

Erledigt

Kernel Panik NVME (Samsung 970 EVO Plus)

Beitrag von „Stepping_Steve“ vom 19. März 2020, 18:51

Ich hatte in den letzten Tagen 2-3 x einen Absturz.

Nach dem Absturz arbeitet das System erst einmal ganz normal weiter, ist meistens am Anfang, dann friert der Bildschirm ein und ich bekomme den Fehlerbericht dann nach dem Neustart.

Kann das wegen der Samsung 970 EVO Plus sein? Die macht doch so weit ich weiß ein Problem mit Hackintoshs, da bin ich letztes Jahr schon bei meiner Erstinstallation gescheitert.

Ich habe mein MacOS auf einer anderen Platte installiert, auf der 970 EVP Plus ist Windows 10 drauf.

Ich hänge mal meine EFI mit dran, falls irgendjemand eine Idee hat, ob ich evtl. dort ändern kann.

*** Panic Report ***

```
panic(cpu 0 caller 0xfffff7f9cbd9b30): nvme: "Fatal error occurred. CSTS=0x1 US[1]=0x0
US[0]=0x99 VID/DID=0xa808144d
```

```
.
Revision=1B2QEXM7\n"@/BuildRoot/Library/Caches/com.apple.xbs/Sources/IONVMeFamily/IONVMeFamily-
387.270.1/IONVMeController.cpp:5334
```

FW

Backtrace (CPU 0), Frame : Return Address

```
0xfffff83e3d5ba10 : 0xfffff8019fb058d mach_kernel : _handle_debugger_trap + 0x47d
```

```
0xfffff83e3d5ba60 : 0xfffff801a0ec0f5 mach_kernel : _kdp_i386_trap + 0x155
```

```
0xfffff83e3d5baa0 : 0xfffff801a0dd82a mach_kernel : _kernel_trap + 0x50a
```

```
0xfffff83e3d5bb10 : 0xfffff8019f5d9d0 mach_kernel : _return_from_trap + 0xe0
```

```
0xfffff83e3d5bb30 : 0xfffff8019faffa7 mach_kernel : _panic_trap_to_debugger + 0x197
```

```
0xfffff83e3d5bc50 : 0xfffff8019fafdf3 mach_kernel : _panic + 0x63
```

```
0xfffff83e3d5bcc0 : 0xfffff7f9cbd9b30 com.apple.iokit.IONVMeFamily :
```

__ZN16IONVMeController13FatalHandlingEv + 0x10e

0xffffffff83e3d5be20 : 0xffffffff801a65ec27 mach_kernel :
__ZN18IOTimerEventSource15timeoutSignaledEPvS0_ + 0x87

0xffffffff83e3d5be90 : 0xffffffff801a65eb49 mach_kernel :
__ZN18IOTimerEventSource17timeoutAndReleaseEPvS0_ + 0x99

0xffffffff83e3d5bec0 : 0xffffffff8019fef7a5 mach_kernel : _thread_call_delayed_timer + 0xef5

0xffffffff83e3d5bf40 : 0xffffffff8019fef345 mach_kernel : _thread_call_delayed_timer + 0xa95

0xffffffff83e3d5bfa0 : 0xffffffff8019f5d0ce mach_kernel : _call_continuation + 0x2e

Kernel Extensions in backtrace:

com.apple.iokit.IONVMeFamily(2.1)[6B5003B4-7D0F-30D6-B6CD-7EFECA4995E3]@0xffffffff7f9cbcc000->0xffffffff7f9cc0bfff

dependency: com.apple.driver.AppleMobileFileIntegrity(1.0.5)[D22E4DCD-BCF5-389C-9BFB-84A953F6D5CD]@0xffffffff7f9b0ae000

dependency: com.apple.iokit.IOPCIFamily(2.9)[109B7A58-40CC-3FD4-BDAF-E7BA84B833C0]@0xffffffff7f9a895000

dependency: com.apple.driver.AppleEFINVRAM(2.1)[26909910-BA39-306B-A7D2-3EEC7CF9BD28]@0xffffffff7f9b1c9000

dependency: com.apple.iokit.IOStorageFamily(2.1)[AE1E0557-33D7-30A9-BC98-C442B799D4AA]@0xffffffff7f9aea4000

dependency: com.apple.iokit.IOReportFamily(47)[611D1E50-87AF-397D-B499-818E63E052B3]@0xffffffff7f9b8c4000

BSD process name corresponding to current thread: kernel_task

Boot args: dart=0 darkwake=no keepsyms=1 debug=0x100 agdpmo=pikera

Mac OS version:

18G3020

Beitrag von „KMBeatz“ vom 19. März 2020, 19:00

Soweit ich es noch im Kopf habe ist die Evo 970 sehr zickig. Da gibt es glaube ich eine Firmware womit macOS dann funktioniert.

Beitrag von „Romsky“ vom 19. März 2020, 19:13

Das betraf eigentlich nur die 970. Die 970 plus war davon eigentlich nicht betroffen... oder war es genau andersherum?

Beitrag von „CMMChris“ vom 19. März 2020, 19:17

Es war genau anders herum. Mich wundert ja dass er damit macOS überhaupt installiert bekommen hat. Normal geht ja das schon nicht. Firmware Update machen und dann sollte es laufen.

Beitrag von „Romsky“ vom 19. März 2020, 19:19

Ja, manchmal läuft eine Installation ohne Probleme durch.

Beitrag von „Stepping_Steve“ vom 19. März 2020, 19:25

Ich hatte in meinem Beitrag doch erwähnt, dass ich nicht auf die EVO 970 Plus installieren konnte, wegen dem bekannten Problem.

Nur jetzt nach der Neuinstallation der Grafikkarte kommt es eben zu diesem Fehler.

Ich schau mal, wo ich das Firmware Update herbekomme - ich arbeite schließlich bei Samsung



Beitrag von „Stepping_Steve“ vom 22. März 2020, 18:13

habe die neue Firmware installiert - falls es jemanden interessiert, diese ist hier zu finden:

<https://www.samsung.com/semico...isite/ssd/download/tools/>

https://s3.ap-northeast-2.amaz...970_EVO_Plus_2B2QEXM7.iso

Dazu gibt es eine Anleitung in Englisch, die ist aber selbsterklärend.

Vorher aber bitte eine Datensicherung der Platte vornehmen, damit nichts schief gehen kann.

Seit dem Update auch keinen Kernel-Panic mehr, scheint also geholfen zu haben.

Und nochmal zur Info: auf der Samsung EVO 970 Plus ist NICHT MacOS drauf, sondern Windows 10, aber Mojave lädt ja auch die Daten der Festplatte, deswegen kann ich den Kernel-Panic schon nachvollziehen. Warum das mit meiner alten Grafikkartenkonfiguration nicht aufgetreten ist, ist mir ein Rätsel, aber egal. Funktioniert jetzt alles bestens.

Thema kann von meiner Seite aus geschlossen werden



Beitrag von „al6042“ vom 22. März 2020, 18:17

<https://www.hackintosh-forum.de/forum/thread/47145-kernel-panik-nvme-samsung-970-evo-plus/>

Hervorragend und Glückwunsch zum Erfolg.

Geschlossen werden hier nur noch Threads die entweder gegen die Forenregeln verstoßen oder anderweitig unpassend ausarten.

Wenn der Vorgang abgeschlossen ist, kannst du das mit einem Klick beim "Erledigt"-Kästchen signalisieren.



Beitrag von „Romsky“ vom 22. März 2020, 22:36

Edit by al6042 -> Das Zitieren muss hier nicht sein. Ein einfaches Username hätte gereicht...

Das ist komisch. Die PM981 / 981a machen ja auch Probleme, aber nur wenn man darauf schreiben will. Lesen klappt einwandfrei.

Beitrag von „mhaeuser“ vom 22. März 2020, 22:44

NVMeFix hilft bei der PM981: <https://github.com/acidanthera/NVMeFix>

Beitrag von „itisme“ vom 24. November 2020, 21:11

[mhaeuser](#) Weißt Du evtl. auch, ob die NVMeFix-Kext (bei mir v1.0.4 mit OC 0.6.3) bei der PM991 ebenfalls helfen sollte?

Beitrag von „KMBeatz“ vom 24. November 2020, 22:40

PM981 ist definitiv nicht für macOS geeignet. Da helfen keine Kexte Scripts etc.

Finger weg