

Erledigt

Kein Ruhezustand nach Ruhezustand

Beitrag von „baerhund“ vom 15. Juli 2013, 19:06

Hallo!

Bei meinem Hack funktionierte Sleep nur manuell, also nicht nach einer bestimmten Zeit automatisch. In der org.chameleon.boot.plist hatte ich darkwake=0. Ich hab dann ein bisschen mit verschiedenen Zahlen bei darkwake rumprobiert, damit auch automatischer Sleep funktioniert. Mit darkwake=10 klappt das auch, allerdings funktioniert Sleep nun nicht mehr nachdem der PC schon einmal im Ruhezustand war. Weder automatisch noch manuell und auch nicht mehr mit darkwake=0.

Ich hoffe jemand kann mir weiter helfen

Beitrag von „DoeJohn“ vom 15. Juli 2013, 19:36

Hast du es schon mal mit darkwake=no probiert?

Beitrag von „baerhund“ vom 15. Juli 2013, 19:37

Ja dann ging sleep gar nicht mehr. Darkwake 1 und 8 hab ich auch schon versucht

Beitrag von „Ehemaliges Mitglied“ vom 15. Juli 2013, 20:29

Versuch mal hier das Tool: [Pleasesleep](#)

Beitrag von „baerhund“ vom 15. Juli 2013, 20:45

Danke aber auch damit funktioniert es leider nicht. Eben ist mir noch aufgefallen, dass nachdem der Hack schon im sleep war, er sich zwar nicht nochmal direkt schlafen legen lassen wollte, es aber ca. 30 Minuten nachdem ich auf Ruhezustand geglickt hatte getan hat. Ohne dass ich dann nochmal auf Ruhezustand geglickt habe

Ich verstehe nur nicht warum es vorher ging und jetzt mit dem gleichen Flag nicht mehr. Gibt es vielleicht irgendeinen Cache den man leeren kann oder etwas ähnliches?

Beitrag von „Griven“ vom 15. Juli 2013, 22:11

Vorab vielleicht mal zum Verständnis, der Flag Darkwake ist vom Typ Boolean kennt also nur yes oder no oder eben 1 oder 0. Des weiteren hat er nichts mit der Schlaffähigkeit Deines Hacks an sich zu tun sondern legt lediglich fest ob der Hack/Mac bei bestimmten Aufwachereignissen (Wake on Lan) quasi nur halb aufwacht sprich nur die Platten, Prozessor und Ram aufwachen die Grafik aber weiter schläft oder eben nicht. Anders gesagt bewirkt Darkwake = 0 lediglich, dass der MAC/HACK immer komplett aufwacht und Darkwake=1 eben das er bei bestimmten Aufwachereignissen zwar aufwacht der Monitor aber dunkel bleibt.

Dein Schlafproblem hat also nichts mit dem darkwake flag zu tun sondern muss irgendwo anders begründet liegen. Wenn der Hack beim ersten mal schlafen geht und im Anschluss nicht mehr liegt das meist daran, dass das Bios beim Wake in Teilen neu geschrieben wurde (CMOS Problematik) und dann eben bis zum nächsten Neustart nicht mehr korrekt funktioniert. Hier schafft eine gepackte AppleRTC.kext in den meisten Fällen Abhilfe. Du Findest sie unter anderem in Multibeast.

Beitrag von „baerhund“ vom 15. Juli 2013, 22:14

Erstmal ein großes Dankeschön für die ausführliche Erklärung. Ich versuchs gleich mal mit der rtc

Beitrag von „DoeJohn“ vom 15. Juli 2013, 22:17

Also bei mir ist das so, seitdem ich anstatt "darkwake=0". "darkwake=no" eingegeben habe, geht mein Häcki nun auch automatisch (nach einer vorgegebenen Zeit) in den Ruhezustand. Vorher bei "darkwake=0" konnte ich den Ruhezustand nur manuell herbeiführen!

Beitrag von „baerhund“ vom 15. Juli 2013, 22:50

Hatte ich oben falsch geschrieben: Bei darkwake=no funktioniert es doch, aber nicht besser als bei den anderen.

Die gepatchte AppleRTC hilft leider nicht. Immer noch das gleiche Problem, habs mit darkwake=0 und darkwake=No probiert

Beitrag von „TuRock“ vom 15. Juli 2013, 22:58

Lösche SleepEnabler.kext (falls vorhanden) und AppleIntelCPUPowerManagement.kext aus S/L/E Ordner, zugriffsrechte reparieren !

darkwake=0 in org.chameleon.boot.plist eintragen !

Neustarten !

So sollte es bei dir funktionieren !

Beitrag von „baerhund“ vom 15. Juli 2013, 23:01

Sorry für die vielleicht dumme Frage aber kann es ohne AppleIntelCPUPowerManagement.kext nicht zu einer Kernel Panic kommen? Oder läuft das alles über die SSDT?

Beitrag von „TuRock“ vom 15. Juli 2013, 23:06

Übertaktet !

Das könnte auch der Grund sein warum Sleep bei dir nicht richtig funktioniert !

Beitrag von „DoeJohn“ vom 15. Juli 2013, 23:09

Also bei mir funktioniert "Sleep", nachdem ich die "NullCPUPowermanagement.kext" gelöscht hatte und in die org.chameleon.boot.plist "darkwake=no" eingegeben hatte! Ich habe eine DSDT für mein Board und habe ein übertaktetes System und keine "sleepenabler.kext"!

Beitrag von „baerhund“ vom 15. Juli 2013, 23:17

Nicht wirklich übertaktet ich habe aber den Turbo bei allen Kernen fest auf 39 gesetzt (also vielleicht doch schon übertaktet?) damit der Multiplikator auch wirklich bis dahin hoch geht

Ok du hattest recht ich hab jetzt mal alles auf Auto gestellt und es geht wieder. Wie könnte es denn trotz dem kleinen "overclock" klappen?

Beitrag von „Dr. Ukeman“ vom 16. Juli 2013, 22:18

Warum willst du denn übertakten? So wie du es gesetzt hattest wirst du es nicht merken und holst dir mehr Probleme ins Haus. Evtl hat es einen Sinn warum alle Kerne niedriger getaktet werden als einer.

Gesendet vom iPhone via
Tapatalk

Beitrag von „baerhund“ vom 16. Juli 2013, 22:21

Ja hast du wohl recht ich dachte vorher nur dass ohne übertakten nicht bis 3,9 GHz hochgetakten wird. Der Terminal Befehl von turock aus nem anderen thread zeigt mir aber an dass es doch so ist. Ist also gut auf auto

Beitrag von „mull7965“ vom 29. Juli 2013, 20:24

[Zitat von Griven](#)

Vorab vielleicht mal zum Verständnis, der Flag Darkwake ist vom Typ Boolean kennt also nur yes oder no oder eben 1 oder 0. Des weiteren hat er nichts mit der Schläffähigkeit Deines Hacks an sich zu tun sondern legt lediglich fest ob der Hack/Mac bei bestimmten Aufwachereignissen (Wake on Lan) quasi nur halb aufwacht sprich nur die Platten, Prozessor und Ram aufwachen die Grafik aber weiter schläft oder eben nicht. Anders gesagt bewirkt Darkwake = 0 lediglich, dass der MAC/HACK immer komplett aufwacht und Darkwake=1 eben das er bei bestimmten Aufwachereignissen zwar aufwacht der Monitor aber dunkel bleibt.

Das stimmt leider nur halb. Auch wenn es noch so häufig geschrieben wird, Darkwake ist kein Boolean. Man kann das glücklicherweise selbst nachsehen, direkt im Sourcecode des Kernels von OS X. (Für alle denen das nicht bekannt ist, Apple macht schon seit langer Zeit den Kernel und einige andere "low-level" Bestandteile von OS X unter einer Open Source Lizenz verfügbare.) Eine gute Beschreibung der Behandlung des Darkwake Flags findet sich in einem Thread in einem anderen Hackintosh Forum das hier nicht erwähnt werden darf. Wer's mit Google/Bing/etc nicht findet, oder wer lieber Deutsch liest, für den folgt eine kurze

Zusammenfassung.

Der Bootloader, also zum Beispiel Chameleon, hat mit Darkwake nichts am Hut und schickt das Flag mitsamt Wert direkt an den Kernel. Dieser liest das Flag mit Hilfe der Funktion **PE_parse_boot_argn** aus (Zeile 864 in [IOPMrootDomain.cpp](#)).

Code

```
1. PE_parse_boot_argn("darkwake", &gDarkWakeFlags, sizeof(gDarkWakeFlags));
```

Geschrieben wird der Wert in eine Variable die eine Bitmaske mit verschiedenen Flags enthält:

Code

```
1. // gDarkWakeFlags
2. enum {
3. kDarkWakeFlagHIDTickleEarly = 0x01, // hid tickle before gfx suppression
4. kDarkWakeFlagHIDTickleLate = 0x02, // hid tickle after gfx suppression
5. kDarkWakeFlagHIDTickleNone = 0x03, // hid tickle is not posted
6. kDarkWakeFlagHIDTickleMask = 0x03,
7. kDarkWakeFlagIgnoreDiskIOInDark = 0x04, // ignore disk idle in DW
8. kDarkWakeFlagIgnoreDiskIOAlways = 0x08, // always ignore disk idle
9. kDarkWakeFlagIgnoreDiskIOMask = 0x0C,
10. kDarkWakeFlagAlarmsIsDark = 0x0100
11. };
```

Alles anzeigen

Interessanterweise hat **PE_parse_boot_argn** überhaupt keine spezielle Behandlung der Zeichenketten *yes* bzw. *no*, sie kennt lediglich die Typen String und Number (ab Zeile 104 in [bootargs.c](#)). Warum funktioniert dann *no* für einige Leute?

Weil es eben nur Number und String gibt, wird *no* als String angesehen und die Bytes für "n" und "o" einfach in die Variable vom Typ uint32 geschrieben. Das entspricht dann der Zahl 28526 ("n" = 110, "o" = 111, $111*256+110 = 28526$). Man kann sich jetzt selbst überlegen welche der o.g. Flags damit getroffen werden.

Hoffe das hilft zur Klärung.