

Hackintosh Hilfe in Berlin

Beitrag von „whatthefaq“ vom 27. März 2021, 13:24

Hallo Forum,

Vor zwei Jahren hat ein sehr technisch versierter Freund einen Hackintosh für mich konfiguriert und High Sierra darauf installiert. Bei diversen Versuchen auf Big Sur zu updaten bin ich gescheitert, meine PC Kenntnisse reichen nicht aus um auf die ganzen Fehlermeldungen flexibel eingehen zu können. Ich wohne im Westen Berlins und würde gerne Wissen ob jemand der ebenfalls in Berlin wohnt mir gegen eine im Vorfeld besprochene Entlohnung mit meinem Problem helfen würde?

Vielen Dank und schönes Wochenende an alle.

Beitrag von „jeve stobs“ vom 27. März 2021, 13:44

hallo willkommen im forum,

vllt trägst du deine komponenten ein damit

dir jemand helfen kann der sich dann dein

setup mal anschaut 😊

Beitrag von „HackBook Pro“ vom 27. März 2021, 13:47



Bitte lade mal deinen EFI Ordner hoch 😊.

Beitrag von „whatthefaq“ vom 27. März 2021, 13:54

Zitat von jeve stobs

hallo willkommen im forum,
vllt trägst du deine komponenten ein damit
dir jemand helfen kann der sich dann dein
setup mal anschaut 😊

Asus Prime Z370-A

Core i7-8700K

32GB 3200-Predator

Sapp8GB RX580 Nitro

und zum booten hat mein Freund Clover installiert, darüber starte ich den PC dann immer als Mac mit OS High Sierra als Betriebssystem. Jetzt würde Ich gerne auf Big Sur updaten oder alternativ einen Dualboot einrichten mit Windows 10, aber habe beides wie erwähnt alleine nicht hinbekommen auch nach intensiver Recherche.

Vielen Dank schonmal fürs antworten 😊

Beitrag von „theCurseOfHackintosh“ vom 27. März 2021, 14:03

Du solltest am besten auf opencore umsteigen. Clover nutzt auch opencore für big sur. Hier gäbe es eine 0.5.5 opencore efi [Asus Prime Z370-A](#)

Erstelle aber am besten eine neue nach Guide <https://dortania.github.io/Ope...ig.plist/coffee-lake.html>

Beitrag von „whatthefaq“ vom 27. März 2021, 14:04

[Zitat von HackBook Pro](#)



Bitte lade mal deinen EFI Ordner hoch

DANKE 😊 Was ist der EFI Ordner bzw. wo finde ich den ? Bin leider wirklich kein PC Ass, bin nach Berlin gezogen und jetzt kann meiner früher Ansprechpartner in diesen Sachen mir nichtmehr helfen...

Beitrag von „HackBook Pro“ vom 27. März 2021, 14:12

Lese dich ein (der Guide, den [theCurseOfHackintosh](#) verlinkt hat), dann wirst du es schon herausfinden 😊.

Beitrag von „griven“ vom 27. März 2021, 14:18

Hallo [whatthefaq](#),

lade Dir mal das Tool Kextupdater ([Kext Updater - Neue Version 3.x](#)) von unserem User [Sascha 77](#)

herunter und starte es. Das Tool gibt Dir eine einfache Möglichkeit um zum einen alle Deine Kexts (vergleichbar mit den Treibern unter Windows) aktuell zu halten und zum anderen auch die EFI Partition einzublenden auf der sich Dein Clover befindet.



Klick einfach auf die Schaltfläche EFI und gib ggf. Dein Kennwort ein und die Partition wird eingebunden und erscheint als Laufwerk auf dem Desktop und/oder im Finder. Auf der nun eingebundenen Partition findest Du einen Ordner namens EFI diesen Ordner kopierst Du Dir nun auf den Schreibtisch und löschst im Anschluss aus dem auf den Schreibtisch kopierten Ordner den Ordner "Apple" (der Apple Ordner ist nur für echte Mac's relevant für Hacks nicht). Wenn der Apple Ordner gelöscht ist komprimierst Du den auf den Schreibtisch kopierten Ordner und lädst ihn hier im Forum hoch. Ausgestattet mit dem Ordner und abhängig davon wie weit Dein Interesse geht Dich selbst in die Materie einzuarbeiten kann man Dir dann entweder einen Basis OpenCore Ordner an die Hand geben der in der Lage ist Dein System zu starten und ein Upgrade auf BigSur ermöglicht Dir aber das "Feintuning" überlässt oder aber zusammen mit Dir eine Lösung erarbeiten die dann keine Wünsche mehr offen lässt aber nicht von der verlangt und erwartet das Du Dir das Thema von Adam und Eva angefangen komplett rauf schaffst.

Beitrag von „whatthefaq“ vom 27. März 2021, 14:36

- Vollzitat entfernt -

Vielen Danke für die ausführliche Anleitung. Ich bin den Schritten gefolgt und hoffe dass ich

das richtige hochgeladen habe.

Wenn sich eine Möglichkeit bietet bei der mir jemand live helfen könnte wäre mir das am liebsten, aber wenn das nicht möglich ist versuche ich mich natürlich mit Hilfe eurer Tipps durchzuschlagen auch wenn die Chancen dass ich dann erfolgreich bin ungefähr so groß sind wie wenn Frodo ohne Sam nach Mordor gegangen wäre... 🤪

Beitrag von „theCurseOfHackintosh“ vom 27. März 2021, 14:42

Du hast nur einen Kext im other Ordner... Wo hast du macos heruntergeladen? Deine clover efi müsste für big sur sowieso überarbeitet werden, daher besser auf opencore umsteigen. Habe ich weiter oben verlinkt.

Bitte keine Vollzitate!

Beitrag von „griven“ vom 27. März 2021, 14:46

Das mit der EFI hat doch schonmal ganz prima geklappt 🤪

Im nächsten Schritt wäre es nun ganz gut wenn Du den KextUpdater erneut bemühen würdest diesmal aber um einen Systemreport zu erzeugen. Die Schaltfläche dafür findest Du ebenfalls direkt im Hauptfenster des Tools. Der Systemreport sammelt ein paar Informationen zu Deinem System unter anderem darüber welche Treiber installiert sind und wo diese installiert sind. Mir erscheint es wichtig gerade diesen Punkt abzuarbeiten denn nach einem Blick in Deinen EFI Ordner sieht es für mich danach aus als hätte der Bekannte der Dir den Rechner eingerichtet hat einen Großteil der notwendigen Kexts direkt ins System installiert was eher schlecht ist und was wir ändern müssen bevor wir darüber nachdenken können ein Update zu wagen. Nach dem Klick auf die Schaltfläche fängt das Tool damit an die notwendigen Informationen zusammen zu tragen und legt diese dann am Ende in eine Datei namens "Systemreport.zip" auf dem Schreibtisch ab. Diese Datei bitte auch hochladen.

Beitrag von „HackBook Pro“ vom 27. März 2021, 14:48

[theCurseOfHackintosh](#) Vermutlich an nem mac, aber mir ist da was anderes aufgefallen...

Beitrag von „whatthefaq“ vom 27. März 2021, 14:51

Ich habe den Computer wie erwähnt nicht selbst konfiguriert deswegen kann ich sehr schlecht auf die Fragen eingehen. Sorry wusste nicht wie man das mit den Zitaten machen soll werde es mir merken 😊

Hier ist der Systemreport den griven angefragt hat 😊

Beitrag von „griven“ vom 27. März 2021, 14:59

Ein kurzer Blick in den Report bestätigt meine Vermutungen 😊

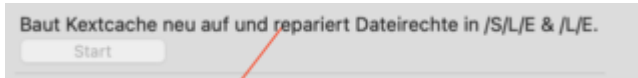
Dein Bekannter hat die Extensions direkt ins System gepackt anstatt sie in den EFI Ordner zu legen legen wo man sie eigentlich hinlegt:



All loaded Non-Apple Kexts	
FakeSMC_ACPISensors	/Library/Extensions
AppleALC	/Library/Extensions
FakeSMC_CPUSensors	/Library/Extensions
FakeSMC	EFI/OC/Extensions/Other
FakeSMC_GPUSensors	/Library/Extensions
IntelBiosEthernet	/Library/Extensions
FakeSMC_IOPSensors	/Library/Extensions
Lilu	/Library/Extensions
LogInjectKext	/
USBInjectAll	/Library/Extensions
WhateverGreen	/Library/Extensions

man kann beides machen aber die elegantere und bessere Variante ist es diese Dateien in den Ordner /EFI/Clover/Kexts/Other zu legen und es Clover bzw. dann in Zukunft OpenCore zu überlassen diese Dinge ins System einzubringen. Ich muss jetzt erstmal für ein paar Stunden weg vom Rechner würde Dich aber bitten in der Zwischenzeit die im Screenshot gezeigten Dateien auf die EFI Partition in den Ordner /EFI/Clover/Kexts/Other zu kopieren und aus dem

Ordner /Library/Extensions (auf der Festplatte) zu entfernen. Wenn du das erledigt hast kannst Du im KextUpdater auf die Schaltfläche "Werkzeuge" klicken und in dem sich öffnenden Fenster den Punkt



ausführen und durchlaufen lassen. Auf die Weise schaffen wir dann nämlich schon mal eine saubere Basis für alle weiteren Aktionen.

Beitrag von „whatthefaq“ vom 27. März 2021, 15:28

[Zitat von griven](#)

Auf die Weise schaffen wir dann nämlich schon mal eine saubere Basis für alle weiteren Aktionen.

Habe die Dateien aus dem Screenshot in den Ordner kopiert und dann aus der library gelöscht. Danach habe ich wie angewiesen das Kextupdater Werkzeug durchlaufen lassen.

Nur eine Abweichung hatte ich, Die Datei in meiner Library hieß G13joystick anstatt logjoystick...weiß nicht ob das relevant ist

Beitrag von „GerhardM“ vom 27. März 2021, 16:58

[whatthefaq](#) Du merkst hier im Forum wird Dir geholfen. Trotzdem rate ich Dir aus eigener Erfahrung, Dir ein paar Grundlagen anzueignen. Es schafft nicht nur Erfolgserlebnisse, wenn man Änderungen erfolgreich durchgeführt hat (gilt für Clover und OpenCore gleichermaßen), sondern macht Dich auch "robuster" bez. auftretender Störungen.

Beitrag von „shane52“ vom 27. März 2021, 17:29

Schau mal hier da hat es auch jemand geschafft für Catalina dann klappt es auch für Big Sur

https://www.reddit.com/r/hacki..._1556_asus_prime_z370a_i/

Und schaue Dir mal die Videos von Chris an die sind auch sehr Informativ!

<https://www.youtube.com/watch?v=ZYyuoC1WWfg>

Beitrag von „whatthefaq“ vom 27. März 2021, 18:16

[Zitat von GerhardM](#)

Du merkst hier im Forum wird Dir geholfen.

Ja ich bin wirklich happy wie schnell und fundiert mir geantwortet wird 😊 Ich versuche es hinzukriegen, bisher haben mir die Anweisungen von griven sehr geholfen. bin grade dabei mich durch die anderen Sachen alle durchzulesen aber werd nicht aus allem schlau.

Beitrag von „umax1980“ vom 27. März 2021, 18:27

Wenn man sich ein wenig reinfuchst dann läuft das ab einem gewissen Punkt ganz gut alleine. Wichtig ist immer eine Sicherung des funktionierenden EFI Ordners, am besten auf einem dafür gemachten USB Stick. So kannst du das System im Problemfall einfach vom Stick starten und es kommt zu keinem Komplettausfall.

Beitrag von „g-force“ vom 27. März 2021, 18:33

<https://www.hackintosh-forum.de/forum/thread/53479-hackintosh-hilfe-in-berlin/>

Ein guter Start für das Grundverständnis "Hackintosh" ist oft mein Guide (siehe Signatur).

Beitrag von „Schwarzkopf“ vom 27. März 2021, 19:05

Willkommen im Hackintosh Forum

Beitrag von „griven“ vom 27. März 2021, 20:40

[whatthefaq](#) wenn Du nun alles in die EFI verlagert hast und das Tool durchgelaufen ist und nach einen Neustart noch alles wie vorher funktioniert haben wir jetzt ein Basis auf der wir aufbauen können. Wenn Du möchtest baue ich Dir aus der Clover Vorlage eine OpenCore EFI sprich ich würde hier das was in Deiner Clover Konfiguration jetzt enthaltene 1:1 auf OpenCore übertragen. Mit dieser Blaupause kannst Du dann im ersten Step versuchen den Rechner zu starten hierzu kommt die OpenCore EFI zunächst auf einen USB Stick und der Rechner wird über diesen USB Stick gestartet. Das Vorgehen mit dem USB Stick hat den Vorteil das die bestehende und funktionierende Clover Installation vorerst unangetastet bleibt und als Notanker zur Verfügung steht falls was schief geht. Wenn es gewünscht ist kann ich auch gerne dokumentieren und darlegen welche Teile der Clover Konfiguration in welche Teile der OpenCore Konfiguration aufgehen und welche Einstellungen aus welchen Gründen vorgenommen wurden.

Beitrag von „whatthefaq“ vom 27. März 2021, 21:06

[Zitat von griven](#)

Wenn Du möchtest baue ich Dir aus der Clover Vorlage eine OpenCore EFI

Das wäre wunderbar. Ich finde die Art wie du mich Schritt für Schritt durch den Prozess navigierst sehr verständlich und weniger überfordernd als wenn ich den ganzen Kontext meiner Handlungen komplett durchblicken müsste. Ich habe bereits einen USB den ich dafür nutzen wollte. Wenn du mir weiter vorgibst was ich zu tun habe kriege ich es vielleicht wirklich

ohne einen live Besuch von einem Experten hin. Danke nochmal für die Bemühungen 😊

Beitrag von „griven“ vom 27. März 2021, 21:08

Sehr gerne 😊

Ich mache mich später ran spätestens aber morgen denn aktuell sind wir im Teammeeting und da kann ich parallel nicht sinnvoll basteln aber wir bekommen das hin 😊

So, Meeting ist durch und eine mehr oder weniger erholsame Nacht später kann es nun also losgehen und wir können das Projekt "wie komme ich von einem alten Clover zu einem aktuellen OpenCore" angehen. Bevor wir loslegen können müssen wir natürlich erstmal alle Zutaten besorgen hier gehe ich persönlich immer so vor das ich zunächst eine Bestandsaufnahme mache und davon ausgehend entscheide was benötigt wird, was aktualisiert werden muss und was 1:1 übernommen werden kann. Fangen wir also mit dem Bestand an hier wäre also mal der IST Zustand aus den vorliegenden Informationen:

Bootloader: irgendeine ältere Clover Version

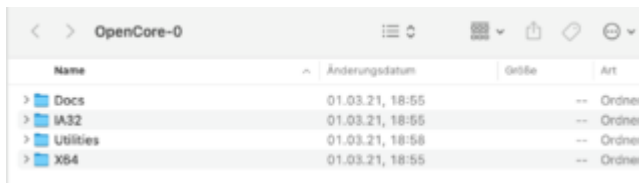
Extensions:

- FakeSMC nebst Plugins
- Lilu
- AppleALC
- IntelMausiEthernet
- UsbInjectAll
- WhatEverGreen

Ausgehend vom Ist Zustand entscheide ich nun was ich hiervon unverändert übernehmen kann was weg kann und was ggf. aktualisiert werden sollte und komme zu dem Ergebnis das es in diesem fall nicht schadet alles in der jeweils aktuellen Version neu zu besorgen bzw. eben einige Dinge auch zu ersetzen da es hier inzwischen bessere Alternativen gibt. Zum herunterladen der notwendigen Extensions bediene ich mich des KextUpdaters und wähle hier wie folgt aus:



Das Ergebnis der Aktion ist ein Ordner namens Kext-Updater der nun auf dem Desktop zum liegen gekommen ist und der alle Downloads aus dem Tool beinhaltet. Wie vielleicht auffällt habe ich die bisher verwendete FakeSMC aussortiert und an ihrer statt VirtualSMC ausgewählt der Rest ist 1:1 geblieben und liegt nun in der jeweils aktuell Form vor. Der Grund dafür das FakeSMC aussortiert wurde liegt darin das dieser Kext schon seit geraumer Zeit nicht mehr gepflegt wird und in virtualSMC eine aktuellere und weiterhin aktiv entwickelte Alternative gefunden hat. Bezogen auf die Extensions haben wir hier also schon mal alles zusammen was wir brauchen und somit geht es im nächsten Schritt an den Bootloader selbst. Ich habe mir hierzu angewöhnt die jeweils aktuell Release Variante von der Acidanthera Github Seite zu verwenden (<https://github.com/acidanthera/OpenCorePkg/releases>) zumindest dann wenn ich nicht an meiner eigenen EFI schraube alternativ kann man aber auch hierzu den KextUpdater bemühen der letztlich auch nichts anderes macht als die Release Version von Github zu laden. Egal woher man den Loader nun nimmt am Ende des Downloads steht eine Verzeichnisstruktur die wie folgt aussieht:



Die für uns nun relevanten Ordner sind die Ordner Docs und X64. Der Docs Ordner enthält neben der gesamten Dokumentation zu OpenCore auch alle nötigen Vorlagen im X64 Ordner finden wir eine Vorlage für unser künftiges EFI Verzeichnis sprich hier haben wir dann auch direkt die Struktur des Verzeichnisses vorgegeben und wir können schon mal damit anfangen die Struktur zu füllen.

X64	01.03.21, 18:55	--	Ordner
EFI	01.03.21, 18:55	--	Ordner
BOOT	01.03.21, 18:55	--	Ordner
OC	01.03.21, 18:55	--	Ordner
ACPI	01.03.21, 18:55	--	Ordner
Drivers	01.03.21, 18:55	--	Ordner
Kexts	01.03.21, 18:55	--	Ordner
OpenCore.efi	01.03.21, 18:55	500 KB	Dokument
Resources	01.03.21, 18:55	--	Ordner
Tools	01.03.21, 18:55	--	Ordner

Alle Extensions die wir heruntergeladen haben bzw. die wir direkt aus dem alten EFI Ordner übernehmen können wir also schon mal in den Ordner **Kexts** ziehen und haben damit schon einen guten Teil auf dem Weg hin zu OpenCore erledigt. Im nächsten Schritt geht es nun darum den Ordner ACPI mit Inhalt zu füllen hierzu muss man wissen das OpenCore anders als Clover von sich aus nichts am ACPI des Rechners verändert hier sind wir also selbst dafür verantwortlich alle notwendigen Fixes selbst einzubringen aber keine Sorge bei Desktop Systemen hält sich der notwendige Aufwand dabei in sehr engen Grenzen und darüber hinaus liefern die AcidAnthera Jungs auch alles Nötige direkt mit. Um zu ermitteln welche ACPI Fixes wir wirklich brauchen empfiehlt sich an der Stelle ein Blick in den schon mehrfach erwähnten Dortania Guide denn hier wird in Abhängigkeit zur eingesetzten Systemgeneration sehr übersichtlich aufgezeigt was nötig ist. In Deinem Fall brauchen wir für ein Z390 System folgende "Zutaten":

- SSDT-Plug.aml
- SSDT-EC-USBX.aml

die wir im Ordner Docs -> AcpiSamples -> Binaries finden und von dort in den Ordner X64/EFI/OC/ACPI kopieren (für den Moment soll uns das reichen und wir akzeptieren einfach das wir die Dateien benötigen ohne zu hinterfragen wozu sie notwendig sind). An der Verzeichnisstruktur haben wir nun fast alles Notwendige getan fehlt eigentlich nur noch ein Schritt nämlich das hinzufügen der Ressourcen sofern wir zum Beispiel die grafische Bootauswahl verwenden möchten. Die Ressourcen finden wir unter <https://github.com/acidanthera/OcBinaryData> wobei wir uns das Github Repo der Einfachheit halber einfach als ZIP herunterladen. Aus dem heruntergeladenen Ordner kopieren wir aus dem Ordner Resources die Ordner "Font", "Image" und "Label" und fügen sie in unsere EFI Ordnerstruktur in den Ordner Resources ein die Frage ob wir die bestehenden Ordner ersetzen möchten beantworten wir mit "ja für alle". Im Endergebnis sieht der EFI Ordner also bisher so aus:

Name	Änderungsdatum	Größe	Art
BOOT	01.03.21, 18:55	--	Ordner
OC	Heute, 10:32	--	Ordner
ACPI	Heute, 10:55	--	Ordner
SSDT-EC-USBX.aml	01.03.21, 18:55	306 Byte	ACPI M...Binary
SSDT-PLUG.aml	01.03.21, 18:55	900 Byte	ACPI M...Binary
Drivers	01.03.21, 18:55	--	Ordner
Kexts	Heute, 10:31	--	Ordner
AppleALC.kext	Heute, 04:03	1,9 MB	Kernel-...terung
IntelMausI.kext	Heute, 04:04	172 KB	Kernel-...terung
SMCProcessor.kext	Heute, 04:05	64 KB	Kernel-...terung
SMCSuperIO.kext	Heute, 04:05	95 KB	Kernel-...terung
USBInjectAll.kext	Heute, 04:09	385 KB	Kernel-...terung
VirtusSMC.kext	Heute, 04:05	113 KB	Kernel-...terung
WhateverGreen.kext	Heute, 04:03	506 KB	Kernel-...terung
OpenCore.efi	01.03.21, 18:55	500 KB	Dokument
Resources	01.03.21, 18:55	--	Ordner
Tools	01.03.21, 18:55	--	Ordner

Damit sind die Vorbereitungen auch schon abgeschlossen und wir kommen zum spannenden Teil nämlich dem anpassen der config wobei uns auch hier ein Vorlage praktischerweise direkt mitgeliefert wird. Zum bearbeiten der config benötigen wir einen .plist Editor (auch wenn es grafische Editoren gibt empfiehlt es sich für OpenCore eher einen .plist Editor zu verwenden) hier gibt es unterschiedliche, sowohl kostenfreie als auch kostenpflichtige, zur Auswahl. Ich habe mich über die Jahre an PlistEditPro gewöhnt (kostenpflichtig) am langen Ende ist es aber reine Geschmacksache. Öffnet man die beigelegte sample.plist erkennt man folgende Struktur die nun von oben nach unten an unsere Gegebenheiten angepasst wird:



Ich werde versuchen zu jedem Punkt den ich in der Config anfasse eine kleine Erklärung zu liefern beschränke mich hierbei aber wirklich auf die nötigsten Punkte denn andernfalls könnte ich auch gleich die Dokumentation von OpenCore posten und das will ja niemand 😊

Vom Grundsatz her finden wir unter vielen Punkten in der Config eine ähnliche Struktur. immer wieder begegnen uns "Add", "Delete", "Patch" und Quirks wobei die Logik immer ist "füge hinzu was noch nicht vorhanden ist" (Add), lösche was definiert wurde (Delete), Verändere was

definiert wurde (Patch) und bügeln was kaputt ist (Quirks). Von der Reihenfolge wird immer zuerst Delete dann Add dann Patch und dann Quirks ausgeführt (sollte man für alles was man tut im Hinterkopf haben). Eine Eigenheit von OpenCore ist es dass anders als bei Clover nichts vom Dateisystem verwendet wird solange es nicht explizit in die config eingetragen wurde was für uns bedeutet dass wir im nächsten Schritt also die Dateien die wir zuvor im Dateisystem abgelegt haben quasi in der Config einchecken. Fangen wir oben mit dem Punkt ACPI an wenn wir hier den Knoten Add aufklappen gibt es im Sample schon eine ganze Litanei an eingetragenen Objekten und im Grunde müssen wir uns hier nur die beiden Einträge aussuchen die mit den Dateien korrespondieren die wir in den Ordner /EFI/OC/ACPI gelegt haben und diese aktivieren.



die restlichen Einträge benötigen wir nicht diese können wir entweder löschen oder einfach so lassen wie sie sind (ich lösche sie der Übersicht halber immer) und damit sind wir mit dem Knoten ACPI schon fertig und können weiter zu Booter wandern. Unter "Booter" ist der Punkt Quirks der spannende Punkt denn alles was wir hier einstellen können/müssen beeinflusst direkt wie sich der Booter von MacOS (boot.efi) auf unserem System verhält. Ich habe hier für Dein Setup folgende Einstellungen gewählt:

Property-List	Typ	Wert
Booter	Dictionary	3 Schlüssel-Wert-Paare
MmioWhitelist	Array	2 geordnete Elemente
Patch	Array	1 geordnete Elemente
Quirks	Dictionary	10 Schlüssel-Wert-Paare
AllowRelocationBlock	Boolean	NO
AvoidRuntimeDefrag	Boolean	YES
DevirtualiseMmio	Boolean	NO
DisableSingleUser	Boolean	NO
DisableVariableWrite	Boolean	NO
DiscardMemoryMap	Boolean	NO
EnableSafeModeSlide	Boolean	YES
EnableWriteUnprotector	Boolean	YES
ForceExitBootServices	Boolean	NO
ProtectMemoryRegions	Boolean	NO
ProtectSecureBoot	Boolean	NO
ProtectUefiServices	Boolean	NO
ProvideCustomSlide	Boolean	YES
ProvideMaxSlide	Integer	0
RebuildAppleMemoryMap	Boolean	NO
SetupVirtualMap	Boolean	YES
SignalAppleOS	Boolean	NO
SyncRuntimePermissions	Boolean	NO

und habe mich dabei mehr oder weniger an die Voreinstellung gehalten was man im übrigen bei aktueller Hardware in den meisten Fällen auch getrost machen kann (bei älterer oder exotischere Hardware wie zum Beispiel bei X-Serie Boards ist hier ggf. mehr Arbeit nötig). Weiter im Text geht es mit dem Knoten Kernel der Knoten hat in der config folgende Struktur:

Kernel	Dictionary	7 Schlüssel/Wert Pairs
Add	Array	7 geordnete Elemente
Block	Array	1 geordnete Elemente
Emulate	Dictionary	5 Schlüssel/Wert Pairs
Force	Array	1 geordnete Elemente
Patch	Array	7 geordnete Elemente
Quirks	Dictionary	10 Schlüssel/Wert Pairs
Scheme	Dictionary	3 Schlüssel/Wert Pairs

Übersetzt auf Clover entspricht:

- Add -> Den Kext Verzeichnissen unter /Clover/Kexts/ wobei unter Clover die Extensions nicht explizit in der Config eingetragen werden
- Block -> Hier hat Clover keine Entsprechung
- Emulate -> Entspricht dem Bereich CPU in Clover (CloverConfigurator)
- Force -> Entspricht "Force Kext2Load"
- Patch -> Entspricht "Kext2Patch" und "Kernel2Patch"
- Quirks -> Entspricht Quirks in neueren Clover Versionen und ist bei alten Versionen nicht vorhanden
- Scheme -> ist in Clover nicht vorhanden

In OpenCore müssen wir also unsere Extensions in die config eintragen damit OpenCore diese verwendet hierbei gilt es zu beachten das die richtige Reihenfolge eine Rolle spielt denn alle Extensions werden in exakt der Reihenfolge geladen in der sie in die config eingetragen wurden wenn also Extensions Abhängigkeiten zueinander haben müssen wir das an der Stelle berücksichtigen. Ausgehend von dem was wir zuvor in das Verzeichnis Kexts kopiert haben tragen wir also ein:

Kernel	Dictionary	7 Schlüssel/Wert Pairs
Add	Array	7 geordnete Elemente
0	Dictionary	8 Schlüssel/Wert Pairs
Arch	String	x86_64
BundlePath	String	Libkext
Comment	String	Patch engine
Enabled	Boolean	YES
ExecutablePath	String	Contents/MacOS/Libk
MaxKernel	String	
MinKernel	String	10.0.0
PathPath	String	Contents/info.plist
1	Dictionary	8 Schlüssel/Wert Pairs
2	Dictionary	8 Schlüssel/Wert Pairs
3	Dictionary	8 Schlüssel/Wert Pairs
4	Dictionary	8 Schlüssel/Wert Pairs
5	Dictionary	8 Schlüssel/Wert Pairs
6	Dictionary	8 Schlüssel/Wert Pairs

An der Stelle ein paar Worte zu den einzelnen Punkten da ich denke das es dem Verständnis zuträglich ist...

- Arch -> Entspricht der zu verwendenden Architektur mögliche Werte sind hier X86_64 (64Bit), I386 (32Bit) oder Any
- BundlePath -> Der volle Dateiname der einzutragenden Extension
- Comment -> Ein frei wählbarer Kommentar der Beschreibt um was es sich handelt
- ExecutablePath -> Der Pfad zum ausführbaren Teil der Extension (eigentlich immer /Contents/MacOS/name des Kext ohne die Endung .kext
- MaxKernel und MinKernel -> Die MacOS Version bis zu der die Extension maximal verwendet werden soll oder die mindestens vorliegen muss damit sie verwendet wird (entspricht den Versionsordnern in Clovers Kexts Verzeichnis)
- PlistPath -> Pfad zur Info.plist eigentlich immer /Contents/Info.plist

In Deinem Fall ist die Liste der Extensions glücklicherweise übersichtlich und wir daher schnell mit dem eintragen fertig 😊 Wichtig ist das Lilu.kext an erster Stelle in der Liste steht weil alles was danach kommt von diesem Kext abhängig ist (WhateverGreen, AppleALC, VirtualSMC...) haben wir hier alles eingetragen geht es weiter mit den Punkten MISC, NVRAM und PlattformInfo. Ich spare jetzt die Punkte MISC und NVRAM für den Moment aus und reiche sie in einem weiteren Beitrag nach und gehe nur noch schnell auf den Punkt PlattformInfo ein welcher dem Punkt SMBIOS in Clover entspricht.

PlatformInfo	Dictionary	8 Schlüssel/Wert-Paare
Automatic	Boolean	YES
CustomMemory	Boolean	NO
Generic	Dictionary	10 Schlüssel/Wert-Paare
AdvertiseWindows	Boolean	NO
MLB	String	MD0000000000000001
MaxBIOSVersion	Boolean	NO
ProcessorType	Zahl	0
ROM	Daten	6 Bytes: 11223344 5566
SpoolVendor	Boolean	YES
SystemMemoryStatus	String	Auto
SystemProductName	String	iMac18,1
SystemSerialNumber	String	W00000000001
SystemUUID	String	00000000-0000-0000-0000-000000000000

Unter OpenCore ist es relativ einfach die Systemdefinition einzutragen und zu erzeugen denn alles was wir dazu brauchen sind im Kern 4 Werte aus unserer alten Clover Config den Rest können wir getrost OpenCore überlassen denn es wird uns ein schöner Automatismus mitgeliefert der aus unseren eingegeben vier Werten ein komplettes und valides SMBIOS erzeugt. In Deinem Fall ist der Rechner aktuell als iMac18,3 deklariert und das übernehmen wir auch erstmal so (später wäre es sinnvoll auf iMacPro 1,1 zu gehen bei Deinem Setup) alles was wir brauchen ist ausser dem Modell noch die Seriennummer, die SystemSerialNumber (MLB) die SystemUUID (SMUUID) und den ROM Wert um den Rest kümmert sich OpenCore. Also fix eingetragen und Haken an den Punkt PlattformInfo. Soweit erstmal für den Moment die

resultierende EFI findest Du hier: [EFI.zip](#) die kannst Du Dir herunterladen und auf einen wie folgt vorbereiteten USB Stick packen:



Diesen Stick steckst Du dann an den Rechner und wählst ihn im Bios Bootmenu aus. Wenn alles geklappt hat sollte Dich der Bootpicker von OpenCore begrüßen.

EDIT: Ich habe die EFI nochmal ausgetauscht weil mir noch zwei kleine Fehler aufgefallen sind sorry dafür 😞

Beitrag von „whatthefaq“ vom 28. März 2021, 10:17

Hier sind zwei Bilder die grt angefragt hat und meinte es wäre gut sie hier zu teilen.

EDIT: Irgendwie wollen die Bilder nicht laden ? Dabei sind es keine großen Dateien, liegt das an mir oder passiert sowas manchmal? Konnte die Bilder leider auch als zip nicht schicken bisher...aber die text Daten von dem einen kann ich zumindest schonmal rauskopieren.

EDIT 2: Ich habe grade den PC zum ersten mal wieder neu gestartet um zu überprüfen was grven gesagt hat bezüglich der Basis. Leider startet er jetzt nicht mehr 😞 Ich komme nachwievor ins Clover menü aber wenn ich den mac starten will lädt er kurz zeigt das Apple zeichen dabei aber bricht dann ab und bleibt bei einem black screen ohne dass ich noch irgendwas machen kann außer ausschalten. Hoffe das kann man schnell beheben weil ich den PC normalerweise für Illustrationsarbeiten für das Musiklabel meines Bruders nutze.

diskutil list

/dev/disk0 (internal, physical):

#: TYPE NAME SIZE IDENTIFIER

0: GUID_partition_scheme *1.0 TB disk0

1: EFI EFI 209.7 MB disk0s1

2: Apple_HFS Secondary Drive 999.9 GB disk0s2

/dev/disk1 (internal):

#: TYPE NAME SIZE IDENTIFIER

0: GUID_partition_scheme 1.0 TB disk1

1: EFI EFI 209.7 MB disk1s1

2: Apple_APFS Container disk4 1000.0 GB disk1s2

/dev/disk2 (internal, physical):

#: TYPE NAME SIZE IDENTIFIER

0: GUID_partition_scheme *6.0 TB disk2

1: EFI EFI 209.7 MB disk2s1

2: Apple_HFS Time Backup 6.0 TB disk2s2

/dev/disk3 (internal, physical):

#: TYPE NAME SIZE IDENTIFIER

0: GUID_partition_scheme *6.0 TB disk3

1: EFI EFI 209.7 MB disk3s1

2: Apple_HFS Backup 6.0 TB disk3s2

/dev/disk4 (synthesized):

#: TYPE NAME SIZE IDENTIFIER

0: APFS Container Scheme - +1000.0 GB disk4

Physical Store disk1s2

1: APFS Volume Main Drive 506.6 GB disk4s1

2: APFS Volume Preboot 87.4 MB disk4s2

3: APFS Volume Recovery 519.0 MB disk4s3

4: APFS Volume VM 20.5 KB disk4s4

/dev/disk5 (external, physical):

#: TYPE NAME SIZE IDENTIFIER

0: FDisk_partition_scheme *62.0 GB disk5

1: Apple_HFS USB 62.0 GB disk5s1