# OpenCore-Version ohne Booten bestimmen

## Beitrag von „HAI“ vom 20. August 2021, 02:28

Ist ziemlich lästig, wenn man erst booten muß, bei einer anderen EFI, um die Version zu bestimmen und auch ziemlich fehleranfällig.

Bei einer "übergebenen" EFI hier im Forum kann dann sehr schnell die Version ermittelt werden.

**Hier ist eine 100% Methode.**

Von jeder Version **hier genau einmal** den Hash zentral ermitteln (sha-256). Von der Datei "opencore.efi" im EFI-Verzeichnis.

Hier schon einmal die 7er plus die historische 0.0.1.

Code

1. CA70EE4DE2B7365795A9FF25E9686D134B6C2952457F02EC2A98D85D3A4A2313 - 0.0.1

Code

1. 5984E2A40124CCCCE2871D755188BBD21A0CEE0A3D15A2EE9E46936D5086D36F - 0.7.2

Code

1. EBB13921259DE09C71927B6D8705B521521A5A3C2FFF0E1C28E552C2AFD7BF39 - 0.7.1

Code

1. CA33FD96EA1C088A54E17123F4A3D42B8D68E4DFFA4A138B070B93AE8774AAC1 - 0.7.0

Diese Methode funktioniert auch, wenn sich dort das Team entschließt, bei einer Version keine Änderung der zentralen "opencore.efi" vorzunehmen. Genau eines wird sich dann ändern, das ist die Versionsnummer. Diese kleine Änderung bewirkt eine starke Veränderung des hashes,

das ist eine Eigenschaft einer solchen Prüfsumme. Vorne und hinten 3 Stellen zur Prüfung reichen aus. Man muß nicht alle 64 vergleichen.

Damit bestimmt man die Versionsnummer, wie im Hackintool beim Booten. An einer funktionierenden EFI bleibt selbstverständlich noch eine Menge zu tun.

Hier noch die zentrale Datei für die Versionsnummer zur Veranschaulichung.

```
#include <Protocol/OcBootstrap.h>

/**
  OpenCore version reported to log and NVRAM.
  OPEN_CORE_VERSION must follow X.Y.Z format, where X.Y.Z are single digits.
**/
#define OPEN_CORE_VERSION         "0.7.3"

/**
  OpenCore build type reported to log and NVRAM.
**/
#if defined (OC_TARGET_RELEASE)
#define OPEN_CORE_TARGET          "REL" ///< Release.
#elif defined (OC_TARGET_DEBUG)
#define OPEN_CORE_TARGET          "DBG" ///< Debug with compiler optimisations.
#elif defined (OC_TARGET_NOOPT)
#define OPEN_CORE_TARGET          "NPT" ///< Debug with no compiler optimisations.
#else
#error "Unknown target definition"
#endif

```

Sascha_77 möchte noch bitte Deinen Stempel drauf geben. Kann das Hackintool zwischen "DBG" und "REL" unterscheiden (Log wahrscheinlich schon)? Debug hat eine andere Prüfsumme, dann hätten wir genau zwei für ein Release.

Für die Bestimmung dieser Prüfsumme benötigt man etwa 17 Sekunden.

Das ist das Drag and Drop Tool (GUI). Für die commandliner "shasum".

https://www.quickhash-gui.org/

(Bei wichtiger Software, wie z.B.: Linux, wird das ebenfalls gemacht (xxxx.iso). Zusätzlich gibt es dann eine digitale Unterschrift. Die benötigen wir nicht.)

(Man kann natürlich alle Versionen downloaden, d.h. dann auch die Debugs. Dann Zeitstempel und Größe vergleichen. Die Prüfsummenlösung ist einfacher und schneller.)