    os_clover.icns
    banner.png
    background.png
    Selection_big.png
    theme.plist
  Clovy:
  Cäsium:
    thema.svg
config.plist
drivers:
  BIOS:
    FSInject.efi
    SMCHelper.efi
  UEFI:
    CsmVideoDxe.efi
```

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

```

    DataHubDxe.efi
    FSInject.efi
    VBoxHFS.efi
    OsxAptioFix3Drv.efi
    SMCHelper.efi
kexts:
    10.7:
    10.11:
    Off:
    Other:
    misc:
    OEM:

        Inspiron 1525:
            ACPI:
            Herkunft:
            gepatcht:
                DSDT.aml.
            config.plist
            kexts:
            10.5:
            10.6:
                Injector.kext
                VoodooSDHC.kext
            10.7:
                VoodooTSC.kext
            Other:

UEFI:
    ACPI:
        Other:
        patched:
        DSDT.aml.
    config.plist
    kexts:

ROM:
tools:
    Shell64U.efi

```

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 Ordnernamen 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, aber auf Desktops gibt 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 UEFI-Ordner in Ihrem Ordner haben, um verschiedene Konfigurationen für UEFI 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 UEFI-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 UEFI. 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 UEFI 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 UEFI 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 UEFI-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!