

Erledigt

Wechsel von Clover auf OpenCore

Beitrag von „mamids“ vom 16. Dezember 2019, 14:26

Hallo zusammen,

ich nutze schon sehr lange dieses Forum als Mitleser. Mein Hackintosh arbeite auch ziemlich zuverlässig und zumeist bekomme ich die macOS Updates gut hin. Die Grundinstallation hat jedoch ein Bekannter gemacht, der leider nicht mehr zur Verfügung steht. Die Clover Config und auch weitere Details sind für mich daher ein schwarzes Tuch.

Jetzt würde ich gerne auf Catalina wechseln und im Zuge dessen, schauen lassen, inwieweit meine Konfiguration stimmig ist und auch gleich auf OpenCore wechseln.

Zurzeit nutze ich Mojave mit einer AMD RX 570 8 GB mit 3 Monitoren, das sollte auch unter Catalina noch so laufen.

Freue mich sehr über Hilfe und/oder Tipps.

Anbei habe ich den Systemreport vom KU.

Danke und Gruß

Beitrag von „julian91“ vom 16. Dezember 2019, 14:30

Wenn du schon Clover am Laufen hast , geht auch OC.

das macht keinen unterschied 😊

Du kannst deine Clover Config als Richtungsweißer benutzen für OC

Ich empfehle dir diesen Guide für OC

<https://khronokernel-2.gitbook...re-vanilla-desktop-guide/>

Beitrag von „bluebyte“ vom 16. Dezember 2019, 16:07

[mamids](#) ... wenn Clover schon ein schwarzes Tuch für Dich ist, den man komfortabel mit dem Clover-Configurator konfigurieren kann, dann würde ich erst einmal die Finger von OpenCore lassen. Da mußst

Du die Konfiguration mit XCode, mit einem Texteditor oder einem JSON-Editor bewerkstelligen. Selbst

eingefleischte Hackintosher haben da ihre Probleme. Wenn Clover gut läuft, dann bleib dabei. Ich würde

Dir aber dringend dazu raten vor dem Upgrade auf Catalina ein Update von Clover, der EFI-Driver und

der Kernel-Extensions zu machen.

Beitrag von „julian91“ vom 16. Dezember 2019, 16:44

Das Clover einfacher ist als OpenCore würde ich nicht behaupten.

Clover ist mehr verbreitet ja, dadurch vermutlich auch mehr Leute die Support geben können aber das ändert nichts an der Tatsache das man per Configurator und kaum einer gescheiten Dokumentation seitens Clover einfach auch viel Mist setzen kann nach dem Motto viel hilft viel.

Z.b hatte ich einen Haufen ACPI Patches seitens Clover , die ich wohl angeblich brauchen würde .

Mit diesen wollte OC nicht mal starten , rausgenommen und siehe da das System läuft sauber.

Da stört mich einfach auch diese "ich schmier mal meinen Arm mit Vaseline ein und ramme mal alles Durchn egal ob es geht oder nicht" Methodik von Clover auch ein wenig

Ich selber würde mich nicht als eingefleischter Hackintoshler betrachten auch wenn ich 4 Systeme am laufen habe aber ich komme doch mit OpenCore trotz Editor besser zurecht als mit Clover. Durch die Dokumentation und die guten Start Guides die alles Sauber erklären lässt sich schnell eine Plist erstellen.

Besser noch, man MUSS sich informieren in Dokumentation oder in den Guides was welche Funktion macht wenn man diese Auf YES Stellt oder einen wert einträgt , was einem wiederum hilft das ganze zu Verstehen.

Dazu kommt das ALLE meine Systeme mit OpenCore einfach besser und flinker laufen und diesen eindruckt habe nicht nur ich.

Was mich massiv gestört hat das die Änderungen in Clover mit der Ordner Struktur und treiber sehr viel Chaos im EFI verursacht hat und ich mit Clover sogar Probleme hatte Catalina zum laufen zu bekommen was wiederum mit OC einfach fast von Hand lief.

Und das größte Manko seitens Clover ist einfach die veraltete Kext-Injection die es nutz. Irgendwann wird Tag X kommen und das wird dicht sein und dann Schauen viele Clover User in die Röhre.

Will hier nun keine Grundsatzdiskussion auslösen, jeder muss für sich selber wissen was er tut und was er nutzt vor allem.

Allerdings sehe ich in OC eher die Zukunft als in Clover 😊

Beitrag von „bluebyte“ vom 16. Dezember 2019, 17:10

[julian91](#) ... ich habe mich auch schon mit OpenCore versucht. Am Wochenende durchgängig zwei Stunden. Anfangs mit Texteditor, dann mit Xcode und zuletzt mit dem Plist-Editor aus dem App-Store der eigentlich reicht. Obwohl ich sagen muss, das Copy&Paste im Editor wesentlich einfacher geht. Und was Clover betrifft, da halte ich mich mit Einträgen dezent

zurück. Steht zwar manches drin, aber vieles ist auf "disabled". In der Kürze liegt die Würze.

Beitrag von „mamids“ vom 16. Dezember 2019, 19:45

Vielen Dank schon mal ihr beiden, ich wollte aber keine Grundsatzdiskussion hier auslösen. julian91 hat nicht ganz unrecht, da mein Bekannter nicht mehr zur Verfügung steht, habe ich mir Clover etwas genauer angeschaut, und parallel dazu OpenCore. Mein Englisch ist nicht so gut ... Wie dem auch sei, die Dokumentation zu OpenCore ist IMHO besser als die von Clover.

Da ich nicht ganz so viel Zeit habe, tendiere ich zu bluebytes Vorschlag, mit meiner laufenden Clover-Config das Update probieren und mich dann etwas mehr mit OpenCore beschäftigen.

Ich hatte auch eigentlich gehofft, dass sich jmd. mal meine Clover-Config, die ich mit hochgeladen habe, anschaut und evtl. auf obsoletes und/oder falsches hinweist.

Beitrag von „bluebyte“ vom 16. Dezember 2019, 20:27

[mamids](#) ... das im Systemreport der Clover versteckt ist, dass wußte mit Sicherheit niemand von uns.

Beitrag von „julian91“ vom 16. Dezember 2019, 20:35

Wenn ich hier fertig bin schau ich mal drüber ... vielleicht hab ich ja lust und bau dir mal Grund OC 😄 aber heist nicht das ich da dauersupport geben werde 😄

Beitrag von „mamids“ vom 16. Dezember 2019, 20:40

Das wäre super [julian91](#) mir geht es auch nicht um Dauersupport, sondern um einen Clean-Einstieg. 😁 bin auch schon an der OpenCore Config dran ... mal schauen

Beitrag von „julian91“ vom 16. Dezember 2019, 21:58

Kannst du mir zufällig mal deine macAdresse deines Lanports nennen ? bräuchte ich für die config 😊

Beitrag von „mamids“ vom 16. Dezember 2019, 22:08

Gerne, siehe PN 😊

Beitrag von „karacho“ vom 16. Dezember 2019, 22:14

[Zitat von bluebyte](#)

das im Systemreport der Clover versteckt ist, dass wußte mit Sicherheit niemand von uns.



Beitrag von „apfelnico“ vom 16. Dezember 2019, 22:19

Doch, doch. Hatten wir schon länger drüber geschrieben ...

Beitrag von „julian91“ vom 16. Dezember 2019, 22:25

So .. hab hier mal was für dich zum Probieren [mamids](#)

Bitte deine Clover EFI Saven und am besten das ganze per USB stick testen also vom EFI des Stick booten 😊

Hab mal recht Minimal gehalten ein wenig , paar ACPI Patches erst mal rausgelassen , ich traue da clover immer nicht so was nötig ist und was nicht.

hab dir verbose aktiviert das man schauen kann wo er hängt falls er nicht booten will damit 😊

Die anderen dürfen gerne mal drüber schauen , ich selber bezeichne mich nun nicht als Profi und vielleicht hat der eine oder Andere doch noch ne Idee was doch nicht so gut ist:-)

Beitrag von „mamids“ vom 16. Dezember 2019, 22:28

Vielen Dank! Ich teste es gleich mal und gib Feedback ...

Beitrag von „karacho“ vom 16. Dezember 2019, 22:31

Ob das mit der USBPorts.kext klappt? 🤔 Zum testen hätte ich ihm USBInjectall reingetan und den XhciPortLimitPatch auf YES gesetzt.

Beitrag von „julian91“ vom 16. Dezember 2019, 22:31

ich hab damals meine aus Clover direkt übernehmen können, daher sag ich da mal ja

notfall bau ich das schnell um 😊

Beitrag von „karacho“ vom 16. Dezember 2019, 22:32

Er hat aber ein anderes Board als du.

Beitrag von „julian91“ vom 16. Dezember 2019, 22:33

ich hab seine USBPort.Kext aus Clover übernommen.

hab mich natürlich an seinen clover ordner gehalten 😊

Beitrag von „karacho“ vom 16. Dezember 2019, 22:34

Oh...wo ist denn sein Clover Ordner? Den sehe ich nirgends.

Beitrag von „julian91“ vom 16. Dezember 2019, 22:36

in seinem systemreport ordner mit enthalten 😊 waren zwar noch haufen anderer Kexte enthalten die ich aber laut recherche nun mal hinten angestellt habe ... aber gern korrigieren wenn das falsch war

Beitrag von „karacho“ vom 16. Dezember 2019, 22:38

Dann sollte es mit dem USBPorts.kext auch unter OC funktionieren, wenn er den unter Clover schon genutzt hat. Ich dachte zuerst du hättest ihm deinen installiert. Sorry für die Verwirrung.

Beitrag von „julian91“ vom 16. Dezember 2019, 22:39

alles gut 😁 ich hab wie gesagt ein wenig was abgespeckt aber mal die wichtigsten die drinnen waren auch wieder hinzugefügt allerdings natürlich alle aktuell per Updater

Beitrag von „karacho“ vom 16. Dezember 2019, 22:42

Hättest evtl noch die SSDT-EC.aml oder die SSDT-USBX.aml auf YES setzen können. Aber das sieht er ja dann wenn es net bootet.

Beitrag von „mamids“ vom 16. Dezember 2019, 22:44

Also,

OC: Configuration requires vault but no vault provided!

Mehr kommt leider nicht

Beitrag von „julian91“ vom 16. Dezember 2019, 22:44

FUCK

einmal dachte ich, ich hätte nicht diesen dämmlichen YES vergessen ... IMMER bei ACPI passiert mir das :'D

edit :

ja hab da noch was vergessen , moment bekommst gleich nen update

edit 2:

bitte die versuchen ... manchmal frag ich mich wo mein kopf ist

Beitrag von „karacho“ vom 16. Dezember 2019, 22:50

[mamids](#) Beim ersten Start mit OC (wenn man von Clover wechselt) erstmal einen NVRam Reset machen.

Beitrag von „julian91“ vom 16. Dezember 2019, 22:52

timeout hab ich mal auf 10sec gestellt.

willst du schneller booten einfach die zahl des boot eintrags drücken auf der tastatur, aber davor NVRAM reset , wird auch im Boot menü angezeigt 😊

Beitrag von „mamids“ vom 16. Dezember 2019, 23:08

gerade war dieser fehler ... hatte aber kein NVRAM reset gemacht, mache ich jetzt ...



Beitrag von „karacho“ vom 16. Dezember 2019, 23:27

Alle .aml im Abschnitt ACPI->Add auf NO setzen, außer der SSDT-EC.aml...die bleibt auf YES

Edit: Teste mal diese config.

Hab das mal mit dem Guide abgeglichen -> <https://khronokernel-2.gitbook...uide/config.plist/skylake>

Bei weiteren Fehlern guckst du hier -> <https://khronokernel-2.gitbook...eshooting/troubleshooting>

Beitrag von „julian91“ vom 16. Dezember 2019, 23:30

hast du was geändert ?

hab den Guide auch genommen ...

Beitrag von „mamids“ vom 16. Dezember 2019, 23:31

danke karacho, ich teste sie sofort.

hier mal ein bootlog von meiner aktuellen clover version, evtl. hilft das ja.

[bootlog.txt](#)

edit: hatte geändert, aber immer noch selber fehler, probiere jetzt die plist von karacho

Beitrag von „bluebyte“ vom 17. Dezember 2019, 00:05

Mit PlistEditor wie verrückt gesucht. Mit Editor in 3 Sekunden den Fehler gefunden.

FileVault = True

Signature = True

Schon zweimal versucht zu starten.

OC: FileVault

Warum ist das in der Sample standardmäßig auf True?

Wer benutzt FileVault?

Beitrag von „julian91“ vom 17. Dezember 2019, 00:06

[bluebyte](#)

hab ich ihm schon rausgenommen ...

aber ja ich vergess das auch immer 😄

Beitrag von „bluebyte“ vom 17. Dezember 2019, 00:07

[julian91](#) Ich rede von meiner plist

Schließt Require Vault nicht Require Signature mit ein?

Beitrag von „mamids“ vom 17. Dezember 2019, 00:21

Kurze Status-Meldung. SSD wurde nicht mehr erkannt, auch nicht nachdem ich sie ausgebaut und ans Macbook drangehängt habe.

Ich schau mal ob ich mir ne neue besorge und die Daten evtl. noch gerettet bekomme. Melde mich wieder, wenn das System wieder läuft.

Beitrag von „julian91“ vom 17. Dezember 2019, 10:30

Klingt aber nicht so gut !

meld dich einfach dann 😊 dann gehen wir das thema weiter an

Beitrag von „griven“ vom 17. Dezember 2019, 11:01

[Zitat von bluebyte](#)

FileVault = True

Signature = True

Schon zweimal versucht zu starten.

OC: FileVault

Das hat nichts aber auch rein gar nichts mit FileVault2 zu tun.

Leute ernsthaft bei Clover heulen alle rum das es keine wirklich brauchbare Dokumentation der Funktionen und Schalter gibt bei OpenCore gibt es die aber gelesen wird so offensichtlich trotzdem nicht. Die Vault und Signature Werte (RequireVault, RequireSignature) beziehen sich **NICHT** auf die FileVault2 Verschlüsselung von macOS sondern bieten eine ziemlich effektive Möglichkeit mögliche Korruptionen an Files auf der EFI Partition zu erkennen und im Falle eines Falles den Systemstart zu verweigern. Das diese beiden Optionen per default enabled sind ist nur konsequent und in meinen Augen auch richtig denn hierbei geht es um ein sicherheitsrelevantes Feature das einen Systemstart mit korrumpierten Booter Files effektiv unterbindet. Also bitte schreibt nicht einfach drauf los sondern freut Euch das es für OpenCore eine umfassende Dokumentation gibt und lest diese auch bevor ihr Euch in Spekulationen ergeht und damit möglicherweise Fehlinformationen streut die sich am langen Ende nachhaltig festsetzen...

Beitrag von „mhaeuser“ vom 17. Dezember 2019, 11:07

[griven](#) mit "Secure Boot"-Unterstützung (nächstes Jahr) kann man dann auch eine volle Kette aufbauen...

Boot Guard -> UEFI Secure Boot -> OC Vault -> Apple Secure Boot -> [SIP](#)

Beitrag von „julian91“ vom 17. Dezember 2019, 11:55

[griven](#)

Da ich mich noch tiefer damit befasst hatte, hatte ich es ausgelassen.

Hatte zwar die Vermutung das es mit dem EFI zu tun hatte , aber da ich davon keine Ahnung habe lass ich auch erst mal die finger von:-)

Und ich freu mich über die Umfassende Dokumentation , bin immer wieder am nachlesen , allerdings war ich beim Thema Vault noch nicht angekommen:-)

Beitrag von „griven“ vom 17. Dezember 2019, 12:19

Ging auch weniger an/gegen Dich [julian91](#) sondern war eher allgemein gesprochen 🤔

Gerade die Punkte RequireVault und RequireSignature werden ja gerne und oft fälschlicherweise mit FileVault in Verbindung gebracht und da ist es meiner Meinung nach wichtig aufzuklären wo immer dieser Irrglaube auftaucht. Ich finde es eben wirklich kritisch wenn ein Sicherheitsrelevantes Feature abgetan wird mit "Wer braucht schon FileVault" und das aus Unwissenheit bzw. aus dem Versäumnis heraus sich in der Dokumentation zu informieren. Das Problem dabei ist das sich solche Aussagen nachhaltig in den Köpfen festsetzen denn sind wir mal ehrlich die Gro der Leute liest eher Threads in Foren und vertraut den Aussagen "versierter" User als sich durch die Dokumentation zu "quälen".

Ein gutes Beispiel dafür ist auch das Thema [SIP](#) denn auch hier wurde in den diversen Foren, auch hier, proklamiert das man diese am Hack tunlichst immer zu deaktivieren hat. Dieser Unsinn hat sich so nachhaltig in den Köpfen der User festgesetzt das bis heute gefühlt 99,8% der Leute mit deaktivierter [SIP](#) unterwegs sind und so wissentlich ein mächtiges Sicherheitsfeature des Systems aushebeln obwohl dies ganz und gar nicht (dauerhaft) notwendig ist.

Ich finde diesen, doch sehr laxen, Umgang mit sicherheitsrelevanten Themen den die Community "pflegt" ehrlich gesagt zunehmend bedenklich.

So nun aber back to Topic denn schließlich geht es in dem Thread ja nicht um eine Grundsatzdiskussion über Sicherheitskonzepte in Bootern und/oder Betriebssystemen sondern um die Frage wie [mamids](#) nun am elegantesten den switch von Clover zu OpenCore bewerkstelligen kann 😊

Beitrag von „apfelnico“ vom 17. Dezember 2019, 12:57

Zitat von Download-Fritz

... mit "Secure Boot"-Unterstützung (nächstes Jahr) kann man dann auch eine volle Kette aufbauen ...

Boot Guard -> UEFI Secure Boot -> OC Vault -> Apple Secure Boot -> [SIP](#)

Das ist prima, ist doch "Secure Boot" für eine UEFI-Windows Installation schon sinnvoll, leider klappt das dann nicht mit Clover. Insofern freue ich mich, dass OpenCore sich dem annimmt und sowohl Windows wie auch macOS in dieser Umgebung gemeinsam arbeiten können.

Beitrag von „mamids“ vom 17. Dezember 2019, 13:28

Danke [griven](#) ,

hier mal ein kleines Update.

Ich hatte zum Glück noch eine Kopie meines aktuellen EFIs und eine alte SSD da und konnte so noch Catalina installieren. Bin noch nicht komplett durch mit der Einrichtung, aber es scheint bislang alles stabil zu laufen.

Warum meine SSD nach dem NVRam Reset nicht mehr erkannt wird, kann ich mir jedoch

immer noch nicht erklären. Einen physischen Defekt kann man, denke ich, ausschließen. Sie wird einfach nicht mehr erkannt, weder im UEFI noch am MacBook Pro. Werde mal schauen ob ich am We weiter komme, wäre schade um die verlorenen Daten.

Heute morgen habe ich mir noch eine neue SSD bestellt, wenn die da ist werde ich die aktuelle SSD klonen und auf der jetzigen weiter mit OC probieren.

Sobald ich mit der Einrichtung fertig bin, lade ich die aktuelle Conf hoch. Dann bekommen wir das sicher hin 😊 und vielleicht hat noch jmd eine Idee zur Rettung der SSD.

Beitrag von „bluebyte“ vom 17. Dezember 2019, 13:38

[griven](#) ... ich habe mir die Anleitung durchgelesen. Ich habe OpenCore konfiguriert während ich nebenbei Deinen Workshop auf Youtube von Anfang bis Ende angeschaut habe. Da ich öfter mal zurückgespult habe und meine Eingaben getätigt habe, kam ich am Ende auf gut drei Stunden. Ich habe mir die Anleitung durchgelesen, aber ich habe noch nicht alles verstanden. Im Gegensatz zu manch anderen schreie ich hier nicht rum: "Mach mir mal eine EFI!" Sondern ich bin mit Interesse bei der Sache. Mittlerweile kompiliere ich mir die Sachen selbst. Habe mich informiert, warum eine Debug-Version größer ist als eine Release-Version. Schau mir die Repositories auf github an. Ich stehe auf dem Standpunkt: "Was man sich selbst erarbeitet hat, dass kann einem keiner mehr nehmen!" Das gilt sowohl im privaten Bereich als auch auf der Arbeit. Ich habe im Internet recherchiert und bin dann bei Insanelymac auf diesen Kommentar gestossen.

[https://www.insanelymac.com/fo...ndComment&comment=2677706](https://www.insanelymac.com/forum/thread/45608-wechsel-von-clover-auf-opencore/)

Tut mir leid, dass ich da etwas verwechselt habe. Ich verstehe das jetzt so, dass man nicht so einfach die EFI kopieren kann, sondern man muss mit dem script "create_vault.sh" erst eine "vault.plist" anlegen. Richtig?

Beitrag von „bananaskin“ vom 17. Dezember 2019, 13:53

Hallo [mamids](#), wenn deine SSD von Samsung ist.... einfach mal die Firmware neu flashen, (wenn möglich)

da gibt es ein Tool dazu, läuft unter Windows,

Tante Google findet das schon

Samsung SSD Magician Software

Beitrag von „griven“ vom 17. Dezember 2019, 14:29

Ja [bluebyte](#) im großen und ganzen bedeutet es genau das wobei Du den EFI Ordner schon kopieren kannst solange eben die Signatur und die vault.plist mit kopiert werden und im Prozess keine Files verändert werden.

Das create_vault.sh Script erzeugt eine vault.plist Datei in der die SHA-256 hashes aller Files enthalten sind die zu Deiner OC Installation gehören (OC Files, Configs, Extensions usw.) und signiert diese Datei im Anschluss mit RSA Zertifikat und "implantiert" den entsprechenden Key in die OpenCore.efi. Wenn die Optionen RequireVault und RequireSignature in der config aktiviert sind liest OpenCore beim start die vault.plist, natürlich nicht ohne im Vorfeld die Signatur auf Gültigkeit zu prüfen, und gleicht dann die in der Vault.plist abgelegten Hashes mit den Hashes der Dateien auf der EFI ab. Sollte sich vom Zeitpunkt der Erstellung des Zertifikats und der Vault.plist etwas an den Dateien oder an der Vault.plist verändert haben egal ob wissentlich durch den User oder unwissentlich durch Schadprogramme verweigert OpenCore den Systemstart mit einem entsprechenden Hinweis. Aus dem Verhalten ergibt sich natürlich das es nur dann sinnvoll ist beide Komponenten wirklich zu aktivieren wenn man die konfiguration abgeschlossen hat denn ansonsten würde jede Änderung an der config.plist oder irgendeiner anderen Datei im OC Kontext bedingen das die Vault.plist neu aufgebaut und signiert werden muss.

~~In wie weit in dem Zug die OpenCore.efi ebenfalls gegen eine Version getauscht werden muss die noch keinen Public Key enthält kann ich adhoc nicht beantworten würde aber anhand der Dokumentation davon ausgehen das dem so ist weil der Suchbegriff den das Script verwendet um den Key in den EFI Treiber zu schreiben nicht mehr vorhanden ist~~

. [mhaeuser](#) liege ich da mit meiner Annahme richtig?


Edit: ich lag falsch mit der Annahme der PublicKey bleibt der gleiche sprich die OpenCore.efi muss nicht erneut angepackt werden wohl aber muss die Vault.plist nach jeder Änderung an config oder Files neu erstellt und signiert werden. Danke [mhaeuser](#) fürs klarstellen weiter unten 😊

Richtig spannend wird das Thema indes auch erst wenn die SecureBoot implementation komplett ist denn aktuell greift das ganze ja erst wenn OC schon gestartet ist alles was davor passiert (kompromitierte bootX64.efi zum Beispiel) würde so erstmal weitestgehend unbemerkt durchgehen da aktuell noch eine Instanz fehlt die schon auf dieser Ebene prüft.

Beitrag von „mamids“ vom 17. Dezember 2019, 14:32

Danke @[worschdsupp](#) , SSD ist leider von SanDisk

Beitrag von „bluebyte“ vom 17. Dezember 2019, 14:37

Ich hoffe, ich mach das jetzt richtig mit dem vault ...[griven](#) ... vielen herzlichen Dank, dass Du mich auf den Weg gebracht hast. Jetzt habe ich es kapiert. 

Beitrag von „mhaeuser“ vom 17. Dezember 2019, 14:42

[griven](#) nein, der PK bleibt ja gleich... das Schlüsselpaar soll ja nicht jedes Mal gelöscht werden

Beitrag von „bluebyte“ vom 17. Dezember 2019, 20:22

[mamids](#) ... egal ob Sandisk oder Samsung. Ich habe WD, Sandisk und Samsung.

Das Tool für Sandisk gibt es im Netz.

Bootcamp lässt sich schon mal über OpenCore starten.

HighSierra startet. Bricht aber ab. Ohne Error. Fehlermeldung beim Laden von "AtherosE2200Ethernet.kext"

Catalina reagiert gar nicht.

Bei HighSierra "Normalize Headers"? Laut Video von griven soll das ja so.

Beitrag von „mamids“ vom 17. Dezember 2019, 21:07

Danke [bluebyte](#) ,

habe ich schon gefunden - <https://kb.sandisk.com/app/ans...re-update-version-x2316r> Bei mir handelt es sich um eine **SDSSDH3-XXX**, die dort leider nicht vertreten ist. Anderswo konnte ich auch noch nichts finden.

Beitrag von „bluebyte“ vom 17. Dezember 2019, 21:26

[mamids](#) ... hatte das vorhin auch bei CHIP gefunden

https://www.chip.de/downloads/...D-Dashboard_83057942.html

Vielleicht findest Du auch was bei Western Digital. Die sind doch mit Sandisk fusioniert.

Beitrag von „mamids“ vom 17. Dezember 2019, 21:29

[bluebyte](#) ich werde das wenn, über einen Linux Bootstick machen, Windows habe ich schon lange nicht mehr 😊

Beitrag von „bluebyte“ vom 17. Dezember 2019, 21:32

[mamids](#) ... war auch schon mal für ein Jahr ohne Windows. Linux und MacOSX 10.4.9 Tiger.

Vorhin bekam ich einen Schreck. Wollte über OpenCore mein HighSierra starten. Wollte nicht.

Bin dann ins BIOS. Oh Schreck. SSD nicht mehr vorhanden. Mußte sofort an Dich denken.

Nach einem Rechnerneustart war sie aber wieder da.

Beitrag von „SPIDER“ vom 17. Dezember 2019, 22:44

Halo zusammen,

ich lese hier gerade fleißig mit und bin auch selbst schon ein bisschen im Umbruch von Clover zu Open Core, habe aber mit dem letzteren noch etwas Probleme.

[bluebyte](#) Vielen Dank für Deine Beiträge in diesem Thema, dadurch kenn nun auch ich jetzt die Bedeutung der "Require Signatur + Require Vault". Hab jetzt auch eine Signatur in meinem EFI-Ordner und es läuft soweit alles. [griven](#) Du hast es auf dem Punkt getroffen, auch ich war wegen "hören/sagen" bis vor kurzem der Meinung das die auf "NO" stehen müssen, aber ohne es wirklich zu hinterfragen. Ich würde behaupten jeder der nicht allzu komplizierte Hardware hat, sollte mit diesem Thread OpenCore zum laufen bekommen.

Also noch ein riesiges DANKESCHÖN an dieses Großartige FORUM und allen die daran teil haben.



Beitrag von „karacho“ vom 17. Dezember 2019, 22:52

[mamids](#) manchmal setzt ein Nvram Reset das BIOS gleich mit zurück. Dann brauchst du den RTC fix. Schau im BIOS mal nach, ob deine Einstellungen noch stimmen. Festplatten Controller -> AHCI etc. Lies dazu bitte die beiden Links ich weiter oben geteilt habe

Beitrag von „bluebyte“ vom 18. Dezember 2019, 19:15

Für alle die mit vault arbeiten ist es manchmal nervig die Kommandos in der Shell einzutippen oder sich mit den Pfeiltasten durch die history zu quälen. Viele mögen es wissen, manche auch nicht, wie man sich das ganze vereinfachen kann.

Im Terminal den Befehl "history" (ohne die Quotes, Gänsefüßchen) eingeben.

Es erscheint eine Tabelle mit Nummern und den gespeicherten Befehlen.

Jetzt braucht man nur noch ein Ausrufezeichen mit der Nummer aus der History eingeben und der Befehl wird ausgeführt.

Brauch mir so nur !503 und !504 merken.

Habe mich die ganzen Tage mit OC beschäftigt. Video von griven vorwärts und rückwärts. Die Anleitung gelesen. Mir bei Insanelymac einige Sachen durchgelesen. Bei Acidanthera auf guitarhub Notizen gemacht. Die config.plist rauf und runter editiert - fast schon ohne hinzuschauen.

Habe heute gefrustet das Handtuch geworfen. Ich schicke mal die confi.plist hoch. Den Rest braucht man ja nicht, da OC nur das lädt was in der config.plist gewählt ist.

Bei der SSDT wusste ich nicht wie das mit OSY-Weiche funktioniert.

Beitrag von „apfelnico“ vom 18. Dezember 2019, 19:46

[Zitat von bluebyte](#)

Bei der SSDT wusste ich nicht wie das mit OSY-Weiche funktioniert.

Gibt viele Wege. Hier ein ganz eleganter:

1. Zuerst eine "OSDW"-Methode erstellen und oben in der SSDT platzieren (oder ausserhalb in einer anderen, dann verlinken):

Code

1. Scope (\)
2. {
3. Method (OSDW, 0, NotSerialized)
4. {
5. If (CondRefOf (_OSI, Local0))
6. {
7. If (_OSI ("Darwin"))
8. {

- 9. Return (One)
- 10. }
- 11. }
- 12.
- 13. Return (Zero)
- 14. }
- 15. }

Alles anzeigen

Was passiert hier? OSDW wird abgearbeitet, wenn "Darwin" erkannt, dann bekommt "OSDW" den Wert "1", ansonsten "0".

Wieso "OSDW"? Ist in der Apple ACPI zu finden und wird dort "zweiteilig" ermittelt:

Code

```

1. Method (OSDW, 0, NotSerialized)
2. {
3. If ((OSYS == 0x2710))
4. {
5. Return (One)
6. }
7. Else
8. {
9. Return (Zero)
10. }
11. }
12.
13. Method (PINI, 0, NotSerialized)
14. {
15. OSYS = 0x07DC
16. If (CondRefOf (_OSI, Local0))
17. {
18. If (_OSI ("Darwin"))
19. {
20. OSYS = 0x2710
21. }
22. ...

```

Alles anzeigen

... das brauchen wir nicht, ist zu umständlich. Eine generelle Abfrage der Systeme ist schon in der DSDT enthalten.

Warum benutze ich "OSDW" und schreibe nicht "(_OSI ("Darwin"))" direkt rein? Weil ich's elegant haben will. Etliche modifizierte Apple SSDT haben auch abfragen nach "OSDW" drin und diese werden dann auch beantwortet.

Jetzt also die Abfrage, ob ein bestimmtes Device bei einem bestimmten System genutzt werden soll oder nicht:

Code

```

1. Scope (TATA)

```



```

2. {
3. Method (_STA, 0, NotSerialized) // _STA: Status
4. {
5. If (OSDW ())
6. {
7. Return (0x0F)
8. }
9. Else
10. {
11. Return (Zero)
12. }
13. }
14.
15. Method (_DSM, 4, NotSerialized) // _DSM: Device-Specific Method
16. {
17. ...

```

Alles anzeigen

Hier finden wir die Methode `_STA` (Status) und legen fest, dass das vorhandene Device "TATA", welches wir mit "Scope" weitergehend beschreiben, je nach "OSDW"-Wert ein "0x0F" (aktiv) oder "Zero" (deaktiviert) bekommt. Im Anschluss dann eine `_DSM`-Methode, um in das Device weitere Eigenschaften zu injecten.

Natürlich kann ich mir "OSDW" auch als Methode sparen, und schreibe gleich:

Code

```

1. Scope (TATA)
2. {
3. Method (_STA, 0, NotSerialized) // _STA: Status
4. {
5. If (_OSI ("Darwin"))
6. {
7. Return (0x0F)
8. }
9. Else
10. {
11. Return (Zero)

```

```
12. }
13. }
14.
15. Method (_DSM, 4, NotSerialized) // _DSM: Device-Specific Method
16. {
17. ...
```

Alles anzeigen

Beitrag von „Tirola“ vom 19. Dezember 2019, 16:05

[julian91](#) ich bin gerade auch am versuchen von Clover auf OpenCore zu wechseln. Aber spätestens bei der Sample.plist hänge ich total.

Kann ich das einfach aus meiner Config.plist von Clover übernehmen?

beim ersten Teststart häng ich hier.

Beitrag von „bluebyte“ vom 19. Dezember 2019, 17:43

[Tirola](#) steht doch auf dem Bildschirm was fehlt. Der Treiber, den Du bei item 0 angegeben hast. Ansonsten lies die Anleitung und schau Dir das Video von griven auf Youtube an.

Die sample.plist musst Du natürlich noch in config.plsit umbenennen und entsprechend mit einem Plist-Editor, mit dem XCode-Editor oder einem vergleichbaren JSON-Editor anpassen. Freaks nehmen auch den Textedit oder Nano.

Beitrag von „Tirola“ vom 19. Dezember 2019, 18:09

Umbenannt habe ich sie natürlich. Der treiber ist aber drin im Ordner. Das versteh ich nicht

Beitrag von „bluebyte“ vom 19. Dezember 2019, 19:05

[Tirola](#) schick mal einen Screenshot vom Ordner. Wie sieht es aus wenn Du den Opensource-Treiber VBoxHfs.efi nimmst?

Beitrag von „Tirola“ vom 19. Dezember 2019, 19:34

Das war mein Fehler, hatte da einen fehler beim kopieren gemacht.

jetzt hänge ich etwas weiter

Beitrag von „julian91“ vom 19. Dezember 2019, 19:36

Du solltest dir echt mal die Tutorials richtig durchlesen .. einfach 1:1 kopieren geht nicht !
Sieht man alleine an deinem Drivers Ordner. so kann das nichts werden

edit :

genauso , wieso fährst du mit ner master version von OC und nicht mit der Release 0.5.3 ?

Beitrag von „Tirola“ vom 19. Dezember 2019, 19:47

Ich habe es so gemacht wie im tutorial beschrieben und mit dem macbuild.tool hat mit das terminal das runter geladen.

Und mein driver ordner ist so von clover übernommen.

Beitrag von „bluebyte“ vom 19. Dezember 2019, 19:54

Wie macht man Screenshots? Um einen bestimmten Bereich auszuwählen, drückt man Shift-Windowstaste-4. Aus dem Cursor wird ein Fadenkreuz. Mit dem Fadenkreuz kann man nun einen bestimmten Bereich auswählen. Zuerst den Anfangspunkt mit dem Cursor ansteuern. Linksklick und die Maustaste festhalten. Dann den gewünschten Bereich mit der gedrückten Maustaste auswählen. Die Maustaste loslassen. Der gewünschte Bereich wird gespeichert. Ist auf jeden Fall besser als Handyfotos.

@[Tirola](#) ... und mit welchem der genannten Editoren hast Du die config.plist bearbeitet? Denn da beginnt die eigentliche Arbeit.

Beitrag von „julian91“ vom 19. Dezember 2019, 19:59

[Tirola](#)

hast du dich überhaupt mal damit befasst WELCHE Treiber du bei OC nutzen kannst und sollst

Sieht nämlich gar nicht danach aus, z.B. da der aptiofix NICHTS bei OC zu suchen hat sondern in OC enthalten ist. genauso fehlt dir der Treiber der für OC notwendig ist, FWRUNTIMESERVICES ... bitte erst richtig informieren ...

und in welchem Tutorial steht bitte das man alle clover treiber übernehmen soll in OC ?

Beitrag von „Tirola“ vom 19. Dezember 2019, 20:01

Ich nehme xcode dafür. Ok. Ich werde mich weiter einlesen. Wie man screenshots macht weiß ich. Aber im boot screen? Da geht es leider nur so. Sorry das es vorhin via handyfoto gemacht

wurde. Aber da habe ich am handy geschrieben!

Beitrag von „bluebyte“ vom 19. Dezember 2019, 20:07

[Tirola](#) ... macht nichts. Sind aber mal ausnahmsweise Handy-Fotos, auf denen man alles gut erkennen kann. Habe schon schlimmere gesehen.

Beitrag von „Tirola“ vom 19. Dezember 2019, 20:34

funktioniert die Geschichte mit USB noch genauso wie unter Clover?

Er startet jetzt, abgesehen vom Monster Apple Logo und das meine USB Geräte nicht so wollen wie vorher

Beitrag von „Erdenwind Inc.“ vom 19. Dezember 2019, 20:36

dir fehlt wahrscheinlich ein angepassten USBPort.kext?

Beitrag von „Tirola“ vom 19. Dezember 2019, 20:44

ja, danke. das war es

Beitrag von „Darkchild“ vom 19. Dezember 2019, 23:29

[bluebyte](#)

Kennst du schon fzf. Das macht history nahezu überflüssig.

<https://github.com/junegunn/fzf>

Und so einfach installiert

Code

1. `git clone --depth 1 https://github.com/junegunn/fzf.git ~/.fzf`
2. `~/.fzf/install`

Beitrag von „bluebyte“ vom 19. Dezember 2019, 23:59

[Darkchild](#) ... jedem das seine. Ich kenne das noch aus den 80ern aus DOS- und Linux-Zeiten.

Beitrag von „Tirola“ vom 20. Dezember 2019, 10:30

Muss mich nochmal einklinken.

OC merkt sich bei mir nix, liegt also am NVRam. In dem Guide wird ja beschrieben, was man machen muss.

Erst wird gesagt ich solle die nvram.plist löschen und später Enabling emulated NVRAM (with a nvram.plist)

Muss die nvram.plist dann wieder in die Efi rein?

Beitrag von „bluebyte“ vom 25. Dezember 2019, 22:09

Wechsel auf OpenCore abgeschlossen. Läuft!

Dank dem Video von [griven](#) und der tollen Dokumentation.

Beitrag von „karacho“ vom 26. Dezember 2019, 00:10

Super das jetzt alles äuft. Hut ab. Aber ein bisschen mehr Hintergrund, was du überhaupt getan hast, ausser das Profil von griven zu verlinken, wäre doch auch nicht schlecht. Das könnte vielen Menschen helfen, die das gleiche Problem haben.

Beitrag von „griven“ vom 26. Dezember 2019, 10:53

Naja der Knackpunkt an OC ist halt eben das man sich aktuell "noch" ziemlich tief in das Thema einarbeiten muss. Wer mich kennt weiß das ich hier gerne den Vergleich zu den Anfängen von Clover ziehe denn auch dort hat es anfangs eine ziemliche Lernkurve zu meistern gegeben nur mit dem Unterschied das es zu Clover eben keine ausführliche Dokumentation gab und vieles per Trial and Error und mit den dürftigen Ressourcen zum Beispiel auf InsanelyMac gelöst werden musste. Ich kann mich noch zu gut daran erinnern wie sehr die Leute Clover damals verschrien und verrissen haben.

Viel zu komplex, viel zu viele Optionen die niemand versteht dazu ein irrer Konfigurationsaufwand für gefühlt keinen Vorteil. Das Clover per Design einen komplett anderen Ansatz als Chameleon und Co. verfolgt hat hat damals kaum jemand realisiert und schon erst recht kaum jemanden interessiert in den Augen des größten Teils der Community war Clover vor allem eins nämlich umständlich, aufgeblasen und kompliziert. Das alles ist heute freilich Schnee von gestern zumindest im Bezug auf Clover denn inzwischen etabliert gibt es einen Sack voll Tools die einem den Umgang mit der Konfiguration erleichtern und unzählige Quellen im Internet die fertige EFI Ordner für Clover anbieten und wo man sich einfach bedienen kann.

Mit OpenCore verhält es sich heute ganz ähnlich wie mit Clover damals denn auch OpenCore erfordert eine Lernkurve die zu meistern ist bevor es sich so einsetzen lässt wie man es sich

wünscht. Genau wie Clover damals ist OpenCore heute keine leichte Kost sondern es erfordert Engagement und die Bereitschaft ein paar Stunden zu investieren um zu verstehen wie etwas eingestellt wird und warum manche Einstellungen notwendig sind und andere nicht. Anders als bei Clover damals tut man sich mit OpenCore da aber meiner Meinung nach um einiges leichter denn zu OpenCore gibt es eine Dokumentation die alle Optionen haarklein beschreibt man muss sie halt nur lesen 😊

Ich bin mir ziemlich sicher das es mit der Zeit auch zu OpenCore wieder einen Sack voller Tools geben wird die auf der Endnutzer Seite die Konfiguration erleichtern und natürlich wird es über kurz oder lang auch wieder fertige EFI Ordner an jeder Ecke im Internet geben, wo sich der geneigte User bedienen kann, um das ganze ohne viel Aufwand und ohne viel Hintergrundwissen lauffähig auf die Möhre zu zimmern. Ob das mit den Tools und fertigen EFI Ordnern eine gute Sache ist muss jeder für sich selbst entscheiden ich stehe dem inzwischen eher kritisch gegenüber nicht zuletzt deshalb weil jedes Tool das dem Nutzer Konfigurationsarbeit abnimmt und jeder fertige EFI Ordner unweigerlich dazu führt das sich die User eben nicht mit der Materie beschäftigen und so eben nie verstehen werden wie etwas funktioniert und warum es funktioniert. Vielleicht wird man ja mit dem Alter langsam weise aber heute kann ich gut verstehen wenn von der Seite der Entwickler der Verweis auf die Dokumentation kommt wenn es Fragen gibt die dort beantwortet werden denn es macht ja keinen Sinn alles hundert mal zu erklären wenn es dort aufgeschrieben und für jeden nachlesbar dokumentiert ist 😊

Beitrag von „bluebyte“ vom 26. Dezember 2019, 20:35

[karacho](#) und [griven](#) ... ich habe in den vergangenen Tagen viel gelesen und im Internet rechechirt. Abseits des ganzen Trubels hier im Hackintosh-Forum und meiner letzten Kommentare. Eigentlich hatte ich schon die Konfiguration, so wie sie hätte laufen müssen, schon lange auf dem Rechner. Ich habe dann im BIOS die komplette Virtualisation abgeschaltet und EIST auf disabled (default war auf auto). DisableIOMapper=True und dart=0 funktionieren wohl doch noch nicht so richtig wie beschrieben. Die Soundkarte wurde mal erkannt und mal nicht.

Der Rechner läuft jetzt wieder mit dem BIOS F9. Ich kann mich irgendwie nicht mit dem BIOS F11a anfreunden. Inzwischen funktioniert es auch mit eingeschalteter Virtualisation und vt-d sowie mit EIST=Auto. Habe die DSDT gepatcht und mit eingebaut. Hatte gehofft, dass es mit dem Property Injector klappt. Aber irgendwie haut das mal hin und mal nicht. Jetzt sind auch wieder alle USB-Ports erreichbar. Schnell lernt man die Abhängigkeiten der Treiber zu

verstehen. Nicht so wie bei Clover nach dem Motto: "Alles reinhauen und starten."

Ich habe auch zwischendurch mal den OCC getestet. Bekam beim Boot immer parse-error. Das deutet meistens darauf hin, dass irgendwelche Variablen im falschen Format vorliegen. Also den OCC nur zum Testen verwenden. Die config.plist lässt sich gut mit dem PLISTEditor aus dem App-Store bearbeiten. Zum Preis von 4,49€ kann man da nicht viel falsch machen. Wer die Datei mit einem normalen Text-Editor bearbeitet, der sollte dringend auf das Speicherformat achten. Ich habe mir damit einmal eine plist-Datei zerschossen.

Wenn man die Dokumentation mal gründlich durchliest, dann wird man feststellen, dass die Konfiguration letztendlich gar nicht so schwer ist. Viele default-Werte können übernommen werden. Vieles braucht man gar nicht ausfüllen. Mit der Zeit kennt man sich in den Tabellen aus und kann Wichtiges von Unwichtigem unterscheiden. Einziger Wermutstropfen ist, dass manche Werte im Format Little Endian eingegeben werden müssen. Einige Variablen, die im Format DATA vorliegen, verlieren ihren Schrecken, wenn man mal das Format von DATA auf STRING ändert.

Frage zum USB-Port-Limit: Die Anzahl der USB-Ports ist auf 15 Ports begrenzt. Sind es wirklich 15 physische bzw. logische Ports oder beginnt die Nummerierung bei USB-Port "0" und endet bei USB-Port "15"? Ich habe insgesamt 16 Ports, die sich wie folgt verteilen:

4x USB3.0 = 8 Ports und 8x USB2.0 = 8 Ports.

PLISTEditor aus dem App-Store

<https://apps.apple.com/de/app/plist-editor/id1157491961>

Allen noch einen schönen zweiten Weihnachtstag

Beitrag von „griven“ vom 26. Dezember 2019, 21:02

15 Ports pro Controller ist die Faustformel zum USB Port Limit und damit sind logische Ports gemeint und nicht physische 😊

Ein USB3 Port zählt zum Beispiel immer als 2 Ports denn er kann sowohl als USB3 (XHCI oder auch **SuperSpeed**) als auch als USB2 (EHCI oder **HighSpeed**) Port arbeiten und besitzt somit im Normalfall mindestens 2 Bezeichnungen (zum Beispiel HS01 und SS01) und zählt damit als 2 Ports. Das Limit greift wie gesagt pro Controller und hier haben wir die Crux denn Du wirst feststellen das Dein Mainboard mindestens 2 Controller besitzt zumeist einen XHCI Controller (USB3) und einen EHCI Controller (USB2). Der EHCI Controller kann nur USB2 bedienen hier sind also 15 physikalische Ports möglich wobei nicht jeder Port den der Controller zur Verfügung stellt auch auf dem Board ausgeführt ist zudem kann es sein das einzelne Ports des Controllers intern auf einen HUB laufen was aber nicht schlimm ist weil zur Controller Seite hin bleibt es ein Port. Der XHCI Controller kann sowohl USB2 als auch USB3 bedienen hier hängt die Anzahl der physischen Ports zum einen davon ab wie viele USB3 Ports auf dem Board tatsächlich verfügbar sind (inkl, der Header auf dem Mainboard) und wie viele dieser Ports im Mix Mode betrieben werden ([Bios Einstellungen](#)).

Wenn ich mich richtig erinnere hat das Z77-DS3H drei logische Controller (EHC0, EHC1 und XHC) zwei mal USB2 und einmal USB3 Du müsstest also eigentlich nur herausfinden welche der USB Ports

a) an welchem Controller hängen und

b) nach außen geführt sind wobei hier das Hackintool Dein Freund ist denn das hilft Dir die Ports zu ermitteln die wirklich auch physisch vorhanden sind und diejenigen auszuschließen die zwar logisch am Controller vorhanden sind aber auf dem Board nicht verdrahtet wurden.

Beitrag von „bluebyte“ vom 26. Dezember 2019, 21:19

[griven](#) ... vielen Dank für deine Erklärungen. Das mit den USB3.0-Ports kenne ich bereits. Deshalb schrieb ich ja auch 4x USB3.0 = 8 Ports. Ich finde es merkwürdig, dass Hackintool beim Einstecken der beiden Sticks (jeweils mit USB3.0 und USB2.0) nur die USB3.0-Ports kennzeichnet. Die Ports sind aber in Ordnung. Der Stick wird auf dem Desktop und im Finder angezeigt. Früher stand da immer HS und SS. Jetzt steht da PR und PRT. Na gut, da wo mein Drucker, mein WLAN-Stick und mein Logitech Unify drinstecken, die hießen schon früher PR.

Jetzt passt es wieder ...

mit SSDT **oder** mit USBMap.kext für USB-Ports

Nachtrag: 15 Ports pro Controller.

Beitrag von „karacho“ vom 27. Dezember 2019, 08:55

Moin [bluebyte](#) Hab deine EFI mal kurz überflogen. Da hast du aber ganz schön geackert, und auch alles gut dokumentiert in den Comments in der config. 👍

Du nutzt noch USBInjectAll, dann solltest/könntest du bei Kernel->Quirks das XhciPortLimit auf YES setzen. Bei UEFI->Protocols ConsoleControl auf YES und bei UEFI->Quirks noch RequestBootVarRouting auf YES. RequestBootVarRouting muss man nicht unbedingt haben, wenn du jedoch in den Systemeinstellungen gerne das Startvolume auswählen möchtest, dann brauchst du es. Auch hätte ich an deiner Stelle bei PlatformInfo die Schlüssel DataHub, PlatformNVRAM und SMBIOS gelöscht, so dass nur noch Generic übrig bleibt, wenn Automatic auf YES steht. Die Unterordner in ACPI hätte ich mir auch gespart. Ansonsten gute Arbeit wenn es deinen Rechner so bootet 👍

Beitrag von „mhaeuser“ vom 27. Dezember 2019, 09:19

XhXhciPortLimit *darf nicht* länger als nötig an sein

Beitrag von „karacho“ vom 27. Dezember 2019, 09:25

Ich vergass es zu erwähnen. Ja, nur solange bis er seinen eigenen USBPorts.kext gebastelt hat. Danke für die ergänzung [mhaeuser](#)

Beitrag von „bluebyte“ vom 27. Dezember 2019, 10:05

[karacho](#) und [mhaeuser](#) ... Vielen Dank, dass ihr euch mal die EFI angeschaut habt und für die Rückmeldungen und Hinweise. Werde sie dankend annehmen. Habe eben ausgemistet. Das mit dem RequestBootVarRouting ist natürlich genial. 😊 Die DSDT und SSDT in den ACPI sind wirklich überflüssig. Ich wollte es nur einmal testen. Musste mich mal wieder mit den Tools (USBmap, ssdtPRGen, MaciASL) beschäftigen. Ich möchte mich nicht nur daran erfreuen, dass der Rechner irgendwie läuft, sondern ich will es auch verstehen, warum manches mal so und mal so läuft. Kann es sein, dass mal abgesehen vom **Master-Kext LILU**, die Reihenfolge der Kexts eine entscheidende Rolle spielt? Die Kommentare in der config.plist stammen größtenteils von khronoskernel. Bei NVRAM sollte man sich jedoch mit Kommentaren mit "Comment" innerhalb von Dictionary zurückhalten, sonst werden die ins NVRAM geschrieben. 😊 Bei den Device Properties habe ich jetzt (Forrest Gump würde sagen: "Mal wieder!") die Layout-ID eingetragen. Jetzt klappt das auch mit der Soundkarte. Dachte bisher, dass das der PropertyInjector.kext automatisch erledigt. Falsch gedacht. Den Wert Platform-ID für die IGPU muss ich auch noch setzen damit die richtige IGPU angezeigt wird. Den Schalter "Normalize Headers" habe ich auf enabled, weil ich High Sierra starten möchte. Leider bekomme ich da immer noch Fehler.

Beitrag von „apfelnico“ vom 27. Dezember 2019, 10:35

Die Reihenfolge der Kexte ist elementar. Diese müssen nach Abhängigkeiten sortiert werden. Eine AppleALC beispielsweise wird nicht geladen, wenn nicht zuvor eine Lilu geladen ist. Insofern ist es gerade bei letztgenannten immens wichtig, diese so früh wie möglich zu laden, damit aufbauende Plugins auch ihre Basis finden.

Beitrag von „griven“ vom 27. Dezember 2019, 10:54

[apfelnico](#) und [mhaeuser](#) interessant in dem Zusammenhang wäre vielleicht auch zu wissen wie man herausfindet welche Abhängigkeiten es eigentlich gibt. Bei Extensions mit Plugins ist es ja irgendwie logisch, dass die Plugins Abhängigkeiten haben zum jeweils übergeordneten Kext (VoodooPS2, FakeSMC usw.) ebenso bei Lilu und dessen Erweiterungen, weil auch da ist es entsprechend dokumentiert, aber wie finde ich das bei Extensions heraus, wo es nicht so

eindeutig definiert ist?

Als Beispiel sei mal die BRCMPatchRam3 Geschichte genannt denn hier braucht es ja 3 Extensions neben der BRCMPatchRam3.kext auch den BrcmBluetoothInjector.kext und den BrcmFirmwareData.kext gut die Reihenfolge von BrcmFirmwareData.kext und BRCMPatchRam3.kext kann man sich anhand der Funktion noch logisch herleiten denn der PatchRam3 soll ja eine Firmware in den RAM der BT Karte schreiben daher scheint es logisch zu sein das der BrcmFirmwareData.kext vorher geladen sein sollte weil sonst wird es nichts mit dem Patch aber wie verhält es sich mit dem Injector? Logisch hergeleitet würde ich sagen der kommt ans Ende der Kette weil er als Injector dafür sorgt die Eigenschaften der BT Karte an das OS weiter zu reichen und das sollte er dann vermutlich erst tun wenn die Karte auch dazu bereit oder? Wie gesagt logisch hergeleitet aber gibt es einen Weg das gesichert zu erkennen?

Ein anderes Beispiel ist der VoodooInput.kext der für den VoodooPS2 Branch von Acidanthera benötigt wird auch hier stellt sich die Frage wie finde ich heraus in welcher Reihenfolge die Extensions geladen werden sollten? Voodooinput quasi als foundation vor allem was zu VoodooPS2 gehört oder erst VoodooPS2 dann VoodooInput und dann die Plugins? Wie gesagt wäre cool wenn es da eine Möglichkeit gibt die Abhängigkeiten gesichert zu ermitteln (gibt sicher eine nur kenne ich die noch nicht und damit stehe ich vermutlich nicht alleine da weil bisher musste man sich darum ja nicht wirklich kümmern) wenn Ihr da eine Idee habt oder gar eine Möglichkeit kennt wäre es prima das kurz zu erläutern. Danke schon mal 😊

Beitrag von „bluebyte“ vom 27. Dezember 2019, 15:07

[griven](#) ... bei meiner Soundkarte war es relativ einfach. Erst den RealtekALC.kext weil sich darin die info.plist befindet die vermutlich für den AppleALC.kext die Informationen liefert. AppleALC ohne RealtekALC läuft nicht. Falschmeldung. Es läuft doch ohne RealtekALC.kext.

Meine Kernel-Erweiterungen werden in dieser Reihenfolge geladen.

Item 0 Lilu.kext

Item 1 VirtualSMC.kext

Item 2 WhateverGreen.kext

Item 3 PropertyInjector.kext

Item 4 USBInjectAll.kext

Item 5 realtekALC.kext

Item 6 AppleALC.kext

Item 7 AtherosE2200Ethernet.kext

Item 8 SMCProcessor.kext

Item 9 SMCSuperIO.kext

Was mich immer noch wurmt, ist das das Umschalten der Auflösung kurz vor dem Login. Das war schon bei Clover so. Gibt es da eine elegante Lösung?

Beitrag von „karacho“ vom 27. Dezember 2019, 16:16

Ich denke, den PropertyInjector.kext kannst du dir sparen [bluebyte](#) . AppleALC.kext mit den korrekten DeviceProperties in deiner config (mit layout-id 1, 5 oder 7) sollte für Onboard Audio reichen. Oder brauchst du den PropertyInjector noch für was anderes? 🤔

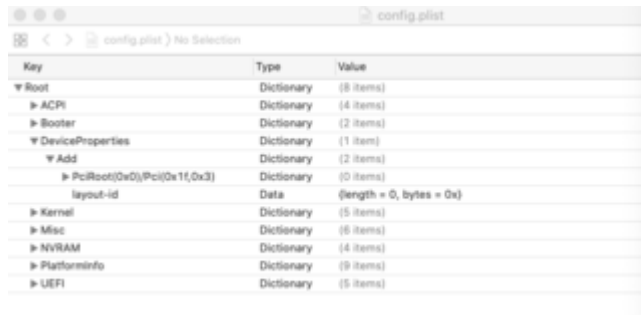
Mit Hackintool->PCI findest du den Device Path deines Audio Devices. Rechte Maustaste darauf und dann Copy Device Path und in deiner config mit Paste über einen bestehenden Eintrag den du nicht nutzt, einfügen und die layout-id eintragen. (siehe Beispiel anhand meiner config). PropertyInjector.kext noch auf NO und dann mal neu booten.

Wurzel	Dictionary	0 Schlüssel/Wert-Paare
ACPI	Dictionary	4 Schlüssel/Wert-Paare
Booter	Dictionary	2 Schlüssel/Wert-Paare
DeviceProperties	Dictionary	1 Schlüssel/Wert-Paare
Add	Dictionary	1 Schlüssel/Wert-Paare
PciRoot(0x0)/Pci(0x11,0x0)	Dictionary	1 Schlüssel/Wert-Paare
layout-id	Daten	4 Bytes: 07000000
Kernel	Dictionary	5 Schlüssel/Wert-Paare
Misc	Dictionary	6 Schlüssel/Wert-Paare
NVRAM	Dictionary	4 Schlüssel/Wert-Paare
PlatformInfo	Dictionary	6 Schlüssel/Wert-Paare
UEFI	Dictionary	5 Schlüssel/Wert-Paare

Beitrag von „StevePeter“ vom 27. Dezember 2019, 17:01

[karacho](#) könntest du mir mal bitte behilflich sein? entweder bin ich zu blind oder..

Wie muss ich für Sound layout-id 1 das eintragen? Ich habe nur dieses Bild im Xcode.



Key	Type	Value
Root	Dictionary	(8 items)
ACPI	Dictionary	(4 items)
Booter	Dictionary	(2 items)
DeviceProperties	Dictionary	(1 item)
Add	Dictionary	(2 items)
PciRoot(0x0)/Pci(0x1f,0x3)	Dictionary	(0 items)
layout-id	Data	(length = 0, bytes = 0x)
Kernel	Dictionary	(5 items)
Misc	Dictionary	(6 items)
NVRAM	Dictionary	(4 items)
PlatformInfo	Dictionary	(9 items)
UEFI	Dictionary	(5 items)

Beitrag von „bluebyte“ vom 27. Dezember 2019, 17:05

[StevePeter](#) ... genau so wie auf dem Screenshot. Der Wert muss im Format Little Endian eingegeben werden. Oh mein Gott, wenn ich noch an die 90er denke. Atari ST, Motorola, 68000er, Big Endian. Eigentlich kein Hexenwerk.

X 00 10 20 30

= 30 20 10 00

X 00 00 00 02

= 02 00 00 00

Beitrag von „karacho“ vom 27. Dezember 2019, 17:12

Läuft dein Audio jetzt mit Layout-id 2 [bluebyte](#) ?

[StevePeter](#) trage es so ein wie bluebyte oben schrieb.

Beitrag von „bluebyte“ vom 27. Dezember 2019, 17:14

[karacho](#) ... ja läuft mit Layout-ID 2

Beitrag von „karacho“ vom 27. Dezember 2019, 17:15

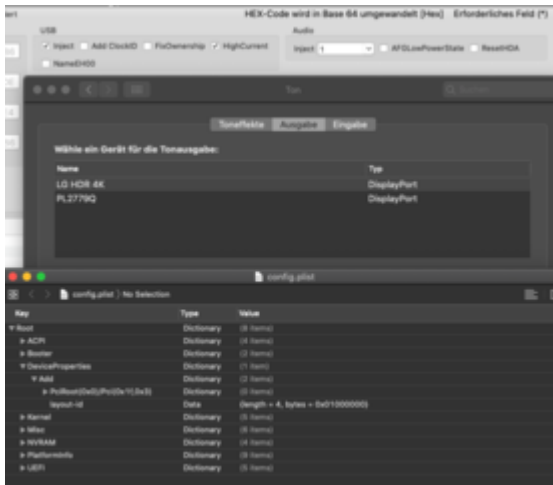
Einwandfrei. Und ohne propertyinjector?

Beitrag von „bluebyte“ vom 27. Dezember 2019, 17:28

[karacho](#) ... ohne PropertyInjector.kext

Beitrag von „StevePeter“ vom 27. Dezember 2019, 17:29

[bluebyte](#) hats drin, aber in den Systemeinstellungen/Sound werden nur die DisplayPorts angezeigt.?? Unter Clover habe ich aber mit Audio Inject 1 auch die Onboards.



Beitrag von „apfelnico“ vom 27. Dezember 2019, 17:33

Kexte alle vorhanden und in der richtigen Reihenfolge geladen?

Audio-ID geht auch einfach per Bootargument.

Beitrag von „bluebyte“ vom 27. Dezember 2019, 18:01

[StevePeter](#) ... gehst Du über Monitorausgang? Was sagt Hackintool? Hast Du ioregexplorer?




Bin jetzt selbst am Grübeln. Jetzt tauchen im ioregexplorer Begriffe auf wie layout-id **und** alc-layout-id.

Habe mich um die Onboard nie so richtig gekümmert. Gehe meistens über mein Focusrite Scarlett, wenn ich meine Gitarren anschließe und in die Saiten hauer. Hatte die sogar eine Zeit lang im BIOS ausgeschaltet.

Beitrag von „karacho“ vom 27. Dezember 2019, 18:18

[StevePeter](#) dann probiere mal den Vorschlag von apfelnico. Lösche den Key unter DeviceProperties und trage dafür bei Nvram bei den boot-args alcid=1 mit ein. Und achte darauf, dass Lilu in deiner config an erster Stelle steht.

Beitrag von „StevePeter“ vom 27. Dezember 2019, 18:40

[apfelnico](#) [karacho](#)  Ihr seid echt super. So funktioniert es mit den int. Sound   Vielen Dank

PS: eine Kleinigkeit hätte ich noch: Wie kann ich die Boot-Reihenfolge ändern? Bei mir steht auf 1 Windows und auf 2 macOS.

Gestartet wird aber immer Windows, ich möchte aber macOS gestartet haben. Das finde ich einfach nicht.