

Erledigt

Natives NVRAM auf dem Z390, Durchbruch?

Beitrag von „Altemirabelle“ vom 18. Januar 2020, 19:50

Es kann sein, dass der native Betrieb von NVRAM auf einem board Z390 vom Vorhandensein des PMCR-Geräts in DSDT abhängt ist (im ursprünglichen DSDT ist es nicht vorhanden).

Mehrere Bestätigungen gibt es schon. Versuche, es über SSDT zu injizieren, funktionieren anscheinend nicht, nur das Einfügen in DSDT.

Ich habe über das Wochenende keinen Zugriff auf meinen Z390, aber vielleicht ist jemand bereit das zu testen?

Code

1. Device (PMCR)
2. {
3. Name (_HID, EisaId ("APP9876")) // _HID: Hardware ID
4. Name (_STA, 0x0B) // _STA: Status
5. Name (_CRS, ResourceTemplate () // _CRS: Current Resource Settings
6. {
7. Memory32Fixed (ReadWrite,
8. 0xFE000000, // Address Base
9. 0x00010000, // Address Length
10.)
11. })
12. }

Alles anzeigen

Beitrag von „al6042“ vom 18. Januar 2020, 20:02

Interessante Sache...

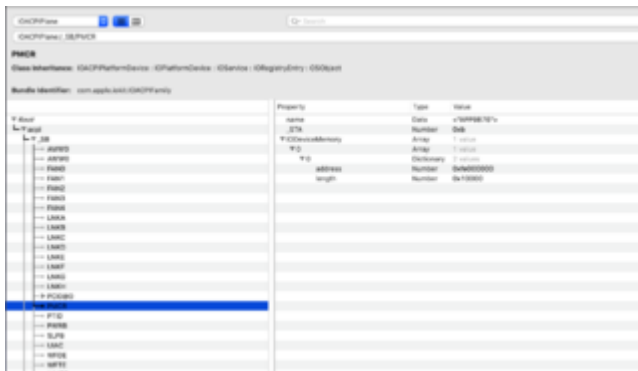
Ich habe den Code mal in die SSDT-AWAC.aml hinzugefügt und neu gestartet.

```

1 1
2 1 * Default ACPI Component Architecture
3 1 * ACPI/ASL Disassembler version 20180413-66290
4 1 * Copyright (c) 1996 - 2018 Intel Corporation
5 1 * Disassembling to non-symbolic legacy ASL operators
6 1 * Disassembly of DSDT.aml, Sat Jan 18 20:01:20 2020
7 1
8 1
9 1 * Original Table Header:
10 1 * Signature "DSDT"
11 1 * Length 00000010 (16K)
12 1 * Revision 0x01
13 1 * Checksum 0x01
14 1 * OEM ID "ACPI"
15 1 * OEM Table ID "Aml"
16 1 * OEM Revision 0x00000000 (0)
17 1 * Compiler ID "ASL"
18 1 * Compiler Version 0x02000000 (2.0.0.0)
19 1
20 1
21 1 OriginalTableLock ("DSDT", 2, "ACPI", "Aml", 0x00000000)
22 1
23 1 External (STAB, Init0) // (from update)
24 1
25 1
26 1 Scope (_SB)
27 1 {
28 1     Method (_INI, 0, NotSerialized) // _INI: Initiation
29 1     {
30 1         If (_OSI ("Darwin"))
31 1             {
32 1                 Start (One, STAB)
33 1             }
34 1     }
35 1 }
36 1
37 1 Device (PNCK)
38 1 {
39 1     Name (_ADR, BitOr (0x00000010, _PCI: Hardware ID)
40 1     Name (_STA, 0x00) // _STA: Status
41 1     Name (_SRG, ResourceTemplate () // _SRG: Current Resource Settings
42 1     {
43 1         Memory32Bit (FixedMemory, // Address Base
44 1             0x00100000, // Address Length
45 1             1)
46 1     }
47 1 }
48 1
49 1
50 1

```

Laut IORegistryExplorer ist das Device jetzt vorhanden:



Beitrag von „Altemirabelle“ vom 18. Januar 2020, 20:16

Bei meinem Bekannten, der die gleiche Asus Prime Z390-A Hauptplatine hat, funktioniert NVRAM, jedoch nur über DSDT.

Auf dem Gigabyte Z390 Designare ebenfalls. Auch Z390 Gaming X.

Beitrag von „al6042“ vom 18. Januar 2020, 20:26

Die Frage wäre dann noch, mit welchem Bootloader bei denen gearbeitet wird, bzw. welche .efi-Treiber eingesetzt werden.

Die Theorie besagt ja, dass mit dem Device und ohne den EmuVariableUefi.efi die NVRAM Unterstützung funktionieren sollte, oder?

Wie teste ich das im Einzelnen mit OpenCore, bei dem es ja diesen .efi-Treiber nicht gibt, da von FwRuntimeServices.efi abgelöst.

Diesen würde ich nun auch ungern entfernen, da die Büchse eben auch sauber läuft.

Hast du zufällig auch Auszüge des IORegs von den Kisten, bei denen das Device über die DSDT eingebunden wurden?

Beitrag von „mhaeuser“ vom 18. Januar 2020, 20:33

[al6042](#) einfach DisableVariableWrite aus

Beitrag von „al6042“ vom 18. Januar 2020, 20:45

Manchmal kann es so einfach sein... 🤔

Ich checke das gleich mal durch...

EDIT:

Scheint über die SSDT nicht zu funktionieren, bleibt bei "PCI configuration end" hängen:



Beitrag von „cobanramo“ vom 18. Januar 2020, 20:48

Ich häng mich mal da auch ein.. wird interessant.

[al6042](#) bei dem photo scheint aber dein "SSDT-PLUG" auch nicht zu funktionieren...

Gruß Coban

Beitrag von „al6042“ vom 18. Januar 2020, 20:54

Interessant...

laut IOReg macht es aber was es soll... da muss ich später nochmal drauf schauen.

Beitrag von „NoBody_0“ vom 18. Januar 2020, 22:12

um den Code zu testen soll nun einfach wie Download-Fritz sagte tun.
muss nvram.plist auch gelöscht werden und LegacyEnable aus!!?

Update:

Ich habe so gemacht

1. DisableVariableWrite aus
2. LegacyEnable aus
3. LegacyOverwrite aus
4. LogoutHook.command und Nvram.plist gelöscht

SSDT.PMCR.aml implementiert



Es scheint bis jetzt zu funktionieren aber 100% bestätigen noch nicht, da unter Test besteht



[mhaeuser](#) [al6042](#) [cobanramo](#)

Beitrag von „cobanramo“ vom 18. Januar 2020, 23:01

Heisst das das du jetzt ohne emuliertes Nvram eine variable setzen kannst und nach neustart dieser auch da ist?

schon getestet?

```
sudo -s
```

```
nvram -c
```

```
nvram myvar=test
```

exit

nach neustart;

```
nvrnm -p | grep -i myvar
```

Gruss Coban

Beitrag von „al6042“ vom 18. Januar 2020, 23:06

Ich habe damit heute Abend nicht weitergemacht, nachdem die SSDT-Geschichte wohl nicht lief...

Beitrag von „NoBody_0“ vom 18. Januar 2020, 23:15

das habe ich als erstes getestet 🤔👉

[cobanramo](#)

Beitrag von „mhaeuser“ vom 19. Januar 2020, 08:11

muster48 Altemirabelle Bitte deklariert (oder ggf. löscht) eure ACPI-Sachen als keine empfohlene Dauerlösung... Die Gerätebeschreibung hier stammt von Apple (zu erkennen an der EISA-ID) und nicht von AMI oder dergleichen, bei sowas kann man nie wissen, wie lange das gutgeht. Morgen schaut sich Vandroiy die Lage an und bietet dann sehr wahrscheinlich noch am selben Tag eine entsprechende Tabelle in der Acidanthera-Repo an, sollte alles reproduzierbar funktionieren. Bei AMI heißt der Controller wohl oft PPMC, falls sich jemand den neusten Z390- (manche sagen, NVRAM funktioniert bei ihnen nach einem Update) oder Z370-

Kram anschauen will.

Aber natürlich danke für's Teilen!

Beitrag von „NoBody_0“ vom 19. Januar 2020, 09:57

danke für deine Meldung [mhaeuser](#)

Aber nach 12 Stunden scheint es nicht nicht zu funktionieren, da ich mich nach dem Start wieder bei Apple anmelden muss und alle Fenster öffnen wieder, die schon offen beim Herunterfahren waren, obwohl ich das Häkchen Beim nächsten Anmelden alle Fenster wieder öffnen rausgenommen habe, und ich glaube das hat mit nvram was zu tun oder?!



Beitrag von „mhaeuser“ vom 19. Januar 2020, 10:02

Puh, keine Ahnung, ob das wirklich mit dem NVRAM zusammenhängt. Morgen wird jedenfalls von Vandroy alles durchgetestet. Wo kommt der Schnipsel eigentlich her?

Beitrag von „Altemirabelle“ vom 19. Januar 2020, 10:24

Der Schnipsel kommt von einem user CODYQX4, der das zufällig auf Z390 Designare entdeckt hat. Natürlich von der Seite auf T 😊

Beitrag von „danielosx86“ vom 19. Januar 2020, 15:00

Hello, new SSDT (PPMC+PMCR) is better solution - reboot and shutdown works fine.

Beitrag von „DSM2“ vom 19. Januar 2020, 15:03

Kann mich natürlich täuschen aber sieht für mich wie ein Schnipsel von Osy86 aus.

Beitrag von „HackMac1“ vom 19. Januar 2020, 15:08

[Zitat von danielosx86](#)

Hello, new SSDT (PPMC+PMCR) is better solution - reboot and shutdown works fine.

It looks like its worked by mine Asus Prime Z390-A

Beitrag von „CMMChris“ vom 19. Januar 2020, 15:29

[Zitat von muster48](#)

da ich mich nach dem Start wieder bei Apple anmelden muss und alle Fenster öffnen wieder, die schon offen beim Herunterfahren waren, obwohl ich das Häckchen Beim nächsten Anmelden alle Fenster wieder öffnen rausgenommen habe, und ich glaube das hat mit nvram was zu tun oder?!

Nein hat es nicht

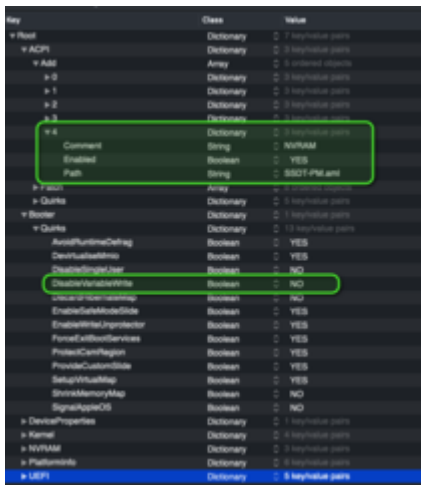
Beitrag von „NoBody_0“ vom 19. Januar 2020, 15:43

[CMMChris](#) okay danke für deine Meldung

Woran liegt es denn?

Beitrag von „al6042“ vom 19. Januar 2020, 16:28

Also... ich habe das eben nochmal mit der obigen SSDT getestet und dabei folgendes in meiner config.plist eingebaut:



Und was soll ich sagen... die Sch****e geht... 😊

Code

1. Last login: Sun Jan 19 16:19:04 on console
2. al6042@al6042-iMac ~ % nvram -p
3. bluetoothInternalControllerInfo %90%82%ac%05%00%00%00%b0%14%a4^`%d6%c14
4. fmm-computer-name al6042-iMac
5. prev-lang:kbd de:0%00
6. bluetoothActiveControllerInfo
 %90%82%ac%05%00%00%00%00%00%b0%14%a4^`%d6%c14
7. SystemAudioVolumeDB %f5
8. SystemAudioVolume H
9. TestVar2 MeinTest2

10. LocationServicesEnabled %01
11. TestVar MeinTest
12. boot-args shikigva=1 keepsyms=1 dart=0 debug=0x100 brcmfx-country=#a
13. csr-active-config g%00%00%00

Alles anzeigen

Beitrag von „CMMChris“ vom 19. Januar 2020, 16:33

Habe es jetzt auch mal bei jemandem auf nem Z390er getestet und ja, nativer NVRAM Zugriff vorhanden. Hammer.

Beitrag von „HackMac1“ vom 19. Januar 2020, 16:37

[Zitat von al6042](#)

Also... ich habe das eben nochmal mit der obigen SSDT getestet und dabei folgendes in meiner config.plist eingebaut:

interessanterweise musste ich bei mir noch folgenden Kernelpatch hinzufügen:

Code

1. <key>Patch</key>
2. <array>
3. <dict>
4. <key>Base</key>
5. <string></string>
6. <key>Comment</key>
7. <string>AppleRTC patch</string>
8. <key>Count</key>
9. <integer>1</integer>
10. <key>Enabled</key>
11. <true/>
12. <key>Find</key>

13. <data>dTMPtw==</data>
14. <key>Identifizier</key>
15. <string>com.apple.driver.AppleRTC</string>
16. <key>Limit</key>
17. <integer>0</integer>
18. <key>Mask</key>
19. <data></data>
20. <key>MaxKernel</key>
21. <string></string>
22. <key>MinKernel</key>
23. <string>18.5.0</string>
24. <key>Replace</key>
25. <data>6zMPtw==</data>
26. <key>ReplaceMask</key>
27. <data></data>
28. <key>Skip</key>
29. <integer>0</integer>
30. </dict>
31. </array>

Alles anzeigen

Andernfalls erscheint bei jedem Neustart "The system has POSTed in safe mode". Unter ACPI ist der RTC-dingens 300 Patch aktiv.

Beitrag von „al6042“ vom 19. Januar 2020, 16:42

Clover oder OpenCore?

Das geht aus deiner Signatur nicht hervor.

Beitrag von „Doctor Plagiat“ vom 19. Januar 2020, 16:51

[Zitat von al6042](#)

ich habe das eben nochmal mit der obigen SSDT getestet

Welche denn von den beiden? Die SSDT-PMCR oder die SSDT-PM.

Beitrag von „al6042“ vom 19. Januar 2020, 16:52

Die SSDT-PM.aml. da mein gestriger Versuch mit einbinden in die SSDT-AWAC.aml nicht so wirklich geholfen hatte.

Beitrag von „HackMac1“ vom 19. Januar 2020, 17:00

[Zitat von al6042](#)

Clover oder OpenCore?

Das geht aus deiner Signatur nicht hervor.

Sorry, OpenCore 0.54. Ich verwende deine aus dem OpenCore Sammelthread

Beitrag von „al6042“ vom 19. Januar 2020, 17:05

Die habe ich eben aktualisiert... 😊

Beitrag von „Doctor Plagiat“ vom 19. Januar 2020, 17:07

[al6042](#) Das ist ja der Wahnsinn! Es funktioniert !!! 👍

```
csr-active-config      g%00%00%00
boot-args              debug=0x100 keepsyms=1
myvar test
```

Beitrag von „al6042“ vom 19. Januar 2020, 17:10

Ich bin unschuldig... habe es nur getestet... 😊

Beitrag von „Doctor Plagiat“ vom 19. Januar 2020, 17:12

Mehr habe ich auch nicht getan, aber das ist doch auch wichtig.

Beitrag von „Altemirabelle“ vom 19. Januar 2020, 17:34

Ja bei mir auch positiv, zuerst hab ich High Sierra getestet. NVRAM Test: positiv,
Also auf Gigabyte Z390 M GAMING reicht nur die SSD. Die SSDT-PM.aml

EDIT: Mojave auch **positiv!**

Beitrag von „NoBody_0“ vom 19. Januar 2020, 18:15

[al6042](#)

nachdem du die SSDT.PM.aml getestet hast, taucht Device PPMC im IOReg.?

Beitrag von „al6042“ vom 19. Januar 2020, 18:20

Beide Einträge sind im IOReg zu sehen:



Beitrag von „macdream“ vom 19. Januar 2020, 18:38

n'Abend allerseits,

wo liegt der Vorteil gegenüber EmuVariableUefi.efi? Ist es nur ein "fremder" Treiber weniger, oder hat der Patch sonstige Vorteile?

Wie würde man das Ganze in Clover einbauen ?

Beitrag von „Tirola“ vom 19. Januar 2020, 18:54

Kann man diese ssdt erstellen? Wenn ja wie?

Beitrag von „al6042“ vom 19. Januar 2020, 18:55

Ich habe keine eigene SSDT erstellt, sondern die aus [#17](#) genutzt.

Beitrag von „Tirola“ vom 19. Januar 2020, 19:02

Funktioniert das mit clover genauso wie mit opencore?

Beitrag von „DSM2“ vom 19. Januar 2020, 19:07

Welche Gründe sprechen den dagegen?

Beide können mit einer einer SSDT was anfangen.

Beitrag von „JimSalabim“ vom 19. Januar 2020, 19:31

Läuft bei mir. Z390 Designare mit Clover. Ich verwende FwRuntimeServices und OCQuirks. Hab meine OCQuirks.efi entsprechend dem Screenshot von [al6042](#) angepasst, die EmuVariable.efi sowie die nvram.plist gelöscht und auch die Clover-RC-Skripte aus dem /etc-Ordner entfernt. Die SSDT-PM hab ich in CLOVER/ACPI/patched kopiert. Das war's.

Getestet wie folgt:

```
sudo nvram MyVar=Depp
```

Nach dem Neustart

```
sudo nvram -p | grep "MyVar"
```

ergibt

MyVar Depp

Sehr schön. 😊

Beitrag von „Toskache“ vom 19. Januar 2020, 19:39



Das ist sicherlich klasse! Oder?

Kann das jemand in "Nutzen" übersetzen? Was geht jetzt besser oder anders?

Beitrag von „Altemirabelle“ vom 19. Januar 2020, 19:41

Super @[JimSalabim](#) und Gruß nach Erlangen, hab mal dort gearbeitet.

Beitrag von „macdream“ vom 19. Januar 2020, 19:46

[Toskache](#)

Ja, das würde mich auch interessieren.

[JimSalabim](#)

▮ [Zitat von JimSalabim](#)

Beitrag von „macdream“ vom 19. Januar 2020, 19:57

Ja, jetzt kommt die Erleuchtung, die QcQuirks.plist hatte ich nicht im Sinn, die nutze ich erst seit kurzem. Zuvor immer ein kompilierte "Version" ohne plist genutzt.

Vielen Dank.

Ach, hast du auch eine Idee warum diese Lösung besser ist als die EmuVariable Lösung? Wenn ich recht verstehe, machen doch beide das Gleiche?

Beitrag von „JimSalabim“ vom 19. Januar 2020, 20:06

Wie ich sehe, brauche ich jetzt auch slide=0 nicht mehr. Ich bilde mir ein, dass das vorher notwendig war, zumindest wenn man auf dem Designare-Board mit aktivierter iGPU ohne Memory Allocation Error booten wollte.

[macdream](#) Wo in der Praxis die genauen Vorteile liegen, muss jemand beantworten, der sich damit besser auskennt als ich.

Beitrag von „Doctor Plagiat“ vom 19. Januar 2020, 20:18

Ich hatte die Hoffnung, dass mit "funktionierendem NVRAM" mein Boot-Volume Catalina an erster Stelle verbleibt. Hat sich leider nicht erfüllt.

Linux kann ich booten so oft wie ich will, nach einem Neustart ohne Auswahl startet macOS, so wie es sein soll.

Sobald ich einmal Windows starte, schiebt sich dieses OS in der Bootreihenfolge nach vorne.

Beitrag von „anonymous_writer“ vom 19. Januar 2020, 20:21

Hi [Doctor Plagiat](#) ,

versuche mal diese Version. Damit bleibt immer das zuletzt ausgewählte aktiv, wenn das auch reicht.

<https://github.com/n-d-k/OpenCorePkg>

Beitrag von „Doctor Plagiat“ vom 19. Januar 2020, 20:22

Die habe ich gerade im Einsatz, geht leider auch nicht.

Beitrag von „anonymous_writer“ vom 19. Januar 2020, 20:25

Hm, schade. 🙄

Hat bei mir aber auch vorher schon geklappt.

Ich hatte das auch mal, habe aber keine Idee was ich geändert habe. 🤖

Beitrag von „kneske“ vom 19. Januar 2020, 20:27

Nochmal bitte für Blöde...

OCQuirks aktiv und Clover...was muss genau passieren damit NVRAM nativ wird? Oder genutzt werden kann?

Die SSDT und OCQuirks.plist alleine bewirken aber nicht, dass es sich einen Testeintrag bei mir speichert...

Beitrag von „Doctor Plagiat“ vom 19. Januar 2020, 20:29

Merkwürdig dass sich nur Windows vordrängt. Schade dass man keine Variable dafür setzen kann.

In Clover hat es immer mit dem Auswählen des Bootvolumens funktioniert, aber ich will natürlich nicht mehr zu Clover zurück.

Beitrag von „anonymous_writer“ vom 19. Januar 2020, 20:33

Hallo [Doctor Plagiat](#) ,

versuche testweise ob mit diesem Bootloader das Problem behoben wäre. Der kann alles Booten Clover, OC und sogar eine Windows das mit TPM Bitlocker verschlüsselt ist. Letzteres geht bei mir nur mit diesem Bootloader.

<https://www.rodsbooks.com/refind/>

<https://sourceforge.net/projects/refind/>

Beitrag von „mhaeuser“ vom 19. Januar 2020, 20:38

[Doctor Plagiat](#) Wird Windows *in* oder *statt* OpenCore der Standard?

EDIT: An der Stelle ganz gelegen, möchte ich betonen, dass es für den n-d-k-Fork selbstverständlich keinen Support gibt

Beitrag von „Doctor Plagiat“ vom 19. Januar 2020, 20:38

Ja Danke, ich kenne refind, wollte jetzt aber nicht noch einen Bootloader vorschalten.

Beitrag von „anonymous_writer“ vom 19. Januar 2020, 20:39

Ging auch mehr um den Test ob es damit weg ist das Problem.

Beitrag von „Doctor Plagiat“ vom 19. Januar 2020, 20:42

[mhaeuser](#) Windows drängelt sich im UEFI-Boot-Mangement an die erste Stelle und verschiebt OpenCore an die zweite Stelle. Ist jetzt vielleicht etwas blöd ausgedrückt.

Wenn ich Linux botte passiert das nicht, da startet dann automatisch wieder OpenCore.

Beitrag von „mhaeuser“ vom 19. Januar 2020, 20:45

[Zitat von Doctor Plagiat](#)

In Clover hat es immer mit dem Auswählen des Bootvolumes funktioniert, aber ich will natürlich nicht mehr zu Clover zurück.

Dann versteh ich die Aussage hier nicht... wenn Windows vor Clover geschoben wird, was bringt es dann irgendwas in Clover auszuwählen? Ist doch das selbe Fehlerbild wie mit OC, oder sollte jedenfalls so sein.

Beitrag von „Doctor Plagiat“ vom 19. Januar 2020, 20:47

Als ich Clover noch genutzt habe hat das Auswählen des Bootvolumens immer funktioniert, benutze ich ja aber nicht mehr und möchte das auch nicht.

n-d-k habe ich nur mal getestet, ob es da auch passiert. Verstehe dass es dafür keinen Support gibt.

Beitrag von „mhaeuser“ vom 19. Januar 2020, 20:51

Wieso sollte die Auswahl des Bootvolumens mit OC nicht funktionieren? Ich verstehe nur Bahnhof.

Beitrag von „Doctor Plagiat“ vom 19. Januar 2020, 21:02

Dann versuche ich es nochmal zu erklären.

Wenn ich macOS mit OC starte und anschließend in Systemeinstellungen -> Startvolume die Catalina-SSD als Boot-Volume auswähle, sollte doch immer automatisch Catalina booten, es sei denn ich wähle ein anderes Betriebssystem.

Leider funktioniert das nur bis ich mal Windows ausgewählt habe, danach bootet automatisch Windows. Ich muss es in der UEFI-Firmware oder in Linux mit efibootmgr wieder umstellen.

EDIT: mit automatisch meine ich vorausgewählt.

Beitrag von „mhaeuser“ vom 19. Januar 2020, 21:04

Richtig, ich verstehe aber nicht, inwiefern die Lage mit Clover besser sein soll als mit OC. Dass

die Firmware Windows vorschreibt sollte BL-unabhängig und nicht einfach behebbar sein.

Beitrag von „schmalen“ vom 19. Januar 2020, 21:12

Habe auch die Daten von [al6042](#) aus seinem Post übernommen, und siehe da, es funktioniert auch bei meinem Z390er.

Beitrag von „Doctor Plagiat“ vom 19. Januar 2020, 21:14

[mhaeuser](#)

Da stecke ich nicht tief genug in der Materie um das zu verstehen. Ich kann da nur mitteilen, dass man in Clover einstellen konnte welches bootbare Medium vorausgewählt werden soll und das hat dann immer funktioniert. Auch wenn man zwischendurch mal ein anderes BS hochgefahren hat, startete danach wieder macOS, da es in Clover vorausgewählt war.

Mit OpenCore funktioniert das zwar wenn ich zwischendurch mal Linux starte, aber sobald ich einmal Windows hochgefahren habe, funktioniert es nicht mehr.

Beitrag von „mhaeuser“ vom 19. Januar 2020, 21:19

[Doctor Plagiat](#) Sorry, kann dazu nichts mehr sagen. Was *im* Bootloader eingestellt ist, sollte doch vollkommen egal sein, weil du beschreibst, dass Windows sich *vor* den Bootloader drängelt. Wenn Windows vor OC ist, ist egal, was OC als Standard auswählt, genau wie in Clover die Standardauswahl egal wäre, würde Windows vor diesen gesetzt werden. RequestBootVarRouting ist aber an, oder?

Beitrag von „Altemirabelle“ vom 19. Januar 2020, 21:23

Nur ein Info, die SSDT-PM.aml funktioniert auf dem Asus Prime Z390-A, nachdem das Ding als erstes geladen wird.

Beitrag von „Doctor Plagiat“ vom 19. Januar 2020, 21:32

[mhaeuser](#)

RequestBootVarRouting steht auf true

Ich dachte was im Bootloader eingestellt ist, bleibt dann auch gültig.

Irgendwie habe ich das Gefühl jeder von uns spricht eine andere Sprache. Du der Profi stehts ganz tief in der Materie und ich der einfache Anwender kratze eben nur an der Oberfläche.

Konnte mir nicht vorstellen, dass es so schwer ist ein Problem zu beschreiben.

Trotzdem DANKE! dass du versucht hast mich zu verstehen um eventuel eine Lösung herbei zuführen.

Beitrag von „apfelnico“ vom 19. Januar 2020, 21:34

PMCR bzw PMC1 hatte ich schon immer aktiv, nur dass es eben hier auf dem X299 nativ dazu gehört. Wusste nicht, dass es der Z390-Plattform so helfen kann. Wen es interessiert (Original aus Asus Prime X299 Deluxe), ich habe es mal aus DSDT und SSDT herausgepopelt und isoliert zum probieren:

Beitrag von „macdream“ vom 19. Januar 2020, 22:04

[mhaeuser](#)

[Zitat von Download-Fritz](#)

Doctor Plagiat Sorry, kann dazu nichts mehr sagen. Was *im* Bootloader eingestellt ist, sollte doch vollkommen egal sein, weil du beschreibst, dass Windows sich *vor* den Bootloader drängelt. Wenn Windows vor OC ist, ist egal, was OC als Standard auswählt, genau wie in Clover die Standardauswahl egal wäre, würde Windows vor diesen gesetzt werden. RequestBootVarRouting ist aber an, oder?

Das kann ich nur bestätigen, ich habe mit Clover exakt das gleiche Verhalten. Das im BIOS voreingestellte (Clover)-Laufwerk wird immer zuverlässig angesprochen, wenn ich jedoch Win10 einmal boote, ist beim nächsten Start das Win10 Laufwerk im BIOS voreingestellt. Es übrigens auch egal, ob man Windows mit Shift oder Ctrl runterfährt.

Der Booteintrag im BIOS wird geändert. Da können weder der Clover noch der OC-Bootloader was dran ändern.

Beitrag von „al6042“ vom 19. Januar 2020, 22:04

Altemirabelle

Bei mir muss der Eintrag nicht als erstes geladen werden.

Ich habe den Eintrag einfach hinten dran geklemmt.

Beitrag von „Altemirabelle“ vom 19. Januar 2020, 22:33

Ach so ich dachte es geht bei dir nicht. 😊

Beitrag von „kaneske“ vom 19. Januar 2020, 22:37

Läuft auf Z390 Phantom Gaming ITX, musste aber sorted Order in Clover haben.
SSDT-PM.aml auf 1.Stelle, dann geht es.

Beitrag von „kuckkuck“ vom 19. Januar 2020, 22:44

[Zitat von Doctor Plagiat](#)

Ich dachte was im Bootloader eingestellt ist, bleibt dann auch gültig.

Zwei Sachen:

Erstmal allgemein was DF wahrscheinlich meint:

Sehen wir es mal vereinfacht so: dein Computer startet -> dein Computer startet (weil als standard ausgewählt) OpenCore vom Datenträger -> OpenCore startet nun (weil als Standard in macOS ausgewählt) macOS. MacOS startet also nur, nachdem überhaupt OpenCore die Kontrolle hatte.

Wenn du jetzt Windows bootest und Windows schiebt sich im BIOS vor den OpenCore Datenträger, dann sieht es so aus: dein Computer startet -> dein Computer startet (weil jetzt als Standard ausgewählt) Windows direkt vom Datenträger. OpenCore kommt insofern garnicht zum Zug und kann deswegen nichts an der Situation ändern. Windows wird an OpenCore "vorbei" gestartet.

Die andere Sache ist: vielleicht musst du dein Problem einfach mal ein bisschen besser beschreiben. Der * symbolisiert im OC BootPicker ja die Standard-Bootoption. Was ist jetzt das Problem: nach einem Windows Start steht das * vor Windows und nicht mehr vor macOS oder sonstwo? Oder, nach einem Windows Start startet automatisch Windows und der Bootpicker erscheint garnicht, bei manuellem Start vom OC Datenträger ist die Standard-Bootoption aber nach wie vor macOS?

Beitrag von „mhaeuser“ vom 19. Januar 2020, 22:46

[Zitat von macdream](#)

[mhaeuser](#)

Da können weder der Clover noch der OC-Bootloader was dran ändern.

Genau das meinte ich und deswegen, [Doctor Plagiat](#), verstehe ich deine Aussagen zu Clover leider nicht. Wenn OC nicht mehr Standard ist (und somit keinerlei Kontrolle haben *kann*), liegt es auch nicht an OC, wenn der neue Standard statt dem gewünschten startet. Analog dürfte keine Option in Clover Einfluss auf dieses Verhalten haben

Beitrag von „JimSalabim“ vom 19. Januar 2020, 23:55

[Zitat von macdream](#)

Das kann ich nur bestätigen, ich habe mit Clover exakt das gleiche Verhalten. Das im BIOS voreingestellte (Clover)-Laufwerk wird immer zuverlässig angesprochen, wenn ich jedoch Win10

einmal boote, ist beim nächsten Start das Win10 Laufwerk im BIOS voreingestellt. Es übrigens auch egal, ob man Windows mit Shift oder Crtl runterfährt.

Der Booteintrag im BIOS wird geändert. Da können weder der Clover noch der OC-Bootloader was dran ändern.

Bootest du Windows über Clover oder über F12 vom Bios aus? Wenn ich Windows über Clover boote, habe ich das Problem nicht. Beim Boot mit F12 kommt es mir jedoch ebenfalls bekannt vor. Dann drängelt sich die Windows-Platte gerne mal in den Boot Option Priorities im Bios vor. Das hat jedoch mit den Clover- oder OpenCore-Einstellungen logischerweise überhaupt nichts zu tun.

Beitrag von „macdream“ vom 20. Januar 2020, 09:16

Guten Morgen,

Bislang habe ich immer über Clover gebootet, aber auch wenn ich über F12 das Bootlaufwerk ändere passiert das gleiche. Ich vermute Windows ändert die Bootreihenfolge ungefragt, das ist wohl eine der "Segnungen" des UEFI-BIOS.

Beitrag von „thomaso66“ vom 20. Januar 2020, 09:45

[Zitat von kaneske](#)

Läuft auf Z390 Phantom Gaming ITX, musste aber sorted Order in Clover haben.

SSDT-PM.aml auf 1.Stelle, dann geht es.

Mit OC ist die Stelle an der es eingefügt wird egal, ich habe es einfach am Ende hinzugefügt.

Thema Thema Windows drängelt sich bei der Boot Reihenfolge vor im UEFI.

Das tritt aber unabhängig vom verwendeten Bootloader seit dem Letzen Update von Windows bei mir auf.

Beitrag von „locojens“ vom 20. Januar 2020, 09:53

Danke für diesen Durchbruch!! Habe bei meinem ASUS ROG STRIX Z390-F Gaming mit OpenCore 0.5.5 nun auch beschreibbares NVRAM (NVRAM als solches war ja schon immer da).

Beitrag von „elmacci“ vom 20. Januar 2020, 10:10

Kurz auch von mir ein Erfolgsbericht zu dieser Methode - allerdings nicht auf einem Z390-Mainboard, sondern einem Laptop. Ich gehe davon aus das hier gewisse hardwarespezifische

Ähnlichkeiten existieren (WhiskeyLake CPU i7 8565U) und es deshalb auch funktioniert - zumindest war das der Hintergedanke bei mir um es mal mit dieser Methode zu probieren.

Hintergrund:

Habe bei meinem Razer Blade Stealth 2019 das Problem gehabt dass ein Start des Systems mit OptioV3, AptioMemory oder QcQuirks (mit verschiedenen Settings in der QcQuirks.plist) funktioniert, aber bei einem Restart oder beim Herunterfahren eine Kernel Panic geschieht. Konnte man wundervoll nachvollziehen mit keepsyms=1 bootarg.

Die einzige Möglichkeit das zu umgehen war der Einsatz von EmuVariable in Verbindung mit den RC Skripten.

Hinzunahme von SSDT-PM und Anpassung der QcQuirks.plist wie von [al6042](#) beschrieben hat dazu geführt das:

- NVRAM nativ funktioniert (Variable gesetzt, nach Neustart vorhanden - ebenso merkt sich Clover das vorher ausgewählte Bootvolume)
- Herunterfahren und Neustart ohne Kernel Panic funktioniert.

Wirklich schöne Sache!

Interessanterweise habe ich im System nur das Device PMCR über IORegistry, aber nicht das Device PPMC.

Trotz Definition in der AML.

cheers

Beitrag von „karacho“ vom 20. Januar 2020, 10:29

An die, wo Windows sich vordrängt. Testet das einmal. Stellt im Bios bei den Bootoptionen

alle Laufwerke, außer dem OC Laufwerk, auf disabled.

Beitrag von „mhaeuser“ vom 20. Januar 2020, 10:30

So, die Acidanthera-SSDT ist da: <https://github.com/acidanthera.../AcpiSamples/SSDT-PMC.dsl>
Ausnahmsweise ist es Mal korrekt, das Gerät als Apple-Gerät zu identifizieren, weil der Treiber explizit darauf prüft. Der PMCR-Teil, den ich erwähnt hatte, ist doch nicht nötig, er ist wegen Umstellungen im Chipsatz weggefallen - der Teil eurer SSDTs sollte nichts bewirken (positiv oder negativ). Die Acidanthera-SSDT deaktiviert den Apple-PMC für alle anderen Oses.

Beitrag von „CMMChris“ vom 20. Januar 2020, 10:36

Interessant wäre nun noch zu erfahren warum genau das Device den Zugriff auf nativen NVRAM erlaubt. Für mich als Laien in dem Bereich ergibt das nicht so wirklich Sinn da ich es so verstehe, dass der Kernel Slide Kram noch eine Ebene niedriger ist als die ACPI Interpretierung und somit vorher stattfindet. Kannst du das vielleicht mal erläutern [mhaeuser](#) ?

Beitrag von „mhaeuser“ vom 20. Januar 2020, 10:46

[CMMChris](#) Wissen wir tatsächlich selbst nicht. Der PMC (siehe Datenblatt) hat einige private Register ("Chipset Initialization Register"), mit denen es zu tun haben könnte. Wir wissen ja nicht Mal wirklich, was das Problem war... ein Kern (der, der den SetVariable-Aufruf angestoßen hat) ist in eine Deadloop geraten - was er da getrieben hat, weiß niemand genau.

Beitrag von „Altemirabelle“ vom 20. Januar 2020, 11:06

Noch ein positiver Effekt, man braucht vermutlich "slide=0" nicht mehr. Noch zu früh das definitiv zu sagen. Aber 2 Neustarts mit Mojave und High Sierra und der Rechner arbeitet wie

gewöhnt.

Beitrag von „locojens“ vom 20. Januar 2020, 11:12

Altemirabelle "slide=0" war noch wofür? Welcher Bootloader, welches System? Ich habe als einziges Argument "alcid=7" drin bei OC 0.5.5 ...

Beitrag von „CMMChris“ vom 20. Januar 2020, 11:14

slide=X ist unabhängig vom Bootloader und steuert KASLR (Kernel Slide).

Beitrag von „locojens“ vom 20. Januar 2020, 11:20

ach Aptiofix jetzt hab ichs, Clover

Beitrag von „Doctor Plagiat“ vom 20. Januar 2020, 12:15

[Zitat von kuckkuck](#)

vielleicht musst du dein Problem einfach mal ein bisschen besser beschreiben.

Ich war der Meinung das Problem richtig, ausführlich und deutlich beschrieben zu haben. Sorry wenn das nicht der Fall war.

[Zitat von Download-Fritz](#)

liegt es auch nicht an OC, wenn der neue Standard statt dem gewünschten startet.

Das ist klar. Wenn sich der Windows-Boot-Manager im Bios den ersten Platz gesichert hat, kommt OC gar nicht zum Zug, egal wo der Picker steht.

Ich hatte nur geglaubt, dass "funktionierender NVRAM" dafür sorgt, dass OpenCore im Bios immer an oberster Stelle steht und damit immer zuerst gebootet wird. OK, geht nicht, da war ich wohl zu naiv.

Ich habe die letzte Stunde damit verbracht die neueste master von acidanthera zu downloaden, kompilieren zu lassen, aktuelle kexts und drivers eingefügt und die config geprüft und überarbeitet.

Das hat sich gelohnt, es funktioniert jetzt. OC startet auch nach einem Windows-Besuch, sofern man Windows über OC bootet. Der Picker bleibt auch nach einem Windows-Boot immer bei Catalina stehen.

Nur wenn ich über F11 (AsRock-Bios-Bootmenü) Windows starte, mogelt sich das BS im Bios wieder ganz nach oben.

DANKE für die Unterstützung.

Beitrag von „pstr“ vom 20. Januar 2020, 12:24

[Doctor Plagiat](#) wo befindet sich Dein Windows nochmal ? Auf eine separaten Platte/SSD oder auf dem gleichen Laufwerk ? Vielleicht einen anderen Thread eröffnen, hat mit nativem NVRAM nichts zu tun.

Beitrag von „Altemirabelle“ vom 20. Januar 2020, 12:34

[@locojens](#)

slide=X ist unabhängig vom Bootloader ... und unabhängig von der macOS-Version. Hab weitere 2 Neustarts gemacht. Funktioniert.

Benutze noch Clover.

Beitrag von „JimSalabim“ vom 20. Januar 2020, 12:55

Ich mache das zwar sowieso nach jedem EFI-Ordner-Update, aber ich habe festgestellt, dass nun nach einem [Clover-Update](#) oder nach dem Switchen zwischen verschiedenen EFI-Ordnern immer ein NVRAM Reset notwendig ist, sonst bootet er nicht. Ergibt das Sinn?

Beitrag von „revunix“ vom 20. Januar 2020, 12:58

Bei mir funktioniert das ganze nun auch super.

NVRAM Emulation rückgängig machen: `sudo defaults delete com.apple.loginwindow LogoutHook`

OpenCore Settings:

DisableVariableWrite = No

LegacyEnable = No

Und dann funktioniert es auch mit dem AutoBoot.

wo befindet sich Dein Windows nochmal ?

auf einer seperaten SSD.

[Zitat von pstr](#)

Vielleicht einen anderen Thread eröffnen

Hat sich doch erledigt. Siehe Post #89

[Zitat von karacho](#)

Stellt im Bios bei den Bootoptionen alle Laufwerke, außer dem OC Laufwerk, auf disabled.

Habe ich noch nicht getestet, da gerade erst gelesen.

Aber was soll da passieren? Da in diesem Fall OC das einzige bootbare Medium ist kann nur OC starten.

PS: Wer z.B. 2 NVMe 's im Rechner hat, kann das nicht testen, da sich eine NVMe-SSD nicht deaktivieren lässt. Ist jedenfalls auf meinem Board so.

Beitrag von „revunix“ vom 20. Januar 2020, 13:47

[Zitat von Doctor Plagiat](#)

PS: Wer z.B. 2 NVMe 's im Rechner hat, kann das nicht testen, da sich eine NVMe-SSD nicht deaktivieren lässt. Ist jedenfalls auf meinem Board so.

OT: Bei meinem Board hat ASUS diese Funktion mit UEFI Updates wieder entfernt. Richtig nervig wenn man die NVMe unter der Grafikkarte hat...

Beitrag von „karacho“ vom 20. Januar 2020, 13:49

Zitat von Doctor Plagiat

PS: Wer z.B. 2 NVMe 's im Rechner hat, kann das nicht testen, da sich eine NVMe-SSD nicht deaktivieren lässt.

Da hast du Recht. Hatte ich nicht bedacht in meinem Post.

Beitrag von „Doctor Plagiat“ vom 20. Januar 2020, 13:54

Zitat von Un!x

Bei meinem Board hat ASUS diese Funktion mit UEFI Updates wieder entfernt.

Ist ja interessant dass es bei ASUS mal ging, aber die werden ihre Gründe gehabt haben, wenn sie es wieder entfernt haben. Gab es vielleicht Probleme.

Beitrag von „apfelnico“ vom 20. Januar 2020, 14:05

Zitat von vit9696

PMC device has nothing to do to LPC bus, but is added to its scope for faster initialisation. If we add it to PCI0, where it normally exists, it will start in the end of PCI configuration, which is too late for NVRAM support.

Diesen "Trick" finde ich interessant. 😊