

OpenCore Bootloader

Beitrag von „Noir0SX“ vom 13. April 2019, 09:48

[vit9696](#), [mhaeuser](#) und andere, zu lesen unter der Githubseite unter Credits und den vielen Commits, haben mit <https://github.com/acidanthera/OpenCorePkg> eine weitere Möglichkeit für einen Bootloader geschaffen.

OpenCore ist zur Zeit noch viel Handarbeit, das geht solange es kein Release gibt schon los mit dem erstellen der Dateien.

Alles um OpenCore zum Booten zu bewegen spielt sich in der plist ab. Ein Beispiel dieser config Datei und eine PDF mit den Erklärungen und Aufbau dieser und der Verzeichnisstruktur findet man unter Docs <https://github.com/acidanthera/OpenCorePkg/tree/master/Docs>

Am bestehen alles auf Github downloaden, da hat man es immer local und muss nicht irgendwelche Copy & Pasta Geschichten machen.

Da sich zur Zeit viel tut, hier mal der aktuelle Stand hier [OpenCore Bootloader](#)

Статус на апрель 2019: übernommen von applelife.ru

- Поддерживается как UEFI загрузка, так и DuetPkg (legacy)
- Запуск с APFS и HFS+
- ACPI патчер (добавление, удаление, бинарные патчи, релокейт)
- Apple-совместимая реализация bless
- Инжект DeviceProperties
- Генерация DataHub и SMBIOS
- Символьный патчер на ядро и кексты
- Инжект/патч/блокировка кекстов в prelinkedkernel
- Установка/recovery/FV2
- Конфигурирование через config.plist с открытой документацией
- Простейший boot picker для запуска
- Прямой запуск с dmg образа

Beitrag von „anonymous_writer“ vom 13. April 2019, 10:19

Hier auch gleich mein Arbeitstand. Leider wird noch nicht alles geladen wie es soll, aber der Laptop, das Zenbook, startet damit. SMBios Nummern habe ich verändert, da die fest in der config.plist eingebunden sind.

Folgende Kexte möchten nicht richtig laden:

VoodooI2C.kext

VoodooI2CHID.kext

BrcmFirmwareData.kext

BrcmPatchRAM2.kext

AsusSMC.kext

Aktuell für mich nicht verständliche Fehlermeldungen:

"No Arguments are initialized for method [FCPU]"

"LowerOnePowerStateTo" zur BT Schnittstelle.

Beitrag von „CMMChris“ vom 13. April 2019, 10:25

Hm, warum denn noch ein Bootloader? Die Arbeitskraft hätte man auch in die Verbesserung und Weiterentwicklung von Clover stecken können.

Beitrag von „Plonker“ vom 13. April 2019, 10:27

Weil Clover nicht auf Open Source basiert.

Beitrag von „Noir0SX“ vom 13. April 2019, 10:59

Um die Dateien auch ohne Vorkenntnisse zu erstellen hat [vit9696](#) eben eine Möglichkeit geschaffen <https://github.com/acidanthera...9d3f5a73805dc1fa1bf0d5011>

Daten von Github downloaden, [macbuild.tool](#) ins Terminal ziehen. Wird wer nicht immer damit umgeht erstmal eine Weile dauern, weil da benötigte Dateien geladen werden.

Das Ergebnis befindet sich dann im Ordner vom Download unter **UDK - Build - OpenCorePkg - RELEASE_XCODE5** (oder die Version die ihr wollt) - **x64**

Beitrag von „derHackfan“ vom 13. April 2019, 11:10

Clover, Ozmosis, OpenCore, auf drei Beinen steht man besser. 😊

Beitrag von „Mocca55“ vom 13. April 2019, 11:24

Wichtig zu erwähnen wäre noch das es sich ausschließlich um einen **UEFI Bootloader** handelt -> Legacy Boards mit UEFI support funktioniert damit nicht!

Beispiel hierfür ist das Board MS7728V2 das in meinem Microstar (Medien PC) verbaut ist...

Gruß Mocca55

Beitrag von „rubenszy“ vom 13. April 2019, 12:05

Reden wir mal lieber auf 2 Beine OpenCore und Ozmosis haben schon sehr viel Ähnlichkeit,

endlich mal die verbesserte Version für die in Jahre gekommene Ozmosis.

Das gute ist Open Source.

Bei Clover musst du Anmerkungen machen, ob die dann in berücksichtigt werden ist dahin gestellt, die machen eh was sie wollen, sieht man ja mit bescheuerten Sound wenn Clover jetzt lädt oder noch besser war das Vektoren Theme völlig sinnlos.

Am besten finde ich das ganze zur Weihnachtszeit, ob du willst oder nicht nach einem Clover Update ist immer das beschissene Weihnachtstheme im NVRAM hinterlegt, wenn man auf so ein Theme Bock hat, dann fügt man es selber ein und wird nicht dazu gezwungen den scheiß zu akzeptieren.

Ich hoffe mal nur so welche Faxen wird bei OpenCore nicht gemacht und wenn doch es ist ja Open Source 😊

Beitrag von „mhaeuser“ vom 13. April 2019, 13:51

[Zitat von CMMChris](#)

Hm, warum denn noch ein Bootloader? Die Arbeitskraft hätte man auch in die Verbesserung und Weiterentwicklung von Clover stecken können.

Ich empfehle einen Blick in den Clover-Quellcode. 😊

[Zitat von derHackfan](#)

Clover, Ozmosis, OpenCore, auf drei Beinen steht man besser. 😊

OpenCore wurde von The HermitCrabs Lab als Nachfolger von Ozmosis angefangen und wird jetzt Open Source hauptsächlich von vit fertiggestellt - ich weiß nicht, wie "angenehm" die Möglichkeit zum Laden vom FW-Chip sein wird, insofern wir dazu kommen, aber es ist zumindest vermerkt

[Zitat von rubenszy](#)

Ich hoffe mal nur so welche Faxen wird bei OpenCore nicht gemacht

Ist nicht in Planung 😄

[Zitat von anonymous writer](#)

Folgende Kexte möchten nicht richtig laden:

VoodooI2C.kext

VoodooI2CHID.kext

BrcmFirmwareData.kext

BrcmPatchRAM2.kext

AsusSMC.kext

Alles anzeigen

[anonymous writer](#) Die Ladereihenfolge entspricht der Deklaration in der PLIST, zum Beispiel müsste VoodooI2CServices vor VoodooI2C deklariert werden.

Beitrag von „rubenszy“ vom 13. April 2019, 14:28

Wenigsten wird es jetzt doch noch was mit OpenCore, paar Jahre hat es gedauert, wie sagt man so schön, was lange wert wird gut.

Schauen wir mal was daraus wird.

Beitrag von „anonymous_writer“ vom 13. April 2019, 15:15

Hallo [mhaeuser](#) ,

das war das Problem. In der Reihenfolge

1. VoodooI2CServices.kext
2. VoodooGPIO.kext
3. VoodooI2C.kext
4. VoodooI2CHID.kext

läuft das Trackpad wie eine Eins.

Danke für deine Info. Ich werde mir das mal für alle Kexte ansehen.

Beitrag von „vit9696“ vom 13. April 2019, 16:12

Zitat von Mocca55

Wichtig zu erwähnen wäre noch das es sich ausschließlich um einen **UEFI Bootloader** handelt -> Legacy Boards mit UEFI support funktioniert damit nicht!

Beispiel hierfür ist das Board MS7728V2 das in meinem Microstar (Medien PC) verbaut ist...

Gruß Mocca55

BIOS booting works with DuetPkg. For now you could grab a compatible Duet on applelife:

<https://applelife.ru/posts/800201/>

Beitrag von „TahsinAhmed“ vom 13. April 2019, 16:21

[Zitat von Plonker](#)

Weil Clover nicht auf Open Source basiert.

Ich dachte immer Clover wäre auch Open Source?

Auf der Sourceforge steht auch das es eine [BSD License](#) hat, was auch eine Open Source [License](#) sein sollte.

Aber ich kenne mich damit nicht wirklich aus.

Beitrag von „Mocca55“ vom 13. April 2019, 16:21

Thanks for the Information [vit9696](#)

I will Test it with Duet.

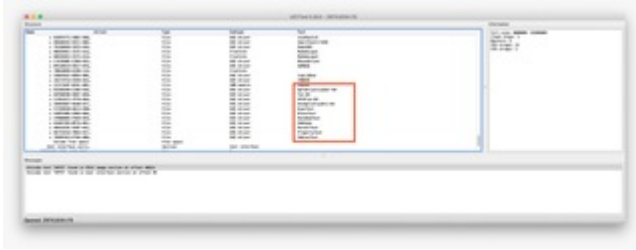
Greetings Mocca55

Beitrag von „derHackfan“ vom 13. April 2019, 16:44

[Zitat von Download-Fritz](#)

ich weiß nicht, wie "angenehm" die Möglichkeit zum Laden vom FW-Chip sein wird

Habe einfach aus OcSupportPKG die Einzelteile entnommen und in ein Rom eingebaut.



Start dann über die Shell -> Boot.efi oder über die CLOVERX64.efi -> Auswahlmenu und läuft soweit auch ganz gut.

Beitrag von „mhaeuser“ vom 13. April 2019, 17:06

Jesus Maria, alles mit "Test" im Namen ist Testcode, mit dem die Codeschnipsel debuggt wurden und GdbSyms ist ein Helper für den GNU-Debugger. OC selbst ist nicht mal drin, oder? Würde meines Wissens auch aktuell noch nicht funktionieren, zumindest nicht ohne Bootstrap.

Beitrag von „derHackfan“ vom 13. April 2019, 18:22

Ja noch ohne Oc,



es war einfach ein Versuch mit dem Rom weil es mich so gereizt hat, ich kann damit macOS Sierra, HS und Mojave booten.

Beitrag von „anonymous_writer“ vom 13. April 2019, 19:16

Hello [vit9696](#) ,

I tried to use the file boot_XHCI from the package DuetPkg. Unfortunately, I did not understand where the file must be placed in the OC EFI folder.

Can you give little bit more help?

Beitrag von „roddy20“ vom 13. April 2019, 21:03

installing boot sectors sample

Code

1. boot0af=~/.src/UDK2018/Clover/CloverPackage/sym/CloverCD/usr/standalone/i386/boot0af;
2. boot1f32=~/.src/UDK2018/Clover/CloverPackage/sym/CloverCD/usr/standalone/i386/boot1f32;
3. diskutil umount \${bootdisk}s1;
4. sudo dd if=\${bootdisk} count=1 of=origMBR;
5. cp -fv origMBR newMBR;
6. dd if=\${boot0af} of=newMBR bs=1 count=440 conv=notrunc;
7. sudo dd if=newMBR of=\${bootrdisk} count=1;
8. sudo fdisk -f newMBR -u -y \${bootrdisk};
9. cp -fv \${boot1f32} newBS;
10. dd if=origBS of=newBS skip=3 seek=3 bs=1 count=87 conv=notrunc;
11. sudo dd if=newBS of=\${bootrdisk}s1 count=1;
12. diskutil mount \${bootdisk}s1;

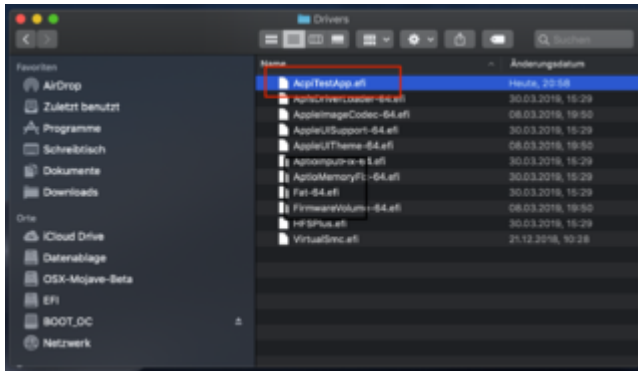
Alles anzeigen

Code

1. bootdisk is /dev/diskN
2. bootrdisk is /dev/rdiskN
3. boot0af and boot1f32 from Clover installer or Clover boot cd

Beitrag von „anonymous_writer“ vom 13. April 2019, 21:12

Jetzt geht BT 😊



Beitrag von „kuckkuck“ vom 13. April 2019, 23:21

[vit9696](#) [mhaeuser](#) Thanks for your great work!

I appreciate that the concepts and ideas behind Ozmosis didn't land in the trash, been following all the commits on Github and from every code-snippet I can retrace I am impressed - unfortunately something new in the hackintosh-world... Thanks for introducing the community to symbol-based patches, comprehensible logic, compact code and security!

Beitrag von „anonymous_writer“ vom 14. April 2019, 00:22

Here another working EFI for my Zenbook.

Now all next are loading.

Unsolved problems:

- > After shutdown reboot
- > Bluetooth only works with AcpiTestApp.efi

```

/dev/disk1s2 on / (apfs, local, journaled)
RCP1 Error: (\_SB\_RTXD.QM0D) Namespace lookup failure, RC_NOT_FOUND (20
[FCPU] 000126 #0020:

Initialized Local Variables for method [FCPU]:
Local0: ffffffff8037cbfa10 <Obj> Integer 0000000000000032
Local1: ffffffff8037cbfa00 <Obj> Integer 0000000000000001
Local2: ffffffff8037cbfa10 <Obj> Integer 0000000000000012
Local3: ffffffff8037cbfa20 <Obj> Integer 0000000000000002
Local4: ffffffff8037cbfa20 <Obj> Integer 0000000000000005
Local5: ffffffff8037cbfa00 <Obj> Integer 0000000000000001

No Arguments are initialized for method [FCPU]

```

```

VirtualSMC: vmm: # postInterrupt reserve overflow
VirtualSMC: vmm: # postInterrupt reserve overflow
VirtualSMC: vmm: # postInterrupt reserve overflow
VirtualSMC: vmm: # postInterrupt reserve overflow
VirtualSMC: vmm: # postInterrupt reserve overflow
VirtualSMC: vmm: # postInterrupt reserve overflow
VirtualSMC: vmm: # postInterrupt reserve overflow
VirtualSMC: vmm: # postInterrupt reserve overflow
VirtualSMC: vmm: # postInterrupt reserve overflow
VirtualSMC: vmm: # postInterrupt reserve overflow
VirtualSMC: vmm: # postInterrupt reserve overflow
VirtualSMC: vmm: # postInterrupt reserve overflow
VirtualSMC: vmm: # postInterrupt reserve overflow
VirtualSMC: vmm: # postInterrupt reserve overflow
VirtualSMC: vmm: # postInterrupt reserve overflow

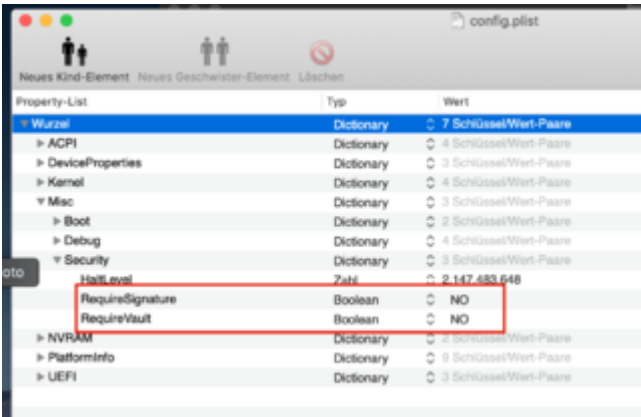
```

Beitrag von „Noir0SX“ vom 14. April 2019, 09:23

Wer es selber nicht gebaut bekommt, hier mal der aktuelle Stand von Github

Beitrag von „anonymous_writer“ vom 14. April 2019, 12:39

Bei der neuen Version von [Noir0SX](#) darauf achten das diese beiden Parameter auf NO gesetzt werden. Ansonsten muss man sich immer erst die entsprechenden Dateien nach der Anleitung erstellen.



Description: Requires `va1111.plist` file present in OC directory.

This file should contain SHA-256 hashes for all files used by OpenCore. Presence of this file is highly recommended to ensure that unintentional file modifications (including filesystem corruption) do not happen unnoticed. To create this file automatically use `create_va1111.sh` script.

Regardless of the underlying filesystem, path name and case must match between `config.plist` and `va1111.plist`.

Note: `va1111.plist` is tried to be read regardless of the value of this option, but setting it to `true` will ensure configuration sanity, and abort the boot process.

The complete set of commands to:

- Create `va1111.plist`.
- Create a new RSA key (always do this to avoid loading old configuration).
- Embed RSA key into `OpenCore.efi`.
- Create `va1111.sig`.

Can look as follows:

```
cd /Volumes/EFI/EFI/OC
./create_va1111.sh
./libTool -sign va1111.plist va1111.sig va1111.pub
efi=$(find $(dirname $0) -type f -name OpenCore.efi | grep '^=BEGIN OC VA1111=' | cut -f1 -d' ')+16}}
dd if=OpenCore.efi if=va1111.pub bs=1 seek=$((efi count=520 conv=notrunc
rm va1111.pub
```

Note: While it may appear obvious, but you have to use an external method to verify `OpenCore.efi` and `BOOTx64.efi` for secure boot path. For this you are recommended to at least enable UEFI SecureBoot with a custom certificate, and sign `OpenCore.efi` and `BOOTx64.efi` with your custom key. More details on customising secure boot on modern firmwares can be found in [Turning UEFI SecureBoot paper \(in Russian\)](#).

Beitrag von „Noir0SX“ vom 14. April 2019, 12:58

Ist halt zur Zeit Standart <https://github.com/acidanthera...94df06ca00e121ee323fad207>

Beitrag von „anonymous_writer“ vom 14. April 2019, 13:22

Besser wäre aber zum Testen wenn es nicht Standart wäre. Mit NO wird direkt aus den Ordner geladen wie bisher.

Beitrag von „Noir0SX“ vom 17. April 2019, 05:41

Hier mal wieder der letzte Stand 17.04.2019

Gibt es weiter unten

Beitrag von „anonymous_writer“ vom 18. April 2019, 21:10

Sagt jemanden die angehängte Fehlermeldung was? Die kommt bei mir beim Herunterfahren des Laptops.

Beitrag von „derHackfan“ vom 19. April 2019, 10:06

Es gibt ein Update... Stand 19.04.2019

Beitrag von „Noir0SX“ vom 19. April 2019, 10:26

[anonymous writer](#)

ist ja oft so das die Laptops nochmal eine grössere Herausforderung sind...

daher kann ich ich deinen Fehler auf dem Laptop bestätigen, der auf den anderen PCs so nicht war

Code

1. Kernel Extensions in backtrace:
2. com.apple.iokit.IOHIDFamily(2.0)[B22CED3C-FF8E-3F4C-94DE-52E156703859]@0xfffff7f9376c000->0xfffff7f937e9fff

Habe es auch mal mit der letzten I2C versucht, bringt aber nix in dieser Beziehung zum Fehler

Beitrag von „anonymous_writer“ vom 19. April 2019, 12:46

Zumindest ist es bei uns gleich. Auf dem Desktop hatte ich denn Fehler auch nicht.



Beitrag von „Noir0SX“ vom 19. April 2019, 16:29

Alle auf aktuellen Stand

Beitrag von „mhaeuser“ vom 19. April 2019, 16:38

An der Stelle mal die Erinnerung, dass auch halbgarer und unfertiger Code auf den Master-Branch gepusht wird... alles, was nicht als Release gekennzeichnet ist, ist nicht "offiziell"

Beitrag von „griven“ vom 19. April 2019, 22:13

[mhaeuser](#) das sollte eigentlich allen die an Board sind bewusst sein das es Work in Progress ist und es eben keine Garantie gibt das immer alles funktioniert wie erwartet. Ich finde es dennoch unglaublich spannend zu beobachten was da entsteht und möchte an der Stelle auch mal meinen Dank an Dich und [vit9696](#) und natürlich auch an alle anderen beteiligten zum Ausdruck bringen. Das was Ihr da macht ist ganz großes Kino 🙌

Beitrag von „derHackfan“ vom 19. April 2019, 22:38

Es gibt ein Update... Stand 19.04.2019

- OpenCoreMisc: Fix log prints
-

Beitrag von „derHackfan“ vom 20. April 2019, 12:59

Es gibt ein Update... Stand 20.04.2019

- OpenCoreAcpi: Put ACPI -> Add to a dictionary
 - **OpenCore: Do not use trailing slash for root directory open**
 - **OpenCore: Remove accidentally committed testing code**
-

Beitrag von „derHackfan“ vom 21. April 2019, 18:18

Es gibt ein Update... Stand 21.04.2019

Beitrag von „Noir0SX“ vom 22. April 2019, 00:00

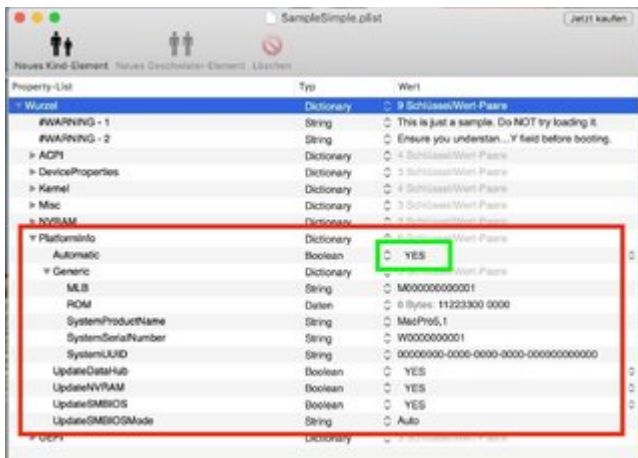
Mit <https://github.com/acidanthera/MacInfoPkg> wird eine Unterstützung für PlatformInfo hinzugefügt, mit der man einen neuen Abschnitt schreiben kann, dabei kann DataHub, PlatformNVRAM und SMBIOS ignoriert werden, wenn Automatic = YES steht. In diesem Fall können diese drei Abschnitte vollständig gelöscht werden.

Noch ist die Datenbank etwas leer (MacPro5,1 und MacBookPro15,2).

Beitrag von „anonymous_writer“ vom 22. April 2019, 00:29

Und hier noch die dazu passende OpenCore.efi.

Bis auf das Herunterfahren welches zu einen Kernelpanik führt funktioniert alles.



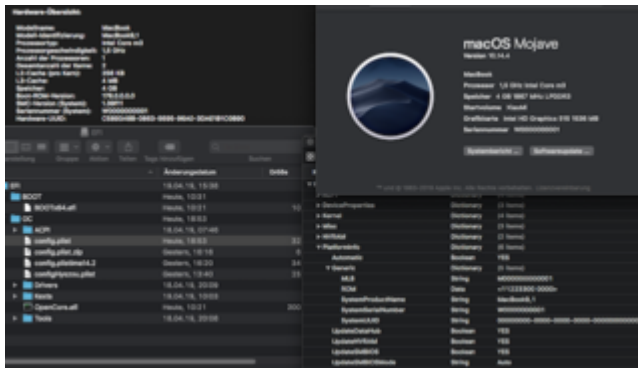
Beitrag von „derHackfan“ vom 22. April 2019, 18:11

Sehr gut! 😊



Danke! 👍

Beitrag von „Noir0SX“ vom 22. April 2019, 19:06



... ..

Beitrag von „cecekpawon“ vom 23. April 2019, 08:04

[Zitat von NoirOSX](#)

Alle auf aktuellen Stand

Thanks for sharing the patched shell _^_ Kudos to all OxC team

Beitrag von „Noir0SX“ vom 25. April 2019, 16:07

Da kann das Testen ja weiter gehen, ohne das ich die vielen USB-Sticks durcheinander bringe





Beitrag von „kuckkuck“ vom 28. April 2019, 20:23

Hat jemand bereits die default boot-options Funktionalität getestet?

Beitrag von „mhaeuser“ vom 28. April 2019, 23:22

Die ist aktuell noch sehr eingeschränkt und wird bei den meisten Setups nicht funktionieren. Vielleicht fix' ich morgen den Bug, der verhindert, dass der OC-Shim in der Bootreihenfolge übersprungen wird, dann sollte es funktionieren, insofern die FW nicht die entsprechende Bootoption löscht. Nutzt man den Shim nicht oder ist er nicht der Standard, sollte es schon funktionieren, zumindest für macOS

Beitrag von „kuckkuck“ vom 29. April 2019, 12:12

Zumindest im Log sollte sich jedoch bereits die Funktionsweise erkennen lassen, oder? Sprich, die richtige default Boot Option/bootnext wird bereits erkannt...

Beitrag von „NoirOSX“ vom 2. Mai 2019, 15:54

[Zitat von anonymous writer](#)

Sagt jemanden die angehängte Fehlermeldung was? Die kommt bei mir beim Herunterfahren des Laptops.

Das kommt eindeutig von I2C ohne diese Kexte kommt die KP samt Verhalten nicht und es ist wie am Desktop. Habe gerade mal ohne probiert.

Beitrag von „NoirOSX“ vom 2. Mai 2019, 16:22

Danke an [Ukr55](#) von applelife sind nun mehr SMBios Daten enthalten. Habe es mal damit neu gebaut.

Zum Testen config.plist, Driver etc von Github Acidanthera nehmen, da ist viel neu oder geändert.

Dateien aktuell hier [OpenCore UEFI Bootloader](#)

Beitrag von „anonymous_writer“ vom 2. Mai 2019, 16:53

Danke [NoirOSX](#) , werde ich Morgenabend testen.

Beitrag von „derHackfan“ vom 2. Mai 2019, 21:36

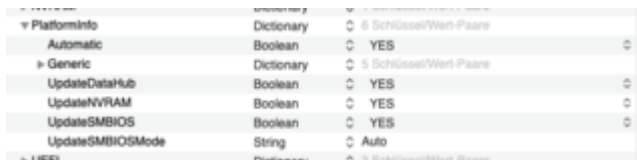
[Zitat von NoirOSX](#)

Zum Testen config.plist, Driver etc von Github Acidanthera nehmen, da ist viel neu oder geändert.

Funktioniert bei mir so nicht.

Habe mal eben das macbuild.tool machen lassen und komme auf ein anderes Ergebnis.

In meinem Fall habe ich BOOTx64.efi und OpenCore.efi (203 KB statt 221 KB) einfach ersetzt, in der config.plist unter Plattforminfo musste ich nichts ändern, danach erfolgt der Start wie gewohnt auf den Schreibtisch.



Key	Type	Value
PlatformInfo	Dictionary	5 Schlüssel/Wert-Paare
Automatic	Boolean	YES
Generic	Dictionary	5 Schlüssel/Wert-Paare
UpdateDataHub	Boolean	YES
UpdateNVRAM	Boolean	YES
UpdateSMBIOS	Boolean	YES
UpdateSMBIOSMode	String	Auto

Hier sollte m.M.n jeder OC Anwender sehr behutsam vorgehen, viel Dokumentation bedeutet nicht gleich Garant auf Erfolg, es mangelt derzeit an Anleitung und oder Unterstützung wann welcher Eintrag in der config angepasst werden muss.

Edit: Generic Daten wurden entfernt, bitte mit Clover Configurator auffüllen.

Gruß

Beitrag von „losinka“ vom 3. Mai 2019, 00:59

+ Generic Auto auf iMac18,3

Danke Vit!

Beitrag von „apfelnico“ vom 3. Mai 2019, 03:16

Anbei mein EFI für mein Asus Prime X299-Deluxe. Habe nur im "Generic" die MLB/ROM/SystemSerialNumber/SystemUUID entfernt, läuft als iMac Pro. Ansonsten alles wie mit Clover. Derzeit noch im Verbose-Mode, aber mit nativer Monitorauflösung von meinem alten "Apple Cinema HD Display" (16:10), 1920x1200. Also kein grober Text mehr wie in der ersten Variante.

Beitrag von „anonymous_writer“ vom 3. Mai 2019, 19:02

Hallo [Noir0SX](#) ,

du hast wie immer Recht, ohne den I2C Kext fährt das Zenbook sauber runter.

Somit kann ich berichten, das Zenbook läuft mit OC ohne irgendein Problem wenn man den I2C Kext nicht nutzt.



Beitrag von „Noir0SX“ vom 3. Mai 2019, 19:40

Nun müssen wir den irgendwie biegen, will mein Touchpad wieder [anonymous_writer](#) 😄

Beitrag von „locojens“ vom 4. Mai 2019, 07:45

Moin! Ich habe es noch immer nicht geschafft meine Vega zur mitarbeit zu bewegen. Bin das Brett vorm Kopf noch nicht los geworden.

Beitrag von „Noir0SX“ vom 4. Mai 2019, 09:15

Bei mir läuft nun **OpenCore v0.2** , im Anhang mal die benötigten bzw die aktuellen Daten zum testen

Dazu noch der **OpenCore Changelog** vom **Release**
<https://github.com/acidanthera/OpenCorePkg/releases> der Version

v0.0.1

Initial developer preview release

WARNING: This release is a developer preview, i.e. it targets experienced engineers and developers. Configuration and feature set are under works, i.e. refinements are expected to happen with 1.0.0 release.

Beitrag von „Noir0SX“ vom 4. Mai 2019, 12:20

Für die schnelle zum bearbeiten der Plist, einfach aus dem Terminal gestartet.

Cross platform GUI plist editor written in python <https://github.com/corpnewt/ProperTree> von CorpNewt

Key	Type	Value
#WARNING - 1	0 String	This is just a sample. Do NOT try loa
#WARNING - 2	0 String	Ensure you understand EVERY field
ACPI	0 Dictionary	4 key/value pairs
Add	0 Array	4 children
0	0 Dictionary	3 key/value pairs
Comment	0 String	DSDT
Enabled	0 Boolean	True
Path	0 String	DSDT.aml
1	0 Dictionary	3 key/value pairs
Comment	0 String	Fix 123
Enabled	0 Boolean	True
Path	0 String	SSDT.aml
2	0 Dictionary	3 key/value pairs
Comment	0 String	Fix 456
Enabled	0 Boolean	False
Path	0 String	SSDT-1.aml
3	0 Dictionary	3 key/value pairs
Comment	0 String	
Enabled	0 Boolean	False
Path	0 String	SSDT-ACPI.aml
Block	0 Array	3 children
0	0 Dictionary	6 key/value pairs
All	0 Boolean	False
Comment	0 String	Drop DMAR
Enabled	0 Boolean	False

Beitrag von „derHackfan“ vom 4. Mai 2019, 19:30

Hier läuft jetzt auch OpenCore v0.2 wobei ich nur OpenCore.efi ersetzt habe, der Unterschied betrug bei mir nur 2 KB, der Rest (ist) von der ESP/EFI ist damit gleich geblieben.

Beitrag von „apfelnico“ vom 4. Mai 2019, 19:45

Wobei auch immer interessant ist, wie sich die config.plist ändert, also welche Funktionen dazu kommen.

Beitrag von „locojens“ vom 5. Mai 2019, 15:22

Wie steht es denn um die NVRam-Geschichte bei Clover gab es ja den EMUVariable.....64.efi glaube der wird es bei OC wohl eher nicht bringen nehme ich mal ganz stark an. Siehe Neustarten Herunterfahren ohne Reset-Taste.

Beitrag von „Sascha_77“ vom 12. Mai 2019, 12:09

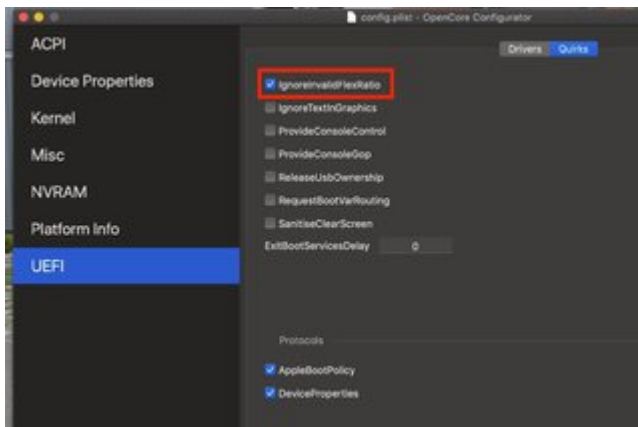
Nachdem mein T530 reibungslos auf OC umzustellen war habe ich mich an mein T440 begeben. Das gibt sich leider ein wenig zickiger. Hab mal die EFI mit angehängen und nen Video hochgeladen. Vllt. hat jemand ja eine Idee woran es liegt, dass er nicht booten mag.

<https://youtu.be/TnmUhkIUPBs>

Beitrag von „anonymous_writer“ vom 12. Mai 2019, 15:40

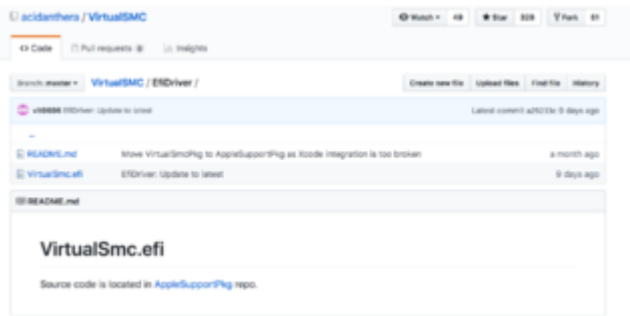
[Sascha 77](#),

versuche mal diesen Haken.

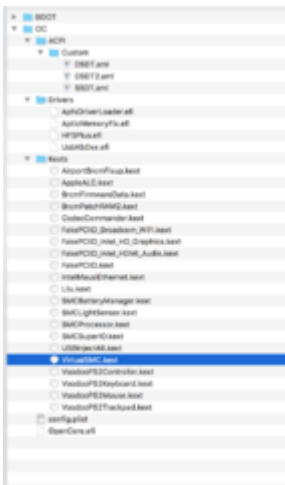


Beitrag von „rubenszy“ vom 12. Mai 2019, 16:33

Ohne das



wird der



nicht geladen.

Beitrag von „apfelnico“ vom 12. Mai 2019, 17:15

Nein, ist nicht zwingend.

Beitrag von „Sascha_77“ vom 12. Mai 2019, 19:27

[anonymous_writer](#)

Der Haken ist bereits gesetzt.

[Zitat von apfelnico](#)

Nein, ist nicht zwingend.

Richtig. Auf meinem T530 gehts auch ohne. Habe es jetzt aber mal trotzdem probiert. Geht nicht.

Beitrag von „anonymous_writer“ vom 12. Mai 2019, 19:49

Hast du die anderen UEFI Einstellungen auch getestet?

Es gibt dazu auch eine Anleitung im OC Build.

Hier mein EFI-Ordner vom Zenbook. Läuft alles außer dem

<https://github.com/alexandred/VoodooI2C>

<https://bitbucket.org/anonymou...uak-oc-folder/src/master/>

Beitrag von „derHackfan“ vom 12. Mai 2019, 22:36

Ein Problem weniger, der SMBus Controller läuft jetzt dank der SSDT-SBUS-MCHC.aml, fehlt nur noch einer... 😊



Beitrag von „apfelnico“ vom 12. Mai 2019, 23:19

In der config.plist gibt es unter PlatformInfo bei den SMBIOS Properties den "ProcessorType".

Bei mir folgend gesetzt:

Code

1. `<key>ProcessorType</key>`
2. `<integer>3841</integer>`

Entspricht Hex 0x0F01

tata, schon wird aus "unbekannt":



Nur Kosmetik, aber schön ist's. 😊

Beitrag von „muhviehstarr“ vom 14. Mai 2019, 00:59

[Sascha 77](#) probier mal

```
<key>AppleXcpmCfgLock</key>
```

```
<true/>
```

Beitrag von „Roy Jones“ vom 14. Mai 2019, 18:57

[Zitat von derHackfan](#)

Ein Problem weniger, der SMBus Controller läuft jetzt dank der SSDT-SBUS-MCHC.aml, fehlt nur noch einer... 😊



[derHackfan](#) : kannst du mir mal einen Tipp geben wie ich den Patch bei meinem z97 d3h einbinde so dass der SMBus Controller bei mir läuft ? Und seit dem ich OpenCore nutze zeigt wird mir bei meiner Grafikkarte der falsche Steckplatz angezeigt.

Beitrag von „Noir0SX“ vom 14. Mai 2019, 19:00

Die SSDT liegt in der Docs in OC

Beitrag von „Roy Jones“ vom 14. Mai 2019, 19:05

[Noir0SX](#) : ja, aber nicht als .aml, sondern als .dsl und im Compiler bekomme ich errors.

Beitrag von „derHackfan“ vom 14. Mai 2019, 19:07

Kein Compile, nur speichern als .aml...

Beitrag von „muhviehstarr“ vom 14. Mai 2019, 19:07

kompiliert doch dann automatisch?

Beitrag von „Roy Jones“ vom 14. Mai 2019, 19:09

es ist genau wie [muhviehstarr](#) sagt, wird automatisch kompiliert.

Beitrag von „derHackfan“ vom 14. Mai 2019, 19:11

Teste bitte mal die hier. 😊

Beitrag von „Roy Jones“ vom 14. Mai 2019, 19:21

Ich danke dir für deine Hilfe, leider greift die .aml bei mir nicht, ich versuche schon ewig ein Patch für den Controller zu finden, leider ohne Erfolg. maciASL zeigt auch ein seltsames Verhalten, weder meine DSDT.aml noch deine SSDT.....aml lassen sich öffnen.

Beitrag von „muhviehstarr“ vom 14. Mai 2019, 19:33

heisst das smbust device auch so?

Beitrag von „Roy Jones“ vom 14. Mai 2019, 19:47

Wie stelle ich das fest ?

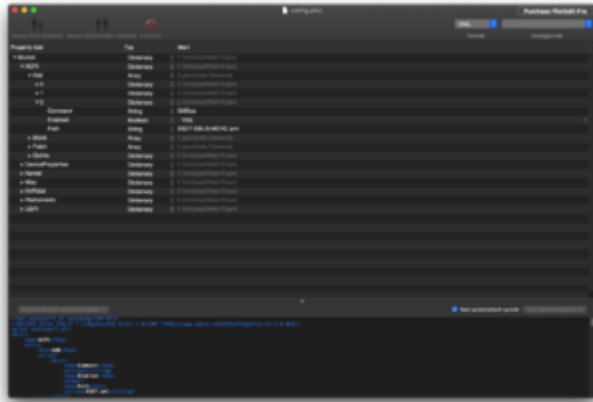
Beitrag von „derHackfan“ vom 14. Mai 2019, 20:58

Hast du sie unter ACPI eingetragen?



Beitrag von „Roy Jones“ vom 14. Mai 2019, 21:38

ja klar, habe ich 😊



Beitrag von „apfelnico“ vom 14. Mai 2019, 22:13

Völlig unnötig diese SSDT. Zumal du das 1:1 schon in deiner DSDT drin hast (Device BUS0 innerhalb von SBUS). Darüber hinaus ist diese SSDT auch so schon merkwürdig. Wozu ist da eine DTGP-Methode eingesetzt, wenn die nirgends aufgerufen wird? Völliger Unfug. Und, benötigst du ein SMBS/SBUS überhaupt? Damit mehr in der PCI-Liste in der Systeminformation steht? 😊

Beitrag von „Roy Jones“ vom 14. Mai 2019, 22:22

[apfelnico](#) : Wie immer sprichst du Klartext und triffst sofort den Kern der Sache, ich danke dir dass du mich aufgeklärt hast, ich stecke leider nicht so tief in der Materie drin um wirklich zu 100% zu wissen was ich brauch und was nicht.

Ich bin einfach nur der naiven Meinung dass alle Bordschnittstellen möglichst an macOS angebunden sein sollten wenn es unterstützt wird.

Danke für deine Info ! 😊

Beitrag von „kuckkuck“ vom 14. Mai 2019, 22:42

Nur weil Geräte im Systembericht erscheinen, sind sie nicht "eingebunden/angebunden". Sie können sowohl erscheinen und nicht richtig konfiguriert sein, als auch nicht erscheinen und vollständig funktionsfähig sein. Der Systembericht ist meist reine Kosmetik... 😊

Beitrag von „derHackfan“ vom 14. Mai 2019, 22:55

[apfelnico](#) führe mich bitte mal ins Helle!

Ich habe stumpf die pcidevices.plist aus dem Hackintosh Tool rübergezogen und nach dem Neustart festgestellt dass der SMBus Treiber nicht geladen wurde, daraufhin habe ich die SSDT-SBUS-MCHC.aml hinzugefügt und siehe da, er wird (jetzt) zu meinem Erfreuen geladen.

Ich wette du würdest bei deiner Möhre nicht eine Sekunde verlieren wenn etwas nicht zu 100% läuft und oder geladen wird, mir persönlich ist es wurscht ob der Zustand für macOS förderlich ist oder ggf. völlig unbedeutend, aber wenn jemand aus der Community einen Wunsch umsetzen möchte dann kann man ihm doch dabei auch beiseite stehen.

Beitrag von „Roy Jones“ vom 15. Mai 2019, 00:12

[derHackfan](#) : das war der Schlüssel = "pcidevices.plist " 😊

Beitrag von „derHackfan“ vom 18. Mai 2019, 23:09

Frisch aus dem macbuild.tool 😊

Beitrag von „derHackfan“ vom 21. Mai 2019, 21:32

OpenCore.efi ist ein bisschen gewachsen. 😊

	vorheriges, 4.6.19		
OpenCore.efi	Heute, 21:17	214 KB	Di
OpenCore.efi*bak	18.05.19, 21:08	210 KB	Di

Beitrag von „Noir0SX“ vom 24. Mai 2019, 15:51

OpenCore Release v1.0.3 november

... <https://github.com/acidanthera/OpenCorePkg/releases> ...

Beitrag von „ammoune78“ vom 26. Mai 2019, 01:01

Hallo freunde

Wer weiß, wie man APIC und Kernel LAPIC als clover zu patchen? Mein Laptop bootet nicht ohne sie.

Vielen dank

Beitrag von „Sascha_77“ vom 26. Mai 2019, 20:54

Mal eine blöde Frage ... wie kriege ich diese ig-platform-id

0x0a260006

zu 4 Byte Hex konvertiert? Möchte meine HD4400 aktivieren.

Beitrag von „anonymous_writer“ vom 26. Mai 2019, 21:24

Umdrehen so wie hier.

Code

1. 0600260a

Beitrag von „bananaskin“ vom 26. Mai 2019, 21:24

wie bitte... 0a 26 00 06 das sind doch 4 Bytes

die 0x davor sagt nur dass diese Code-Folge in Hex-Darstellung ist 😊

Beitrag von „Sascha_77“ vom 26. Mai 2019, 21:38

Ich habe das jetzt mal so hinterlegt:



The screenshot shows a configuration window with two panes. The left pane is titled 'Devices' and lists two entries: 'PciRoot(0x0)/(Pci(0x1b,0x0))' and 'PciRoot(0x0)/(Pci(0x2,0x0))', with the second entry selected. The right pane is titled 'Key*' and lists three entries: 'AAPL,jg-platform-id' with value '0600260A', 'device-id' with value '86801204', and 'framebuffer-patch-enable' with value '01000000'.

Devices	Key*	Value
PciRoot(0x0)/(Pci(0x1b,0x0))	AAPL,jg-platform-id	0600260A
PciRoot(0x0)/(Pci(0x2,0x0))	device-id	86801204
	framebuffer-patch-enable	01000000

Wird aber nicht erkannt. Hab ich was übersehen?

Beitrag von „Noir0SX“ vom 26. Mai 2019, 21:46

... device-id 12040000

Beitrag von „derHackfan“ vom 26. Mai 2019, 22:11

Das kommt jetzt aber mal völlig überraschend... 😊

Beitrag von „dutch64“ vom 26. Mai 2019, 22:16

[NoirOSX](#) Danke!!

Beitrag von „CMMChris“ vom 26. Mai 2019, 22:44

Mackie hat nun auch einen Configurator für OC:
<https://mackie100projects.altervista.org/download/occ/>

Irgendwann muss ich den Loader auch mal testen...

Beitrag von „derHackfan“ vom 26. Mai 2019, 22:51

[CMMChris](#) Sorry aber das Ding ist doch quasi ein alter Hut... 😊

Wenn du den Test mit OpenCore angehst dann halte dich an Xcode, PlistEdit Pro und Co. und wenn ein Configurator dann wäre meine persönliche bitte an dich -> [OpenCore Configurator \(Alpha Software\)](#) und darüber hinaus das ganze dann hier im passenden Unterforum -> [OpenCore](#)

Und dann wünsche ich mir von dir natürlich auch noch jede Menge positive Rückmeldungen.



Beitrag von „CMMChris“ vom 26. Mai 2019, 22:56

Ah ok, hatte hier im Topic nichts davon gesehen deshalb dachte ich dass es noch nicht bemerkt wurde.