



[Erfahrungsbericht] Virtualisierung mit PCI-Passthrough auf Manjaro Linux

Arjab · 17. April 2020

1 2 3 Weiter ▶

**Arjab**

Lt. Junior Grade

Dabei seit: Feb. 2013
Beiträge: 416

17. April 2020

#1

Moin,

falls das nicht das richtige Subforum für einen Erfahrungsbericht ist, kann der Thread auch gerne verschoben werden. Thematisch passen das "Linux" oder "Virtualisierung"-Unterforum ebenso. Da es aber vor allem um die verwendete Hardware geht, dachte ich, dass ich hier am *richtigsten* bin.

Hier ein kleines Inhaltsverzeichnis:

- [Einleitung](#)
- [Update #1 - Einrichtung und SSD-Passthrough](#)
- [Update #2 - Neuinstallation und Fehlercode 43](#)
- [Update #3 - AMD Reset Bug und Benchmarks](#)
- [Update #4 - Boot-GPU einrichten](#)
- [Update #5 - Virt-Manager Speicherpools und Ryzen-Temperatur](#)
- [Update #6 - Linux-Kernel patchen und kompilieren](#)
- [Update #7 - Audio-Passthrough mit PulseAudio](#)
- [Update #8 - Bluescreen und Korrektur](#)
- [Update #9 - Kernel patchen und Grafikkartentreiber](#)
- [Update #10 - Danke für die Blumen](#)
- [Update #11 - Alles ist anders!](#)

Einleitung:

Nachdem mein aktuelles System bereits mit [Manjaro Linux](#) lief und ich mit [KVM/Qemu](#) und [libvirt](#) meine Grafikkarte via [PCI-Passthrough](#) an eine virtuelle Maschine mit Windows 10 durchgereicht habe, hab' ich besagtes System einer Aufrüstung unterzogen, wie ihr [hier](#) nachvollziehen könnt. Jetzt ist der große *Spaß*, alles wieder neu einrichten zu *dürfen*. Also Manjaro Linux auf neuer Hardware installieren, aber vor allem PCI-Passthrough und eine virtuelle Maschine (VM) einrichten. Dabei gibt es ohnehin mehrere Fallstricke, aber vor allem kann es konkret zu folgenden Problemen kommen:

1. Meine alte Grafikkarte ([MSI GTX 970 4G](#)) soll als GPU für das Linux-Hostsystem verwendet werden. Da die neue Grafikkarte ([MSI RX 5700 Evoke](#)) für das Windows 10-Gastsystem verwendet werden soll, muss diese - für die beste Performance - in den ersten PCIe-Slot. Bei meinem neuen Mainboard ([MSI B450 Tomahawk Max](#)) kann man im BIOS jedoch nicht einstellen, von welcher GPU gebootet werden soll. Dafür gibt es zwar Fixes und Workarounds, aber ob's wirklich klappt, weiß ich nicht.
2. Die aktuellen GPUs von AMD haben scheinbar einen Reset-Bug und die Initialisierung der Grafikkarte des Gastsystems nach einem Neustart dessen stürzt ab - oder so. Es gibt auf jeden Fall auch dafür einen Kernel-Patch und ich bin gespannt, ob der funktioniert oder das Problem überhaupt auftritt.

Gerade warte ich noch auf eine Teilbestellung mit [Netzteil](#) und [CPU-Kühler](#), hab' aber aus Ungeduld und Langeweile schon mal Prozessor, Arbeitsspeicher und Mainboard zusammengeworfen und tippe diesen Beitrag auf einem frisch installierten Manjaro Linux. Mit nur einer GPU im System bootet alles ganz problemlos, obwohl die GPU im vierten PCIe-Slot steckt. Mal sehen, ob sich das ändert, sobald ich die RX 5700 dazustecke.



Ich werd' hier in unregelmäßigen Abständen berichten, wie sich der weitere Prozess gestaltet. Fragt mich gerne nach spezifischen Dingen und ich hoffe, auf alles eine Antwort zu wissen. Ansonsten hoffe ich, dass hier einige interessierte Leute mitlesen und meine Bestellung bald ankommt. 😊

[Nach oben ↑](#)**Update #1 (23. April 2020):**

So, gestern kam das Paket mit Netzteil und CPU-Kühler an und ich hab' den ganzen Spaß mal zusammengeworfen. Falls sich jemand dafür interessiert, wie die Kiste von innen aussieht, kann ich gerne mal 'n Foto knipsen, wenn später die Sonne in mein Zimmerchen scheint. Immerhin ist die CPU jetzt im Idle nur noch bei ~ 35° bei offenem Fenster und 18° Außentemperatur. Aber genug davon, der PC bootet problemlos und wie erwartet darf ich mein Festplatten-Passwort auf dem



anzeigt. So weit, so gut!

Ansonsten hab' ich IOMMU bzw. Virtualisierung in den sehr versteckten CPU-Einstellungen meines Mainboards aktiviert, den entsprechenden Kernel-Parameter gesetzt, damit IOMMU auch Software-seitig aktiviert ist, die IDs von GPU und dazugehörigem Audio-Controller ebenfalls als Kernel-Parameter gesetzt und die entsprechenden Module zum Inframts hinzugefügt. Ich erspare euch mal die Details, wen es interessiert kann die einzelnen Schritte in der [Arch Wiki](#) nachlesen.

Tja und nach einem Reboot – ist alles beim Alten, das System bootet mit der RX 5700 und wechselt dann auf die GTX 970. Aber immerhin zeigen mir `dmesg` und `lspci` an, dass beide Geräte erfolgreich vom vfio-Treiber gebunden wurden.



Ich hab' dann mal diese [Anleitung](#) zu Rate gezogen, die mir [@0-8-15 User](#) in dem [Kaufberatungs-Thread](#) verlinkt hat, aber irgendwie hat auch das noch keine Früchte getragen. Ich werd' erstmal die VM einrichten, so wild ist es jetzt nicht, beim Booten kurz das Input des Monitors zu wechseln.

Noch 'ne Notiz am Rande, ich hab' mal Folding@Home angeschmissen und bei ~ 85 % CPU-Auslastung ist die CPU bei maximal 80° C.

Also, *virt-manager* gestartet, ein bisschen rumgebastelt, um meine zweite SSD direkt als Speichergerät zu Verfügung zu stellen (siehe unten), richtige OVMF-Firmware ausgewählt, CPU-"Netzstrutur" und "host-passthrough" manuell eingetragen, 12 von 16 GB Arbeitsspeicher hinzugefügt, als erste Boot-Option ein Windows 10 ISO, als zweite die VirtIO-Treiber und schließlich die SSD ergänzt, Netzwerk erstmal idiotensicher auch als komplettes Passthrough, weil ich vergessen habe NAT einzurichten, Maus und Tastatur ebenfalls (zur Not kann hab' ich noch 'ne zweite Maus) und natürlich die RX 5700 als PCIe-Gerät hinzugefügt.

Um eine SSD als Gerät hinzufügen zu können - weil *Libvirt/virt-manager* das nicht nativ unterstützt - muss man erst einen regulären Speicherpool (*pool.qcow2*) auf der formatierten und gemounteten SSD erstellen, den dann während der Einrichtung der VM und vor der Installation des OS wieder löschen und die SSD als Speichergerät mit `/dev/disk/by-uuid` hinzufügen. Dann wird die bei der Installation als *raw*-Gerät erkannt, man kann erst die nötigen Treiber installieren und dann das OS.

So, VM gestartet und, alles hängt, kein Bild von der RX 5700 und nur die zweite Maus und ein "Ausschalten erzwingen" konnte helfen. Ich mach' jetzt erstmal 'n Püschchen und berichte später nochmal..

EDIT #1:

Okay, ich hab' nochmal mit dem Boot-Medium rumexperimentiert (es musste wohl eine SATA CD-ROM sein) und jetzt ging's, also wenigstens über die Spice-Videoausgabe im Fenster vom virt-manager. Ich hab' zwar vergessen, welcher der richtige VirtIO-Treiber war, aber hab' dann diese hilfreiche [Anleitung](#) gefunden.

Nach der Installation von Windows 10 also noch schnell den VirtIO "Guest Agent" installiert und in dem Zuge auch gleich Firefox und die Radeon-Software. Die RX 5700 wird auf jeden Fall schon im Geräte-Manager angezeigt. Erstmal das System herunterfahren, Windows-ISO und VirtIO Treiber-Image aus der Bootreihenfolge nehmen und die SSD von SATA zu SCSI ändern. Tja und ein Neustarten der VM funktioniert dann nicht, wegen `Unknown PCI header type '127' for device '000:28:00:0'`. Das ist die RX 5700 und der berühmte *Reset Bug* - großartig! Aber ich hab's ja geahnt. Ich verabschiede mich erstmal wieder für eine kurze Recherche oder mindestens einen Reboot meines Host-Systems.

EDIT #2:

Fuck, ich hätte die SSD `by-id` und nicht `by-uuid` einbinden sollen! Ersteres ist Hardware-spezifisch, zweiteres Partitions-spezifisch. Entscheidender Unterschied. Nach 'nem Reboot wird die SSD von Linux als "unformatiert" erkannt und hat eine andere UUID, als die im virt-manager hinterlegte und beim Starten der VM meckert der virt-manager berechtigter Weise, dass die SSD nicht gefunden wird. Also Windows nochmal neu installieren.

EDIT #3:

Hat zwar keiner nach gefragt, aber hier ist jetzt doch mal 'n Bild von der Maschine. 😊



[Nach oben ↑](#)

Update #2 (24. April 2020):

Ich hab' jetzt die VM neu erstellt, die SSD `by-id` eingebunden und Windows 10 installiert und auch nach einem Neustart des Host-Systems wird die SSD korrekt erkannt und die VM bootet problemlos ins Windows. Einige SCSI-bzw. VirtIO-Treiber hab' ich bereits installiert, wie in der bereits oben verlinkten [Anleitung](#) beschrieben. Ich kann die VM aber trotzdem nicht mit der SSD als SCSI-Gerät, sondern nur als SATA-Gerät starten, sonst lande immer in der Shell. Ist aber halb so wild, Hauptsache, das System läuft erstmal und das Problem kommt auf meine To-Do-Liste.



installiert, Windows Updates gemacht, Host neugestartet, Gast auch neu gestartet (weil ich mich noch nicht um den Reset-Bug gekümmert habe) aber die RX 5700 wird nicht erkannt und die Radeon Software sagt mir, die Treiber seien nicht installiert.. Großartige Hilfe!

Nach kurzem Bemühen einer Suchmaschine kam ich dann drauf, mal im Geräte-Manager zu schauen, was die Grafikkarte eigentlich für 'n Gerätestatus hat und sehe den Fehlercode 43. Nach wiederum kurzer Recherche scheint das Problem mit den Kernel-Parametern zu tun zu haben, die den EFI-Framebuffer deaktivieren. Also die VM nochmal herunterfahren, diesmal mit Absturz des Linux-Hosts inklusive, neu gestartet und Kernel-Parameter geändert. Laut dem [erstbesten Reddit-Post](#) soll man `video=efi:off` statt `video=efi:off,efi:off` nutzen. In dem Zuge hab' ich auch gleich `nomodeset` ergänzt und alles neu gestartet, leider wieder ohne Erfolg! Vielleicht hilft ja der erste [Troubleshooting-Punkt](#) aus der Arch Wiki. Aber das schaue ich mir später mal an, nach der erfolgreichen Neuinstallation der VM war der Rückschlag gerade wieder genug für heute..

Was bleibt? To-Dos!

- Fehlercode 43 der Grafikkarte
- SCSI-Treiber der Festplatte installieren
- generell den AMD Reset-Bug patchen (vielleicht [hier](#) und [hier](#))
- Booten von der RX 5700 verhindern

[Nach oben ↑](#)

Update #3 (27. April 2020):

Ich war wohl am Freitag nicht so ganz auf der Höhe und hab' den `video=efi:off`-Parameter schlicht falsch eingetragen. Jetzt hat es funktioniert, der Fehlercode 43 ist verschwunden und die Grafikkarte wird richtig erkannt!

Außerdem hab' ich nochmal zum Reset-Bug recherchiert und mit dem Manjaro Kernel 5.3 bzw. 5.4 LTS ist der erste von zwei Patches enthalten. Hier der [Thread](#) zum *Navi Reset Kernel Patch*, hier der [Beitrag](#) zum Manjaro-Kernel und hier der [Beitrag](#) zum zweiten Patch.

Ich hab' einfach mal von Kernel 5.6.7-1 auf 5.4.35-1 *gedowngraded* und konnte meine VM tatsächlich herunterfahren, ohne dass der Host abgestürzt ist. Ein Neustart der VM hat allerdings nicht funktioniert, die GPU scheint also immer noch nicht richtig zu resetten. Auch ein zweites Mal kann ich die VM herunterfahren und mein Linux-System stürzt nicht ab. Aber ein Blick auf `journalctl -b -p err` - also alle Fehlermeldungen seit dem letzten Boot - zeigt folgende Fehlermeldungen:

Code:

```
1 Apr 27 14:22:16 TaihuLight kernel: AMD-Vi: Completion-Wait loop timed out
2 Apr 27 14:22:16 TaihuLight kernel: iommu ivhd0: AMD-Vi: Event logged [IOTLB_INV_TIMEOUT device=28:00
```

Ich komme wohl nicht drumherum, den Kernel nochmal manuell zu patchen..

Außerdem habe ich mal meine Spiele-Partition ebenfalls `by-id` eingebunden, um nicht jedes Mal auf's neue die knapp 90 GB GTA Online herunterladen zu müssen. Das Spiel läuft auf jeden Fall butterweich auf hohen bis sehr hohen Grafikeinstellungen in 2560x1440 und ~60 Fps. Das Ziel meines ursprünglichen Upgrade-Threads hab' ich auf jeden Fall erreicht, aber das war bei der Hardware auch irgendwie klar. Ich überlege noch, wie ich am besten die Performance von Prozessor und Grafikkarte in der VM mit *bare-metal*-Performance vergleiche, vielleicht nehm' ich da einfach mal Test von Computerbase als Referenz.

Erstmal bin ich wirklich froh, wie schnell die Einrichtung der virtuellen Maschine lief. Mal abgesehen von Recherche am Rande und ein paar "Flüchtigkeitsfehlern" war das schon fast routiniert im Vergleich zu meinem ersten Versuch, bei dem ich mir erst noch wirklich alles aneignen musste. Jetzt werd' ich mir wohl mal aneignen müssen, wie man 'n Manjaro-Kernel patched und kompiliert und denke, der [Thread](#) ist 'n guter Anfang.

EDIT #1:

Ich hab' jetzt mal 'n paar Benchmarks angeschmissen, weil es mich doch sehr interessiert, wie die VM überhaupt performt. ComputerBase hat ja eigene Test bzw. Benchmarks, sowohl vom [Ryzen 5 3600](#), als auch von der [RX 5700](#), die ich hier als Vergleich heranziehe.

Nochmal kurz die Infos zur VM:

Windows 10 Pro v1909 Build 18363.815
AMD Ryzen 5 3600 @3,60 GHz
12 GB G.Skill Aegis DDR4-3200, CL16
MSI Radeon RX 5700 Evoke GP OC, 8GB
Radeon Software v20.2.2

Beginnen wir mit der Grafikkarte!

Ich hab' mir *Shadow of the Tomb Raider* über Steam heruntergeladen, weil es in den Benchmarks von ComputerBase auftaucht, eine Trial Version zur Verfügung steht und einen eingebauten Benchmark hat. Ich hab' mich dabei an die



dazu überreden, exakt den CB-Benchmark zu machen, falls das mit der Trial Version überhaupt möglich ist.

Hier die Ergebnisse:



Die durchschnittlichen 65 FPS deutlich höher, als die 55,3 von ComputerBase, aber das kann an verschiedenen Dingen liegen, vor allem wahrscheinlich am integrierten Benchmark, das einfach nicht die tatsächliche Performance abbildet. Notiz am Rande, ich hab' auf die Schnelle [diesen Artikel](#) gefunden, der witziger Weise anhand von *Tomb Raider* erläutert, warum Benchmark-Tools problematisch sind. Irgendwie 'n unbefriedigendes Ergebnis, aber ich müsste nochmal nach Benchmarks von aktuellen Spielen suchen, die ich auch besitze.

So, erstmal weiter zum Prozessor:

Ich hab' einfach Cinebench R15 und R20 jeweils einmal im Multi Core-Test und einmal im Single Core-Test laufen lassen. Keine Ahnung warum, aber Cinebench R15 erkennt meinen Prozessor als 12 Kerner und mein Betriebssystem als Windows 8, aber das wird dem Benchmark ja keinen Abbruch tun.

Das sind die Ergebnisse, in den Klammern stehen die Ergebnisse von ComputerBase und die Differenz in Prozent:

Cinebench R15

CPU (Multi Core): 1499 cb (1581 cb; -5,19 %)

CPU (Single Core): 190 cb (197 cb; -3,55 %)

Cinebench R20

CPU (Multi Core): 3382 pts (3539 pts; -4,44 %)

CPU (Single Core): 465 pts (474 pts; -1,90 %)

Bis auf den realitätsfernen Grafikkarten-Benchmark kann man wahrscheinlich sagen, dass die Performance relativ ähnlich ist. Laut dem Cinebench-Benchmarks ist der Prozessor im Durchschnitt keine 5% langsamer, als bei dem Benchmark von ComputerBase. Das ist aber auch nicht sonderlich verwunderlich, weil ja alle CPU-Kerne bzw. Thread und die komplette GPU durchgereicht werden.

Ich werde mich jetzt erstmal wieder dem Kernel-Patch widmen..

[Nach oben ↑](#)

Update #4 (28. April 2020):

Nur ein ganz kurzes Update, aber ein großer Erfolg!

Die bereits verlinkte [Anleitung](#) hat gleich in zwei Punkten nicht für mich zugetroffen. (Die Seite ist übrigens von einem Tag auf den anderen mal off- und mal online, darum oben der [Direktlink](#) und hier ein Link zum [Web Archive](#).)

Also 1. ist der Kernel-Parameter in der Anleitung falsch bzw. ich glaube veraltet und damit redundant beschrieben (und man muss ihn richtig abtippen!) und 2. musste ich das [Compatibility Support Module \(CSM\)](#) ausschalten und nicht einschalten. Das CSM ermöglicht älterer Hardware das Booten mit Legacy BIOS statt UEFI. Naja, jedenfalls hab' ich es mehr aus Spaß einfach mal ausgeschaltet und siehe da, mein Computer bootet von der GTX 970! Jetzt bleibt mir endlich das nervige Display-Input wechseln nach dem Eingeben meines Festplatten-Passworts erspart und mir wird obendrein noch das schicke Mainboard-Logo und die Boot-Hotkeys angezeigt - wunderbar. Jetzt aber versprochen als nächstes der Kernel-Patch..

[Nach oben ↑](#)

Update #5 (29. April 2020):

Noch ein kleines Update, weil mir gerade zwei Dinge in den Kopf gekommen sind:

Man sollte die angelegten Speicherpools, die für das Durchreichen von Festplatten bzw. Partitionen nötig sind natürlich wieder löschen. Bei mir waren das einmal eine 20 GB Datei mit der `id` der SSD im home-Verzeichnis und eine entsprechende Datei auf der Partition, auf der meine Spielebibliothek liegt. Außerdem sollte man die im *virt-manager* hinterlegten Pfade dieser Speicherpools löschen. Das ist nicht besonders wichtig, aber verhindert zum einen eine Fehlermeldung im *journalctl* vom *libvirt*-Service und verhindert außerdem Verwirrung beim hinzufügen von weiteren Speicherpool bzw. -geräten. Dazu muss man den *virt-manager* und die Konfiguration irgendeiner virtuellen Maschine öffnen, dort auf "Gerät hinzufügen" -> "Speicher" -> "Benutzerdefinierten Speicher auswählen oder erstellen" -> "Verwalten..." und dann öffnet sich die Übersicht über die "Speicherdatenträger". Hier habe ich zwei Pfade, die auf die ehemals in meinem `/run/media`-Verzeichnis gemounteten Festplatten verweisen und einfach gelöscht werden können.

Außerdem hab' ich (wie viele) den *Ryzen*-Hypetrain mitgefahren und hab' mich (ebenfalls wie viele) über die hohen Temperaturen gewundert. Mit dem Standardtakt von 3,6 GHz komme ich unter Last von ~ 85% CPU-Auslastung durch *Folding@Home* auf 85° C. Nebenbei steht mein alter Intel Core i5-3570 mit Standardtakt unter 100% Volllast durch *Folding@Home* bei gemüthlichen 50° C in der Zimmerecke - allerdings auch ohne Gehäuse. Mit aktiviertem *MSI*

das?

Einerseits möchte ich natürlich alle für [Folding@Home](#) begeistern!

Andererseits möchte ich aber vor allem auf dieses [Statement](#) eines AMD-Mitarbeiters auf Reddit aufmerksam machen, was nicht nur detailliert erklärt, wie und warum die Temperaturen zustande kommen, sondern auch, wie und mit welchen Einstellungen man seinen *Ryzen*-Prozessor am besten nutzt.

Ansonsten läuft meine GTX 970 auch ganz wunderbar unter Manjaro Linux und weniger anspruchsvolle Spiele wie Minecraft (mit Shadern) oder CS:GO laufen hervorragend. Und ich habe letzens noch diese [Anleitung](#) entdeckt, die erklärt, wie man mit [PRIME](#) eine Grafikkarte sowohl unter dem Host- als auch unter dem Gast-System verwenden kann. Wenn ich denn irgendwann mal den Reset Bug bearbeitet habe, juckt es mich außerdem noch in den Fingern, das auszuprobieren..

[Nach oben ↑](#)

Update #6 (2. Mai 2020):

Wie könnte man ein langes Wochenende besser nutzen, als damit, das erste mal einen Linux-Kernel zu patchen, zu kompilieren und diesen dann zu installieren? Genau das habe ich gerade endlich auf meinem Manjaro Linux-System getan und möchte euch berichten, wie das funktioniert hat!

Ich hab' mich vor allem an diese kurze [Anleitung](#) gehalten und die dort verlinkten Wiki-Artikel.

Zunächst sollte man nachschauen, welchen Kernel man aktuell nutzt und welche installiert sind. Das kann man mit dem Befehl `mhwd-kernel -li`. In meinem Fall ist das "5.4.35-1-MANJARO" bzw. "linux54". Ich hab' mich jedoch dafür entschieden, gleich Linux 5.6 zu installieren. Damit habe ich einerseits einen aktuelleren Kernel, andererseits überschreibe ich beim Installieren nicht den aktuell installierten Kernel. Es ist immer ratsam mindestens zwei Kernel installiert zu haben. So kann man bei einem fehlerhaften Kernel immer noch auf einen stabilen Kernel zurückgreifen.

Als nächstes sollte man sich vergewissern, dass man `git` installiert hat und es zur Not einfach installieren:

```
$ pacman -S git
```

Jetzt klonen wir das Git-Repository des bevorzugten Kernels in einen neuen Ordner. Also einen neuen Ordner im

`/home`-Verzeichnis erstellen:

```
$ mkdir /home/arjab/build
```

In das Verzeichnis wechseln:

```
$ cd /home/arjab/build
```

Und das Repository klonen:

```
$ git clone https://gitlab.manjaro.org/packages/core/linux56.git
```

Dabei ist wichtig, `git` nicht mit `sudo` auszuführen, da der `makepkg`-Befehl später nicht als `sudo` ausgeführt werden darf und dann keine Schreibrechte für den Ordner hat.

Jetzt sollte in dem `/build`-Verzeichnis ein Ordner sein, der wie der Kernel benannt ist - also "linux56" - und zahllose Dateien enthält. Unter anderem einige *.patch-Dateien, in die sich unser *Navi Reset Bug Patch* einreicht.

An dieser Stelle sollte man in das `/linux56`-Verzeichnis wechseln:

```
$ cd ./linux56
```

Dann [hier](#) den Text des Patches kopieren und eine neue Datei erstellen:

```
$ nano navi.patch
```

Den Patch einfügen, eine Zeile wie in diesem [Post](#) beschrieben ändern:

```
mmio = ioremap(mmio_base, mmio_size); statt  
mmio = ioremap_nocache(mmio_base, mmio_size);
```

Die Datei speichern und den Editor verlassen.

Jetzt müssen wir dem `PKGBUILD`-Skript noch "mitteilen", dass er unseren Patch beim Kompilieren des Kernels auch einfügen soll. Hierzu bearbeiten wird die `PKGBUILD`-Datei mit: `$ nano PKGBUILD`

In der Datei findet man unter der Zeile, die mit `source=...` beginnt eine Liste mit verschiedenen Dateien, die alle auf `.patch` enden. Hier fügt man eine neue Zeile ein und ergänzt den Dateinamen des Patches. In meinem Fall schlicht `'navi.patch'`.

Außerdem muss unter der Zeile, die mit `prepare()` beginnt ebenfalls der Patch eingetragen werden:

```
patch -Np1 -i "${srcdir}/navi.patch"
```

Dann die Datei einfach speichern und den Editor beenden.

Da sich jetzt die Prüfsumme der `PKGBUILD`-Datei durch den ergänzten Patch geändert hat, müssen wir diese neu

Nun kann man den Kernel kompilieren. Das kann bis zu einige Stunden dauern, bei ging's mit lediglich 'ner Viertelstunde relativ schnell:

```
$ makepkg -s
```

Danach kann man den kompilierten Kernel mit `$ makepkg -i` installieren. Dabei wird quasi `pacman -U` für die erstellten Pakete `linux.56.pkg.tar.xz` und `linux56-headers.pkg.tar.xz` ausgeführt.

Jetzt das System neu starten und wieder mit `mhwd-kernel -li` überprüfen, ob der neue Kernel installiert ist.

Ich hab' euch bis zu dieser Stelle übrigens erspart, jeden noch so kleinen Fehler und meine zahllosen Versuche des Kompilierens zu dokumentieren. So hat dieses Update noch mehr den Charakter einer kurzen Anleitung und wahrscheinlich wird es niemanden interessieren, wie oft ich vergeblich versucht habe `makepkg -s` auszuführen.

Aber vor allem sollte jetzt hoffentlich der *Reset Bug* gepatcht sein!

Also die virtuelle Maschine gestartet, um sie dann sofort wieder herunterzufahren. Leider spuckt `journalctl` mir immer noch einige Fehlermeldungen aus, unter anderem:

Code:

```
1 02.05.20 15:32 kernel AMD-Vi: Completion-Wait loop timed out
2 02.05.20 15:32 kernel ahci 0000:03:00.1: AMD-Vi: Event logged [IO_PAGE_FAULT domain=0x0010 address=
3 02.05.20 15:32 kernel AMD-Vi: Event logged [IO_PAGE_FAULT device=03:00.1 domain=0x0010 address=0xa
4 02.05.20 15:32 kernel iommu ivhd0: AMD-Vi: Event logged [IOTLB_INV_TIMEOUT device=28:00.0 address=
```

Ein erneutes Starten der VM funktioniert zwar ohne Fehlermeldung vom *virt-manager*, aber leider ist `journalctl` währenddessen wild dabei, Fehlermeldungen ausspucken und ein Signal kommt auch nicht aus der RX 5700 - schade eigentlich!

Ich werd' das Prozedere nochmal mit anderen Kernel-Versionen wiederholen und ein bisschen weiter recherchieren. Wenigstens weiß ich jetzt, wie man 'n Kernel patcht und kompiliert und dass der *Reset Bug* nicht so leicht zu beheben ist, wie gedacht.

[Nach oben ↑](#)

Update #7 (3. Mai 2020):

Heute nochmal ein kleines, aber feines Update zum Audio der virtuellen Maschine.

Tatsächlich ist es nämlich häufig relativ problematisch, das Audiosignal der VM verzögerungsfrei und in voller Qualität aus dem Ausgabegerät der Wahl zu kriegen. Außerdem besitzt man - wie auch bei Tastatur, Maus und sonstigen Peripheriegeräten - wahrscheinlich nur ein Paar Lautsprecher oder Kopfhörer, über das man je nachdem das Audiosignal von Host- oder Gastsystem hören möchte.

In meinem Fall wird die Sache noch ein bisschen schwieriger, weil ich ein USB-Audiointerface benutze, an dem sowohl meine Lautsprecher, als auch meine Kopfhörer angeschlossen sind. Natürlich könnte man einfach den USB-Anschluss an die VM weiterreichen, was aber oft viel zu hohe Latenz oder furchtbare Verzerrungen mit sich bringt. Man könnte stattdessen auch das PCI-Gerät, an das der entsprechende USB-Controller angeschlossen ist an die VM weiterreichen. Damit werden aber auch alle anderen USB-Anschlüsse an die VM weitergereicht, die mit diesem PCI-Gerät verbunden sind und meine letzte virtuelle Maschine hat sich mit dieser Lösung oft schwer getan.

Deutlich einfach ist es, das Audiosignal der virtuellen Maschine mit *PulseAudio* an das Host-System quasi zurückzureichen. Der [Abschnitt](#) dazu in der Arch Wiki hat mir ehrlich gesagt nicht auf Anhieb eingeleuchtet, dafür aber dieser [Artikel](#) von *Passthrough Post*. Seit *Qemu 4.0* reicht es nämlich aus, einige wenige Zeilen in der Konfigurationsdatei der VM zu ändern. Dazu muss man die XML-Datei (am einfachsten mit *nano*) editieren:

```
$ EDITOR=nano virsh edit windows10
```

Hier müssen dann folgende Zeilen geändert werden:

Code:

```
1 <domain type='kvm' xmlns:qemu='http://libvirt.org/schemas/domain/qemu/1.0'>
2 ...
3   <qemu:commandline>
4     <qemu:arg value='-audiodev' />
5     <qemu:arg value='pa,id=pa1,server=/run/user/1000/pulse/native' />
6   </qemu:commandline>
7 </domain>
```

Dazu installiert man sich am besten *pavucontrol*, um den Audiostream einstellen zu können. Und ein kurzer Test zeigt, dass der Audiostream meiner VM *win10* mit der gesetzten PulseAudio-ID *pa1* korrekt an mein USB-Audiointerface *Steinberg UR22* weitergegeben wird:



Ich hab' während der Konfiguration übrigens die Grafikkarte aus der VM entfernt, um diese zum Testen einfach hoch- und wieder runterfahren zu können, daher die Videoausgabe über das Fenster vom *virt-manager*.

Andere Peripheriegeräte, wie z.B. Maus und Tastatur oder ein USB-Controller kann man übrigens problemlos als USB-Gerät an die VM durchreichen. Bei Eingabegeräten muss man sich nur darüber im Klaren sein, dass diese dann an die VM weitergereicht werden und nicht mehr für das Host-System funktionieren. Das ist vor allem wichtig, falls die VM abstürzt oder anderweitig nicht funktioniert, aber trotzdem die Eingabegeräte für sich beansprucht. Hierfür kann man entweder auf eine zweite Maus oder Tastatur, über *SSH*, *VNC* oder Software wie *Teamviewer* auf das Host-System zugreifen.

[Nach oben ↑](#)

Update #8 (20. Mai 2020):

Die letzten Wochen hab' ich mich mit verschiedenen Dingen rumschlagen müssen, von denen ich jetzt mal gesammelt berichte. Darunter ein Bluescreen der virtuellen Maschine, die richtigen Einstellungen des SCSI-Treibers für die Festplatte und schließlich Problemen mit PulseAudio.

1. Nach einem Update auf QEMU Version 5 hat mein Windows 10 mir nach dem Hochfahren einen Bluescreen mit der Fehlermeldung *KERNEL SECURITY CHECK FAILURE* angezeigt. Ich dachte, dass bei einem der zahllosen Abstürze meines System aufgrund des Navi Reset Bugs beim Rumexperimentieren in den letzten Tagen irgendwelche Systemdateien beschädigt wurden. Daraufhin hab' ich wie verrückt versucht, das Windows-System mit einem Installationsmedium zu reparieren, aber nichts hat funktioniert. Nach weiterer Recherche bin ich dann irgendwann drauf gekommen, dass es schlicht an QEMU liegt und man - wie in diesem [Post](#) beschrieben - nur in der CPU-Konfiguration der virtuellen Maschine das Modell von *host-passthrough* auf *host-model* ändern muss. Hier gibt es wohl auch noch anderes Workaround über die XML-Konfigurationsdatei der virtuellen Maschine, aber das schaue ich mir bei Gelegenheit an.

2. Ähnlich abgemüht hab' ich mich mit dem fehlenden Treiber für den *SCSI-Controller* und hab' wirklich versucht jeden einzelnen Treiber zu installieren, aber immer ohne Erfolg. Ich weiß schon gar nicht mehr, wie ich hier drauf gekommen bin, aber es lag ebenfalls nicht an Windows, sondern der falschen Konfiguration im *virt-manager*. Hier hat jedes (Speicher-)Gerät einen zugehörigen Controller, entsprechend gibt es auch einen *Controller Virtio SCSI 0*, an dem meine beiden Festplatten angeschlossen sind. Hier war das Modell des Controllers auf *Hypervisor-Standard* gestellt, benötigt jedoch *VirtIO SCSI* um zu funktionieren. Also hab' ich das kurz geändert, es hat auf Anhieb funktioniert und der *SCSI-Controller* ist nicht mehr als "treiberlos" im Geräte-Manager aufgetaucht.

3. Nach diesen zwei wirklich ärgerlichen Flüchtigkeitsfehlern hat außerdem irgendwann der Audio-Stream mit PulseAudio nicht mehr funktioniert und ich hab' bis jetzt nicht verstanden, was das Problem war. Ich hab' schlicht einen entscheidenden Teil der Anleitung aus der [Arch Wiki](#) vergessen umzusetzen oder hier zu dokumentieren und irgendwie hat es bei der ersten Einrichtung trotzdem funktioniert. Darum hier der Nachtrag: Man muss auch die */etc/libvirt/qemu.conf*-Datei ändern und den eigenen Benutzernamen eintragen, damit PulseAudio weiß, an wessen Sitzung das Audiosignal gestreamt werden soll:

```
nano /etc/libvirt/qemu.conf
```

Dann die Zeile `user = "example"` auskommentieren und *example* durch den eigenen Benutzernamen ersetzen. Danach den Computer oder einfach den *libvirtd.service* neustarten:

```
systemctl restart libvirtd.service.
```

So langsam werd' ich leider auch zunehmend pessimistischer was den Navi Reset Bug angeht. Ich werd' gleich nochmal den Kernel 5.7 patchen und installieren, aber denke nicht, dass das funktionieren wird. Entweder ich kaufe mir auch doch noch eine RTX 2060, aber die hat weniger Leistung als die RX 5700 oder ich verwerfe das *PCI-Passthrough*, was wirklich schade wäre! Ich bin unschlüssig..

[Nach oben ↑](#)

Update #9 (21. Mai 2020):

Ich habe gestern den Kernel 5.7 gepatcht, (in nur 19 Minuten!) kompiliert und installiert und wurde nach einem Neustart von einem schwarzen Bildschirm begrüßt. Glücklicher Weise hab' ich nicht wieder Ewigkeiten gebraucht, um das Problem zu finden und zu beheben. Es hat schlicht der proprietäre Nvidia-Grafikkartentreiber gefehlt. Also schnell die für den Kernel passende Version installiert:

```
$ pacman -S linux57-nvidia-440xx
```

Bei der installation werden gleich alle installierten Kernel aktualisiert und nach einem Neustart bin ich auch wieder auf meinem gewohnten Desktop gelandet. Anstatt die Grafikkartentreiber manuell zu installieren, kann man das übrigens auch einfach mit [nvidia-dkms](#) automatisieren.

Wieder ist aber die viel spannendere Frage, ob der gepatchte Kernel den *Navi Reset Bug* beheben konnte, aber leider

beim gleichen Ergebnis raus..

Spannend finde ich dabei aber, dass nicht die üblichen Kernel-Fehlermeldungen auftauchen - also gar keine. Vielleicht werde ich da nochmal im [Thread](#) vom Level1Tech-Forum nachfragen, ob und wie man den Absturz debuggen kann.

Trotzdem bin ich heute optimistischer was die ganze PCI-Passthrough Geschichte und den elendigen *Reset Bug* angeht. Doch noch eine Nvidia Grafikkarte zu kaufen erscheint mir ziemlich sinnlos und wäre auch einfach deutlich teurer, als das ganze Projekt ohnehin schon ist. Und auch ein Dual-Boot System erscheint mir wie ein fauler Kompromis. Tatsächlich gefällt mir die Situation gerade nämlich doch besser, als ursprünglich angenommen und entspricht fast meinen Vorstellungen. Mein System bootet immer und ausschließlich ein Linux-System und kein Windows kommt mir mehr als *bare-metal* Installation ins Haus. Aus meinem Linux-System heraus kann ich dann die virtuelle Maschine starten, zocken und dann leider in einem Rutsch und gezwungener Maßen beide Systeme herunterfahren - aber sei es drum. Tatsächlich dient das Windows Gast-System ja auch "nur" dazu, die Spiele spielen zu können, die unter Linux gar nicht laufen, wie z.B. GTA Online. Wenn ich dann in absehbarer Zukunft doch noch hinbekomme, die RX 5700 unter beiden Systemen zu verwenden, wäre die gesamte Situation noch besser, aber wahrscheinlich macht mir hier auch der *Reset Bug* einen Strich durch die Rechnung.

Abgesehen davon würde ich an dieser Stelle auch nochmal ein positives Zwischenfazit ziehen, wenn ich mir meine bisherige *Wall of Text* so anschau! Nicht nur hat es Spaß gemacht, mal wieder einen Computer zusammenzubauen, ich hab' auch die üblichen Fehler des PCI-Passthroughs bewältigt, sei es von der richtigen Grafikkarte im "falschen" PCI-Slot zu booten oder der klassische *Fehlercode 43*. Außerdem hab' ich meinen ersten Kernel kompiliert und die virtuelle Maschine mit SCSI-Treibern und PulseAudio nach meine Wünschen konfiguriert. Und obwohl die Rückmeldung hier eher verhalten ist, bin ich doch auch froh drüber, diesen Erfahrungsbericht geschrieben zu haben. Was man während einer Pandemie halt so macht..

[Nach oben ↑](#)

Update #10 (30. Mai 2020):

Ich möchte mich auch hier nochmal kurz dafür bedanken, dass mein Erfahrungsbericht es zu einer Notiz auf der Startseite geschafft hat! Ansonsten löchert mich gerne mit Fragen und Anregungen, ich versuche auf alles zu antworten! Gerade weiß ich auch nicht so richtig, wie's weitergehen könnte, als falls ihr noch Ideen habt, was ich noch ausprobieren könnte oder gerne mehr Hintergrundinfos zu bestimmten Themen hättet, immer her damit. Vielleicht probier' ich ja doch mal aus, die RX 5700 auf beiden Systemen zu verwenden und GTA Online unter Linux zum laufen zu bringen..

Und auf Wunsch von [@lokon](#) hier meine BIOS-Version und IOMMU-Gruppen:
BIOS via [dmidecode](#):

Spoiler

IOMMU-Gruppen via [Skript](#):

Spoiler

[Nach oben ↑](#)

Update #11 (20. November 2020):

Ich melde mich mal zurück, weil sich in letzter Zeit einiges geändert hat!

Zum einen hab' ich mein komplettes System neu installiert, meine GTX 970 rausgeschmissen und PCI-Passthrough den Rücken gekehrt. Warum? Vor allem, weil [GTA Online einfach hervorragend mit Lutris](#) läuft. Zum anderen, weil ich ein rastloser Nerd bin und mich auch mal mit [Wine](#), [Proton](#) und [Lutris](#) auseinandersetzen wollte. Nebenbei läuft Counter Strike: Global Offensive auch nativ super auf Linux.

Falls jemanden meine ambivalente Erfahrung mit *Gaming on Linux* interessiert, kann ich dazu gerne noch mehr Erfahrungen teilen.

Zum anderen hat Gnif, der Entwickler des Navi Reset Kernel Patches mit zwei Kollegen ein [Kernel Modul](#) entwickelt, mit dem man einerseits Grafikkarten zurücksetzen kann ohne den Kernel patchen zu müssen. Andererseits sind die neuen Grafikkarten von Nvidias RTX 3000 Serie und vom AMDs Radeon 6000 Serie weder vom Error Code 43 noch vom Reset Bug betroffen. Das behauptet jedenfalls [Linus](#) und im Fall von AMD bestätigt das [Wendell](#) von Level1Tech. Zu Nvidia konnte ich selber gerade nichts finden und auch [Reddit](#) hat noch keine Bestätigung dazu gefunden.

Also, vergesst nahezu alles, was ich geschrieben habe - juhuu! PCI-Passthrough ist gerade wahrscheinlich einfacher denn je, sofern man eine der neuen Grafikkarten besitzt. Und da es sich bei den Fehlern um Software-seitige Probleme handelt, ist davon auszugehen, dass auch ältere Grafikkarten mit einem Treiber oder Firmware-Update nicht mehr von den Fehlern betroffen sein werden. Sofern Nvidia und AMD gütig sind..

Oder man spart sich gleich die Mühe, wenn man die gewünschten Spiele in der [LutrisDB](#) oder [ProtonDB](#) findet und spielt einfach auf Linux!

Austausch über andere Erfahrungen. Und jetzt versuche ich erstmal Red Dead Redemption 2 zum laufen zu bringen..

[Nach oben 1](#)

17. April 2020

Zuletzt bearbeitet: 29. Dezember 2020 (Update #1)

**Ichthys**

Lt. Commander

Dabei seit: März 2008
Beiträge: 1.094

Klingt interessant, ich hab es zwar noch nicht gemacht und aktuell auch nicht solche Projekte, aber werde definitiv ab und zu reinschnuppern. Danke!

[Review: Thermalright True Spirit 90 M Rev. A](#)
[Erlebe alle Cherry MX-Module mit nur einem Gerät!](#)

"Wir haben die Erde nicht von unseren Eltern geerbt, sondern von unseren Kindern geliehen"

Mein High-End-System:

Pentium-S 133 Mhz, 48 MB Edo-Ram, 2 GB FP, 2 MB-Graka, Ess 1868 (Sound). Da kommen eure lahmen Gurken nicht mit, ne! 

Arjab

**Arjab**

Lt. Junior Grade

Ersteller dieses Themas
Dabei seit: Feb. 2013
Beiträge: 416

27. April 2020

 #3

Ich hab' mir übrigens gedacht, in unregelmäßigen Abständen auch hier die Updates zu dokumentieren, damit Leute auch Benachrichtigungen bekommen, dass ich den Post aktualisiert habe. Und natürlich wäre es super, wenn noch 'n paar mehr Leute hier mitlesen, ich würde mich über Feedback freuen! 

Also 27. April 2020: Update #3.

[PCI-Passthrough mit Ryzen 5 3600, GTX 970 & RX 5700?](#)

[Review: Thermalright True Spirit 90 M Rev. A](#)

[Erlebe alle Cherry MX-Module mit nur einem Gerät!](#)

tony_mont4n4 und Ichthys

Zuletzt bearbeitet: 2. Mai 2020

**Ichthys**

Lt. Commander

Dabei seit: März 2008
Beiträge: 1.094

28. April 2020

 #4

Liest sich gut!

Mal eine (erste) Frage. Warum soll die virtuelle Festplatte via SCSI angebunden werden? Was versprichst Du Dir davon?

"Wir haben die Erde nicht von unseren Eltern geerbt, sondern von unseren Kindern geliehen"

Mein High-End-System:

Pentium-S 133 Mhz, 48 MB Edo-Ram, 2 GB FP, 2 MB-Graka, Ess 1868 (Sound). Da kommen eure lahmen Gurken nicht mit, ne! **Arjab**

Lt. Junior Grade

Ersteller dieses Themas
Dabei seit: Feb. 2013
Beiträge: 416

28. April 2020

 #5

Freut mich!

Das ist ganz einfach, eine emulierte SCSI-Schnittstelle ist einfach effizienter, als SATA. Ich hab' keine Ahnung, warum das so ist, aber es steht so in der [Arch Wiki](#).

Und übrigens ist die Festplatte nicht virtualisiert, sondern eine physische 120 GB Samsung 840 SSD und nur die Schnittstelle der VM ist virtualisiert.

[PCI-Passthrough mit Ryzen 5 3600, GTX 970 & RX 5700?](#)

[Review: Thermalright True Spirit 90 M Rev. A](#)

[Erlebe alle Cherry MX-Module mit nur einem Gerät!](#)

**Ichthys**

Lt. Commander

Dabei seit: März 2008
Beiträge: 1.094

28. April 2020

 #6

Ah - nicht richtig gelesen. Das schriebst Du ja.  Ich erstelle immer einen virtuellen Datenträger. Ich habe bislang noch nicht eine ganze Platte durchgereicht. Aber ja, das ist ein interessantes Projekt. Jetzt könnte man natürlich auch fies sagen, dass Du schlicht und ergreifend ein Dual-Boot-System installieren könntest, aber dann ginge natürlich der ganze Spaß verloren. 

Das Konzept dahinter finde ich spannend. Es sind im Prinzip zwei Computer in einem, nur dass sie sich einige zentrale Dinge teilen.

"Wir haben die Erde nicht von unseren Eltern geerbt, sondern von unseren Kindern geliehen"

Mein High-End-System:

Pentium-S 133 Mhz, 48 MB Edo-Ram, 2 GB FP, 2 MB-Graka, Ess 1868 (Sound). Da kommen eure lahmen Gurken nicht mit, ne! **Arjab**

Lt. Junior Grade

Ersteller dieses Themas

28. April 2020

 #7

Ich find' die ganze physische Festplatte durchzureichen sehr praktisch, weil man die auch einfach unter'm Host-System mounten kann, wenn die VM ausgeschaltet ist und ganz normal auf das Dateisystem zugreifen kann. Meine 500 GB Partition mit meiner Steam-Bibliothek konnte ich so auch einfach einbinden bzw. durchreichen und musste nichts neu installieren.

Und ja, Dual-Boot wäre auch 'ne leichte Lösung, aber dann hätte ich weder den Komfort, dass ich aus meinem laufenden System heraus *mal eben* meine VM starten kann (vorausgesetzt, alles ist korrekt eingerichtet), noch die



Ich find' das Prinzip von virtuellen Maschinen und vor allem von PCI-Passthrough auch sehr spannend, weil jedes System sich Komponenten der Hardware teilt und jeweils nahezu exklusiv auf andere Komponenten zugreift. Da sind vor allem der Linux-Kernel und Virtualisierungs- bzw. Hypervisorsoftware weit gekommen!

[PCI-Passthrough mit Ryzen 5 3600, GTX 970 & RX 5700?](#)
[Review: Thermalright True Spirit 90 M Rev. A](#)
[Erlebe alle Cherry MX-Module mit nur einem Gerät!](#)



Ichthys

Lt. Commander

Dabei seit: März 2008
 Beiträge: 1.094

28. April 2020

#8

Tja, ich halte es eher umgekehrt. Windows und, wenn mir mal danach ist und der Nachwuchs mir mal ruhige Minuten gönnt, irgend eine Linux-Distri zum herumexperimentieren. Aber das mag meiner Vergangenheit sowie der Tatsache, dass ich neben Web- auch .net-Entwicklung betreibe, geschuldet sein. Außerdem habe ich zu wenig Zeit, noch herumzuxperimentieren, wie man Spiele unter Linux zum Laufen bekommt. Wobei... Eigentlich zeigst Du gerade einen interessanten Weg auf... Hm... Aber andererseits - meine bessere Hälfte und Linux... 😊

Zuletzt bearbeitet: 29. April 2020

"Wir haben die Erde nicht von unseren Eltern geerbt, sondern von unseren Kindern geliehen"

Mein High-End-System:

Pentium-S 133 Mhz, 48 MB Edo-Ram, 2 GB FP, 2 MB-Graka, Ess 1868 (Sound). Da kommen eure lahmen Gurken nicht mit, ne! 😊



Arjab

Lt. Junior Grade

Ersteller dieses Themas

Dabei seit: Feb. 2013
 Beiträge: 416

30. April 2020

#9

Okay, das ist verständlich! Da bin ich als Studi in einer privilegierten Position und vor allem sehr froh, relativ früh mit Linux angefangen zu haben. Jetzt gibt es kein zurück mehr und das ist auch gut so! Naja und für alles andere hab' ich ja meine VM. 😊

[PCI-Passthrough mit Ryzen 5 3600, GTX 970 & RX 5700?](#)
[Review: Thermalright True Spirit 90 M Rev. A](#)
[Erlebe alle Cherry MX-Module mit nur einem Gerät!](#)



Arjab

Lt. Junior Grade

Ersteller dieses Themas

Dabei seit: Feb. 2013
 Beiträge: 416

2. Mai 2020

#10

2. Mai 2020: Update #6 - Kernel Patch

[PCI-Passthrough mit Ryzen 5 3600, GTX 970 & RX 5700?](#)
[Review: Thermalright True Spirit 90 M Rev. A](#)
[Erlebe alle Cherry MX-Module mit nur einem Gerät!](#)

Ichthys



Arjab

Lt. Junior Grade

Ersteller dieses Themas

Dabei seit: Feb. 2013
 Beiträge: 416

3. Mai 2020

#11

3. Mai 2020: Update #7 - PulseAudio

[PCI-Passthrough mit Ryzen 5 3600, GTX 970 & RX 5700?](#)
[Review: Thermalright True Spirit 90 M Rev. A](#)
[Erlebe alle Cherry MX-Module mit nur einem Gerät!](#)

Ichthys



Ichthys

Lt. Commander

Dabei seit: März 2008
 Beiträge: 1.094

6. Mai 2020

#12

Zitat von Arjab: 😊

Okay, das ist verständlich! Da bin ich als Studi in einer privilegierten Position und vor allem sehr froh, relativ früh mit Linux angefangen zu haben. Jetzt gibt es kein zurück mehr und das ist auch gut so! Naja und für alles andere hab' ich ja meine VM. 😊

Genieß die Zeit! Sie kommt nie wieder. 😊

"Wir haben die Erde nicht von unseren Eltern geerbt, sondern von unseren Kindern geliehen"

Mein High-End-System:

Pentium-S 133 Mhz, 48 MB Edo-Ram, 2 GB FP, 2 MB-Graka, Ess 1868 (Sound). Da kommen eure lahmen Gurken nicht mit, ne! 😊

**Arjab**

Lt. Junior Grade

Ersteller dieses ThemasDabei seit: Feb. 2013
Beiträge: 416**20. Mai 2020: Update #8 - Bluescreen, Korrekturen und wie weiter?**

PCI-Passthrough mit Ryzen 5 3600, GTX 970 & RX 5700?
[Review: Thermalright True Spirit 90 M Rev. A](#)
Erlebe alle Cherry MX-Module mit nur einem Gerät!

Ichthys

**Arjab**

Lt. Junior Grade

Ersteller dieses ThemasDabei seit: Feb. 2013
Beiträge: 416

21. Mai 2020

 #14**21. Mai 2020: Update #9 - Kernel patchen und Grafikkartentreiber**

PCI-Passthrough mit Ryzen 5 3600, GTX 970 & RX 5700?
[Review: Thermalright True Spirit 90 M Rev. A](#)
Erlebe alle Cherry MX-Module mit nur einem Gerät!

**Ichthys**

Lt. Commander

Dabei seit: März 2008
Beiträge: 1.094

28. Mai 2020

 #15

Nun bin ich mal dazugekommen, nachzulesen. Da hast Du ja einige Zeit wieder investiert. Liest sich aber gut! 👍
Wirklich spannend finde ich das Verhalten der Grafikkarte. Mein Eindruck ist hierbei, dass der Hosttreiber der Grafikkarte beim Herunterfahren des Gasts in einen ungünstigen Zustand versetzt wird und dann dort festhängt. Dass dabei sogar nun das ganze System sich verabschiedet, spricht m. E. nach dafür.
Spannend, spannend. :-)

"Wir haben die Erde nicht von unseren Eltern geerbt, sondern von unseren Kindern geliehen"

Mein High-End-System:

Pentium-S 133 Mhz, 48 MB Edo-Ram, 2 GB FP, 2 MB-Graka, Ess 1868 (Sound). Da kommen eure lahmen Gurken nicht mit, ne! 🤪

Arjab

**Arjab**

Lt. Junior Grade

Ersteller dieses ThemasDabei seit: Feb. 2013
Beiträge: 416

29. Mai 2020

 #16

Moin, danke für die Rückmeldung!

Soweit ich das verstanden habe wird beim Herunterfahren des Gast-Systems die Grafikkarte ebenfalls neu gestartet. Dabei tritt dann ein Fehler auf, weil AMD die Fähigkeit der Grafikkarte neu gestartet zu werden entweder schlecht, falsch oder gar nicht implementiert haben. Entsprechend wird dem Host-System eine unvollständig neu gestartete Grafikkarte "übergeben", mit der es dann entweder direkt oder beim nächsten Starten des Gast-Systems nicht klar kommt, wilde Kernel-Errors ausspuckt und im schlimmsten (meinem) Fall komplett abstürzt. Deine Hypothese ist also teilweise richtig und wirklich gefixt werden kann das wohl nur mit neuer Grafikkarten-Firmware.

PCI-Passthrough mit Ryzen 5 3600, GTX 970 & RX 5700?
[Review: Thermalright True Spirit 90 M Rev. A](#)
Erlebe alle Cherry MX-Module mit nur einem Gerät!

Ichthys und konkreter

**konkreter**

Captain

Dabei seit: März 2011
Beiträge: 3.434

29. Mai 2020

 #17

Schöner Artikel. Steckt sehr viel Arbeit und wissen darin. Bei der nächsten Kiste wird mein Setup ähnlich sein. Host Linux zum spielen ne VM....

[@SV3N](#) wäre vdas was für die Notizen?

Arjab, SV3N und Ichthys

**Ichthys**

Lt. Commander

Dabei seit: März 2008
Beiträge: 1.094

29. Mai 2020

 #18[@SV3N](#) Ich plädiere auch dafür! Der Artikel bekommt (gefühl) viel zu wenig Aufmerksamkeit für den Umfang.**Ergänzung (29. Mai 2020)****Zitat von Arjab:** ☺

Moin, danke für die Rückmeldung!

Soweit ich das verstanden habe wird beim Herunterfahren des Gast-Systems die Grafikkarte ebenfalls neu gestartet. Dabei tritt

...entweder direkt oder beim nächsten Starten des Guest-Systems nicht korrigiert wird, keine Fehler ausspuckt und im schlimmsten (meinem) Fall komplett abstürzt. Deine Hypothese ist also teilweise richtig und wirklich gefixt werden kann das wohl nur mit neuer Grafikkarten-Firmware.

Das spricht dafür, dass es wohl nie richtig behoben wird. GraKa-Firmwares sind ja doch eher die Ausnahme. Und was AMD da vor ein paar Monaten mit der 5600 XT abgezogen hat, war zwar sehr spannend, dürfte aber kaum ein Paradigmen-Wechsel in dieser Hinsicht darstellen.

Ob man, rein theoretisch, die Grafikkarte einfach nur deaktivieren und anschließend wieder aktivieren könnte? 🤔
Vermutlich dürfte das aber am Ende aufs Selben hinauslaufen. Gab es da mal irgendeine offizielle Reaktion seitens AMDs (hab gerade keine Zeit, nachzugucken)?

"Wir haben die Erde nicht von unseren Eltern geerbt, sondern von unseren Kindern geliehen"

Mein High-End-System:
Pentium-S 133 Mhz, 48 MB Edo-Ram, 2 GB FP, 2 MB-Graka, Ess 1868 (Sound). Da kommen eure lahmen Gurken nicht mit, ne! 🤪

Arjab, konkreter und SV3N



SV3N
Redakteur

Teammitglied

Dabei seit: Juni 2007
Beiträge: 15.386

29. Mai 2020

#19

Der Artikel erhält völlig verdient eine Notiz auf der Startseite. Sehr schöne und umfangreiche Arbeit.

Danke @konkretor und @Ichthys für melden.

Liebe Grüße
Sven

Arjab, tony_mont4n4, Ichthys und eine weitere Person



Ichthys

Lt. Commander

Dabei seit: März 2008
Beiträge: 1.094

29. Mai 2020

#20

@SV3N Gerne! Könntest Du Dir vorstellen, diesem RX 5700-Problem nachzugehen? Hatte, soweit ich weiß, bei CB noch nichts darüber gelesen. Eventuell wäre das ja mal ein interessantes Thema für andere?

"Wir haben die Erde nicht von unseren Eltern geerbt, sondern von unseren Kindern geliehen"

Mein High-End-System:
Pentium-S 133 Mhz, 48 MB Edo-Ram, 2 GB FP, 2 MB-Graka, Ess 1868 (Sound). Da kommen eure lahmen Gurken nicht mit, ne! 🤪

1 2 3 Weiter ▶

Anmelden, um zu antworten.

Ähnliche Themen

L	Hardware für optimalen VFIO build lukaszervu · 8. Januar 2021 · Linux	Antworten: 4 Aufrufe: 325	8. Januar 2021 Rajjin
D	Linux Mint - 2 Grafikkarten Don_2020 · 2. Oktober 2020 · Linux	Antworten: 7 Aufrufe: 775	2. Oktober 2020 lokon L
B	Gaming on Linux/ Gaming VM Installation mit AMD und Nvidia Grafikkarte Schwierigkeiten Big Ed · 13. Dezember 2020 · Linux	Antworten: 12 Aufrufe: 1.055	19. Dezember 2020 Big Ed B
	Notiz Aus der Community: Virtualisierung mit Manjaro Linux SV3N · 29. Mai 2020 · Linux	Antworten: 10 Aufrufe: 2.738	1. Juni 2020 Bigfoot29 B
	Leserartikel Virt-Manager GPU Passthrough Anleitung (Debian/Ubuntu) Transistor 22 · 27. Juni 2020 · Virtualisierung und Emulation 2	Antworten: 24 Aufrufe: 5.403	1. Juli 2020 Transistor 22

Passend zum Thema



Anmelden



50 26.01.2021

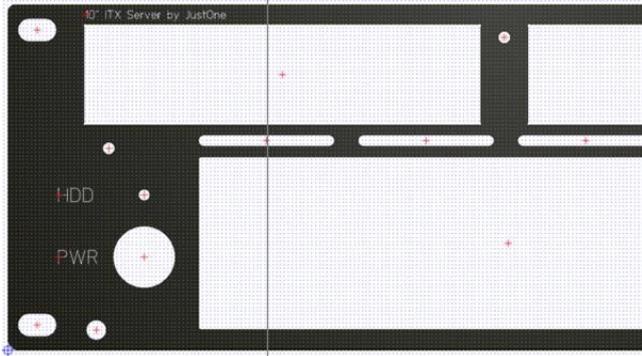


ASROCK MARS 4500U

TEST

Intel-NUC-Alternative mit AMDs APU Renoir

101 25.01.2021



AUS DER COMMUNITY

10-Zoll-Server im Eigenbau

17 23.01.2021

! Nichts mehr verpassen!Erhalte eine Push-Benachrichtigung (oder einen Newsletter) bei Erscheinen neuer Tests und Berichte: [Jetzt anmelden!](#)**📢 Warum Werbebanner?**ComputerBase berichtet unabhängig und verkauft deshalb keine Inhalte, sondern Werbebanner. [Mehr erfahren!](#)**✓ ComputerBase Pro**ComputerBase Pro ist die werbefreie, schnelle, flexible und zugleich faire Variante von ComputerBase. [Mehr dazu!](#)

Feeds Push Facebook Twitter Mastodon YouTube

Einstellungen und
Layout-UmschalterProbleme mit einem
Werbebanner?

© 1999–2021 ComputerBase GmbH

[Impressum](#) [Kontakt](#) [Mediadaten](#) [Datenschutz](#)