

Ideen zur Erstellung eines bootfähigen Klons und Vergrößerung der EFI-Partition unter macOS Sequoia

Beitrag von „DerBeste“ vom 2. Dezember 2024, 13:43

An die Mods: Bitte diesen Beitrag nicht löschen.

Ich möchte euch vorab eine Frage stellen und bei Bedarf meine Idee und Erfahrungswerte hier detailliert beschreiben.

Hallo Community,

Wie erstellt ihr einen 1:1 bootfähigen Klon eurer macOS Sequoia-Partition?

Und wie geht ihr vor, um die Standard-EFI-Partition (in der Regel etwa 200 MB groß) auf ca. 800 MB unter macOS zu vergrößern, damit sie für das Booten geeignet ist?

Ich freue mich auf eure Tipps und Erfahrungen!

Edit: 7.12.2024

Hier die Anleitung:

Um unter macOS eine EFI-Partition zu erstellen oder zu vergrößern, muss man zuerst die [SIP \(System Integrity Protection\)](#) deaktivieren. Dazu muss der Hackintosh oder Mac im Wiederherstellungsmodus neu gestartet werden. Nach dem Starten im Wiederherstellungsmodus öffnet man das Terminal und gibt folgenden Befehl ein:

Code

1. csrutil disable

Danach wird das Terminal-Fenster mit dem Befehl **exit** geschlossen und das System neu gestartet. Nach dem Neustart des Systems geben wir zur Kontrolle den folgenden Befehl ein:

Code

1. diskutil list

Wenn "**disabled**" angezeigt wird, können wir den Prozess fortführen. Jetzt geben wir erneut „**diskutil list**“ im Terminal ein, um alle Partitionen auf dem PC oder Mac aufzulisten. Die betreffende Festplatte (in meinem Fall ist zum Beispiel **disk0**) muss in der Auflistung gefunden werden.

Wichtig: Eine standardmäßig vom System erstellte EFI-Partition kann unter Umständen nicht groß genug sein, um alle benötigten Treiber und Daten aufzunehmen. Falls eine bestehende EFI-Partition, die vom System erstellt wurde und den EFI-Ordner enthält, vorliegt, sollte unbedingt ein Backup davon gemacht werden, um unnötige Datenverluste zu vermeiden.

Achtet bitte genau darauf, welche Partition ihr auswählt! Vermeidet es, versehentlich eine andere Partition zu löschen, da dies zu irreversiblen Datenverlusten führen kann.

Um zwei Partitionen zu erstellen (eine für EFI und eine für den verbleibenden Speicher), solltest du den Befehl folgendermaßen anpassen:

Schritt 1: Partitionieren mit diskutil (Beispiel bei einer Festplatte mit einer Größe von 80 GB)

Versuche anhand diesem Befehl, um das gewünschte Ergebnis zu erreichen:

Die zu erstellende Größe des EFI Ordners zB. **EFI 800M** abzüglich eurer tatsächliche Festplattengröße wird in "**Free Space**" **%noformat% 79.2** eingetragen, siehe Beispiel Code den ihr in das Terminal Fenster einfügt und mit Enter Taste betätigt und mit eurem Passwort bestätigt :

Code

```
1. sudo diskutil partitionDisk /dev/disk0 2 GPT "MS-DOS FAT32" EFI 800M "Free Space"  
%noformat% 79.2G
```

Die Ziffer **2** in den oberen Code : Gibt an, dass zwei Partitionen erstellt werden.

"MS-DOS FAT32" EFI 800M: Erstellt eine 800 MB große EFI-Partition im FAT32-Format.

"Free Space" %noformat% 79.2G: Erstellt die restliche Partition als freien Speicher mit einer Größe von etwa 79,2 GB. (abhängig davon wie groß tatsächlich eure Festplatte ist.)

Schritt 2: Überprüfe das Resultat

Nach dem Partitionieren können Sie den Status der Festplatte mit dem folgenden Befehl überprüfen:

Code

1. `diskutil list`

Dieser Befehl sollte die neue EFI-Partition und die freie Speicherpartition auflisten.

Wenn diese Methode immer noch nicht funktioniert, könnte es hilfreich sein, die Festplatte im Wiederherstellungsmodus zu formatieren, um sicherzustellen, dass keine Systembeschränkungen wie [SIP](#) oder blockierte Partitionierungsoperationen vorhanden sind.

Es scheint, dass nun die EFI-Partition erfolgreich mit einer Größe von 800 MB erstellt wurde, aber du möchtest nun die ursprüngliche EFI-Partition (**disk0s1**) löschen.

Um die kleine EFI-Partition (**disk0s1**) zu löschen, folge den unteren Anweisung:

Schritt 1: Löschen der kleinere EFI-Partition

Verwende „**diskutil**“ zum Löschen der Partition:

Code

1. `sudo diskutil eraseVolume "Free Space" %noformat% /dev/disk0s1`

Schritt 2: Überprüfe die Partitionstabelle

Nach dem Löschen solltest du den Status der Festplatte überprüfen, um sicherzustellen, dass die kleine EFI-Partition entfernt wurde und nur die neue EFI-Partition und die große freie Partition vorhanden sind:

Code

1. diskutil list

Damit sollten Sie die Festplatte in ihrem gewünschten Zustand haben. Lassen Sie mich wissen, ob es funktioniert hat!

Vorher:

Code

1. mein@iMac-Pro-von-mein ~ % diskutil list
2. /dev/disk0 (internal, physical):
3. #: TYPE NAME SIZE IDENTIFIER
4. 0: GUID_partition_scheme *80.0 GB disk0
5. 1: EFI EFI 209.7 MB disk0s1
6. 2: Microsoft Basic Data EFI 79.8 GB disk0s2

Nach Step 1:

Code

1. mein@iMac-Pro-von-mein ~ % diskutil list
2. /dev/disk0 (internal, physical):
3. #: TYPE NAME SIZE IDENTIFIER
4. 0: GUID_partition_scheme *80.0 GB disk0
5. 1: EFI EFI 209.7 MB disk0s1
6. 2: Microsoft Basic Data EFI 799.0 MB disk0s2
7. (free space) 79.0 GB -

Nach Step 2:

Code

1. mein@iMac-Pro-von-mein ~ % diskutil list
2. /dev/disk0 (internal, physical):
3. #: TYPE NAME SIZE IDENTIFIER
4. 0: GUID_partition_scheme *80.0 GB disk0
5. 1: Microsoft Basic Data EFI 799.0 MB disk0s2
6. (free space) 79.0 GB -

Da die **disk0** nicht formatiert wurde muss dieser Schritt nun gemacht werden.

Anleitung: Freien Speicher auf einer Festplatte in APFS formatieren:

Hier ist eine Schritt-für-Schritt-Anleitung, wie wir den freien Speicherplatz auf einer Festplatte in **APFS** formatiert haben, ohne die vorhandene **EFI-Partition** zu löschen:

Schritt 1: Vorbereitungen und Übersicht der Festplatte

Zunächst wird geprüft, welche Partitionen und wie viel freier Speicherplatz auf der Festplatte vorhanden sind. Dafür verwenden wir den Befehl:

Code

1. `diskutil list`

Beispielausgabe:

Code

1. `/dev/disk0 (internal, physical):`
2. `#: TYPE NAME SIZE IDENTIFIER`
3. `0: GUID_partition_scheme *80.0 GB disk0`
4. `1: Microsoft Basic Data EFI 799.0 MB disk0s1`
5. `(free space) 79.0 GB -`

- `disk0`: Die Ziel-Festplatte, die partitioniert werden soll.
- `disk0s1`: Die vorhandene EFI-Partition, die erhalten bleiben soll.
- Freier Speicherplatz: 79 GB, den wir in APFS formatieren wollen.

Schritt 2: Prozesse finden, die die Festplatte blockieren

Um sicherzustellen, dass keine Prozesse auf die EFI-Partition zugreifen (was die Partitionierung blockieren könnte), verwenden wir:

Code

1. `sudo lsof | grep disk0`

Beispielausgabe:

Code

1. fskitd 286 root 3u CHR 1,1 0t0 623 /dev/rdisk0s1
2. com.apple 389 root 3u CHR 1,1 0t0 623 /dev/rdisk0s1

In diesem Fall greifen zwei Prozesse auf **/dev/rdisk0s1** zu: fskitd und com.apple.

Schritt 3: Blockierende Prozesse beenden

Die blockierenden Prozesse werden mit folgendem Befehl beendet:

Code

1. sudo kill -9 <Prozess-ID>

Beispiel für die obigen Prozesse:

Code

1. sudo kill -9 286 389

Schritt 4: EFI-Partition unmounten

Die EFI-Partition wird unmounted, damit der freie Speicherbereich bearbeitet werden kann:

Code

1. sudo diskutil unmount /dev/disk0s1

Erfolgsmeldung:

Code

1. Volume Untitled on disk0s1 unmounted

Schritt 5: Freien Speicherplatz in APFS formatieren

Jetzt wird der freie Speicherplatz in **APFS** formatiert. Dabei bleibt die EFI-Partition erhalten:

Code

1. sudo diskutil partitionDisk /dev/disk0 GPT \ "APFS" "NeuesVolumeName" 100%

- GPT: GUID-Partitionstabellenschema.
- APFS: Dateisystem für die neue Partition.

- NeuesVolumeName: Name der neuen Partition.
- 100%: Verwendet den gesamten freien Speicherplatz für die neue Partition

Schritt 6: Ergebnis überprüfen

Nach der erfolgreichen Partitionierung prüfen wir das Ergebnis mit:

Code

1. diskutil list

Beispielausgabe:

Code

1. /dev/disk0 (internal, physical):
2. #: TYPE NAME SIZE IDENTIFIER
3. 0: GUID_partition_scheme *80.0 GB disk0
4. 1: Microsoft Basic Data EFI 799.0 MB disk0s1
5. 2: APFS NeuesVolumeName 79.2 GB disk0s2

Schritt 7: EFI-Partition wieder mounten

Falls erforderlich, wird die EFI-Partition wieder gemountet:

Code

1. sudo diskutil mount /dev/disk0s1

Viel Freude!