

Selbst kompilieren - bringt's irgendwas?

Beitrag von „QCO“ vom 25. Mai 2020, 22:11

Guten Abend, ich hab zur Abwechslung mal keine "ich brauch Hilfe"-Frage, sondern mal ein richtiges Diskussionsthema...

Gibt es irgendwelche Vorteile OpenCore oder auch kexts selbst zu kompilieren, wenn ich auch Binaries downloaden kann?

Ich bin jetzt selbst kein (aktiver) Software-Entwickler und kenne das bisher eigentlich nur aus dem Linux- und BSD-Umfeld, wenn man aus rechtlichen oder praktischen oder manchmal auch ideologischen Gründen kein Binary bekommt.

Mit github, GPL, homebrew und Tools wie Hackintool ist man ja inzwischen in der komfortablen Lage, dass man für fast alles Quellcode downloaden und selbst bauen könnte. Und das ist auch sehr cool, wenn man mal wegen irgendeines Fixes vor Release selbst bauen kann oder irgendein obskures Tool kein signiertes Release hat usw.

Wenn ich aber gar kein Problem habe: Bringt's mir irgendwas selbst zu bauen?

Beitrag von „kuckkuck“ vom 25. Mai 2020, 23:51

Solange du darauf vertraust, dass die bereitgestellte Release Binary dem Source Code entspricht, sehe ich da keinen Vorteil... Erst recht, wenn du das Ganze mit unverändertem Makefile o.ä. kompilierst. Du weißt halt sicher, was dir danach vorliegt, anders als beim Download durch zB Third Party Tools.

Andere Sache ist es natürlich, wenn sich die gezogene Source von der releaste Binary unterscheidet, oder man selber Veränderungen einbauen/testen will.

Beitrag von „mhaeuser“ vom 26. Mai 2020, 00:33

Hier eher umgekehrt, da wir nur offizielle Builds unterstützen.

Beitrag von „QCO“ vom 6. Juni 2020, 23:15

Okay, danke für die Antworten.

Was ist mit der uralten und wahrscheinlich inzwischen völlig überholten Weisheit, dass selbst kompilieren den Vorteil bringt, auf die eigene Hardware entsprechend zugeschnitten zu sein?

Beitrag von „griven“ vom 6. Juni 2020, 23:30

Die ist, wie Du selbst sagst, uralt und völlig überholt 😊

Nee im Ernst einen Vorteil hat es nur wenn man "Bleeding edge" sein möchte oder eben selbst anpassen möchte/muss ansonsten kann man getrost die bereitgestellten Binaries verwenden solange sie von den Projekten selbst stammen (GitHub zum Beispiel) oder aus verifizierten Quellen kommen. Ich halte es so das ich zum Beispiel bei OpenCore zum testen und experimentieren jeweils selbst baue einfach weil ich dann den jeweils zu Zeitpunkt des Bauens aktuellen Entwicklungsstand habe für produktive Systeme verwende ich die jeweiligen Release Versionen. Bei den Extensions halte ich es ähnlich gibt es Änderungen die für mich relevant sind die aber gerade erst adressiert wurden und somit noch in keinem Release vorhanden sind wird selbst gebaut ansonsten auch hier die Release Version verwendet.

Beitrag von „QCO“ vom 6. Juni 2020, 23:50

Nochmal danke.

Ich bin da halt gerade hart am lernen, was ich alles in der IT nicht mitbekommen habe.

Man wird älter und verpasst so einiges. So in den frühen 2000ern war ich bei allem

Neuerungen direkt dabei, aber dann kam ein Studium, Industrie usw...

Jetzt versuch ich mal an verschiedensten Fronten wieder auf den Stand der Technik zu kommen.

Beitrag von „b33st“ vom 6. Juni 2020, 23:54

Sehe ich jetzt keine Vorteile, solange man nicht die Sources/Makefiles/Configs u.ä. für seine eigene Zwecke/Bedürfnisse dementsprechend anpasst.

Der Quelle muss man jendfalls weiterhin vertrauen können. Sofern man "Bedenken" hat, müsste man eh Zeile für Zeile durchgehen und sich den Code genauestens anschauen, bevor man die kompilieren Binaries ausführt.

Überholt finde ich es jedoch nicht. Nicht zu selten ist es noch bei i-welchen UI-Clients zu finden (zumindest in macOS/Linux), wie z.B:

Code

1. INCLUDEDIR += /usr/include/qt4/Qt
2. INCLUDEDIR += /usr/include/qt4/QtGui

Oder halt ggf. **qt5** wenn man's braucht usw.