

# Script dmgg.py unter Linux geht nicht - doch!

**Beitrag von „Altemirabelle“ vom 2. Oktober 2023, 14:38**

Ich hab da ein Problem mit Linux, für mich ist das schwarze Magie.

Also hab mir ein USB Stick mit Arch Linux nach Anleitung gemacht, drauf ein Script dmgg.py kopiert.

Das Ding überprüft den VRAM einer Grafikkarte, sucht nach defekten chips.

Nachdem ich von USB gebootet habe (intel Karte macht die Ausgabe), gebe ich:

```
fdisk -l >>> mein USB ist also sdc, >> sdc1
```

```
mount /dev/sdc1 /mnt >>> wird gemacht
```

```
cd /run/archiso/bootmnt >>> da habe ich das Problem, erhalte: no such file or directory. Ja aber warum, auf YT geht das. Leider keine Antwort von dem Ersteller der Anleitung.
```

Wenn ich `cd /run/archiso/` eingebe erhalte ich: `airootfs copytoram cowspace`, oder so ähnlich. Ich will halt nur dmgg.py ausführen.

Kann jemand helfen?

---

**Beitrag von „apfel-baum“ vom 2. Oktober 2023, 14:52**

hm, arch .. schonmal `./dmgg.py` + enter, ausprobiert, natürlich ohne die anführungszeichen?

lg 😊

edit-

kannst du bitte mal den gitlink oder link zum script mitteilen?

danke 😊

python sollte natürlich in der live iso vorhanden sein, ansonsten mußt du dich wohl mit pacman herumärgern

du kannst auch ein anderes livelinux via ventoy unter win als iso auf nen usbstick kopieren, z.b. bohdi welches auf debian basiert, oder ein anderes was dir gefällt, und nicht nur linux, sondern -fast- alles isos außer osx soweit ich weiß und diese vom stick starten/bootauswahl

--

edit 2-

ggf. mußt du du das script auch als ausführbar setzen

---

### Beitrag von „Altemirabelle“ vom 2. Oktober 2023, 15:34

Ich komme gar nicht zu dem python, weil ich nicht zu dem Verzeichnis komme, ich kenne mich nicht aus, aber ich denke die Version von Arch Linux ist neuer und "bootmnt" nicht versteht ????

Zu dem script:

---

### Beitrag von „apfel-baum“ vom 3. Oktober 2023, 18:34

ist das deine quelle? [https://repair.wiki/w/AMD\\_Memory\\_Testing\\_Guide](https://repair.wiki/w/AMD_Memory_Testing_Guide)

die nehmen tatsächlich ein arch oder arch-derivat . du mußt zum ausführen de scriptes **root** sein (der darf da alles) welches du normalerweise mittels "**sudo su**" ermöglichst, desweiteren wie unter nem \*nux üblich führst du dort ein program mittels **./** aus. damit du das machen kannst, mußt du entweder direkt in dessen verzeichnis, oder den ausführbefehl entsprechend anpassen, das ist selbst im osx-terminal der fall. also zuerst ins verszeichnis wo das entpackte python script drin ist, danach root werden, im anschluß mittels **./dmgg.py** (im beispiel dann

noch c0000000 20 8 hinzufügen, das script braucht wohl zusätzlich nähere parameter/angaben) und los gehts 😊

entweder passiert jetzt nix, oder es gibt eine ausgabe von fehlern, oder das script läuft -wie es soll- und der ram der grafikarte wird getestet

lg 😊

der seite/wiki entnommen siehe spoilerinhalt

testing memory on a card that does not output a picture

---

## Beitrag von „Altemirabelle“ vom 3. Oktober 2023, 19:34

Hab inzwischen den Fehler gefunden.

Was das ganze macht, sagt ob die AMD karte erkannt ist und sagt ob die DDR6 RAMs defekt sind, und welche.

Meine Parameter für RX570 waren 50000000 10 4. Und PowerColor RX 5700 Red Dragon hat 50000000 10 8.

So sieht das aus:



```
INFO: AMD GPU card: 0000000000000000 (PCI) 0x0000000000000000 (0000000000000000)
Info: 8 GB GDDR6 controller: Advanced Micro Devices, Inc. [AMD/ATI] Renoir [Radeon RX 5700M/5700 XT / 5700S/5700 273 1000 0000]
Subsystem: Tol Corporation / PowerColor Device 2299
Flags: Non-accelerated (not detect), Legacy A, SR, ILL
Memory at 00000000 (00-001, 00-001, 00-001, 00-001) (size=2GB)
Memory at 00000000 (00-001, 00-001, 00-001, 00-001) (size=2GB)
PCI bus: 00:00:00 (size=2GB)
Memory at 00000000 (00-001, 00-001, 00-001, 00-001) (size=2GB)
Information from PCI controller: [size=2GB]
Capabilities: [00] Vendor Specific Information: Len=00 CTD
Capabilities: [01] Power Management: version 3
Capabilities: [02] Express Legacy Endpoint: MSI 00
Capabilities: [03] Vendor Specific PCI/Express Specs
Capabilities: [04] MSI: Enable Count=1/1 Enable=0
Capabilities: [05] Vendor Specific Information: Len=00 Rev=00 CTD
Capabilities: [06] Advanced Error Reporting
Capabilities: [07] Physical Resizable BAR
Capabilities: [08] Power Budgeting CTD
Capabilities: [09] Secondary PCI Express
Capabilities: [0A] System Control Services
Capabilities: [0B] Address Translation Service (ATS)
Capabilities: [0C] Page Request Indication (PRI)
Capabilities: [0D] Process Address Space ID (PASID)
Capabilities: [0E] Latency Tolerance Reporting
Capabilities: [0F] Data Link Feature CTD
Capabilities: [10] Physical Layer 10.8 GT/s CTD
Capabilities: [11] Lane Margining at the Receiver CTD
Kernel driver: to use: amdgpu
Kernel module: amdgpu

Possible GPU address: 0000000000000000

number of chips to test: 8
This tool is working to detect bad chips, learning if can give wrong faulty chip number: why the amount of faulty chips will be good
want the chips number: checkbase from right to left with bits new gap.

No faulty chips found

number of faulty chips: 0
total bytes tested: 0x00000000
total errors count: 0 - every 0x00000000 00: 0x00000000
```