

# DPDK auf einem Debian installieren

Erst ein Debian wie gewohnt installieren.

Hugepages einrichten:

```
nano /etc/default/grub
```

Dort dann an den Inhalt von GRUB\_CMDLINE\_LINUX\_DEFAULT folgendes ergänzen  
default\_hugepagesz=1G hugepagesz=1G hugepages=4 intel\_iommu=on

Konfig anwenden:

```
update-grub
```

Dateisystem für Hugetables anlegen:

```
mkdir /mnt/huge_1GB  
nano /etc/fstab
```

```
nodev /mnt/huge_1GB hugetlbfs pagesize=1GB 0 0
```

reboot

```
reboot
```

Prüfen ob hugepages wirklich da sind:

```
cat /proc/meminfo | grep -i huge
```

```
AnonHugePages:      28672 kB  
ShmemHugePages:      0 kB  
HugePages_Total:      4  
HugePages_Free:      4  
HugePages_Rsvd:      0  
HugePages_Surp:      0  
Hugepagesize:      1048576 kB  
Hugetlb:             4194304 kB
```

```
cat /proc/mounts | grep -i /mnt/huge_1GB
```

```
nodev /mnt/huge_1GB hugetlbfs rw,relatime,pagesize=1024M 0 0
```

Testen, ob iommu läuft:

```
ls -al /sys/kernel/iommu_groups/
```

Das Verzeichnis sollte nun nicht mehr leer sein.

Ein paar Pakete installieren:

```
apt update && apt upgrade  
apt install build-essential python3 meson ninja-build libnuma-dev
```

Kernelmodule laden

```
modprobe vfio-pci
```

DPDK herunterladen

```
wget http://fast.dpdk.org/rel/dpdk-20.02.tar.xz  
tar xJf dpdk-20.02.tar.xz  
cd dpdk-20.02  
export RTE_SDK=$(pwd)
```

Geräte anziehen lassen

```
${RTE_SDK}/usertools/dpdk-devbind.py -s
```

Geräte binden

```
${RTE_SDK}/usertools/dpdk-devbind.py --bind=vfio-pci 0000:01:00.0  
${RTE_SDK}/usertools/dpdk-devbind.py --bind=vfio-pci 0000:01:00.1
```

DPDK bauen

```
cd ${RTE_SDK}  
make defconfig  
make
```

Ein Beispiel ausführen:

```
cd ${RTE_SDK}/examples/helloworld  
export RTE_TARGET=build  
make
```

From:

<https://wiki.clerie.de/> - **clerie's Wiki**

Permanent link:

<https://wiki.clerie.de/notiz/dpdk-debian-installieren?rev=1590145559>

Last update: **2020/05/22 13:05**

