

freebsd hinzufügen von laufwerken-2

freebsd hinzufügen von laufwerken



FreeBSD – How to format & partition additional hard disk

While housekeeping the external hard disk laying around, I managed to squeeze out one spare hard disk out of this cleaning practice. The siz...

[expand title="mehr lesen..."]

While housekeeping the external hard disk laying around, I managed to squeeze out one spare hard disk out of this cleaning practice. The size is big enough to act as a [backup media for my FreeBSD server](#) & desktop workstations. The external hard disk partition layout preferred would be 3 partitions. The 1st (backup of OS & data) & 2nd partition (data backup) will be [FreeBSD](#) UFS & the 3rd will be FAT32 (for media transfer).

Since my workstation doesn't support [FreeBSD](#) UFS file system, I'll partition & format it using the [FreeBSD](#) server. All tools involve are command line utilities and can be done through

[remote secure shell](#). This post can also be serve as a guide for adding new/used additional hard disk to FreeBSD server.

Here goes :

Multiple hard disk slices

1. Create a partition table on the external hard disk :
`$ gpart create -s gpt da0 da0 created`
2. List the partition table :
`$ gpart show da0 => 63 312579603 da0 MBR (149G) 63 312579603 – free – (149G)`
3. Slice the first partition with FreeBSD UFS file system :
`$ gpart add -s 50G -t freebsd-ufs da0 da0p1 added`
4. Slice the second partition with FreeBSD UFS file system :
`$ gpart add -s 50G -t freebsd-ufs da0 da0p2 added`
5. Slice the third partition (remaining hard disk space) with FAT32 file system :
`$ gpart add -t mbr da0 da0p3 added`
6. Format the partitions :
`$ newfs /dev/da0p1a$ newfs /dev/da0p2b$ newfs_msdos -F32 /dev/da0p3`

If you're using the hard disk fully in FreeBSD environment, follow the below steps instead to create a single big slice of the hard disk :

Single(disk space utilized) slice

1. `$ gpart create -s gpt da0 da0 created`
2. `$ gpart show da0 => 63 312579603 da0 GPT (149G) 63 312579603 – free – (149G)`
3. `$ gpart add -t freebsd da0 da0s1 added(take note of the output „da0s1“)`
4. `$ newfs /dev/da0s1`

If the hard disk have existing partition table (equivalent to

GEOM label), the following commands can be used to delete slices & index. The flow is first „Delete“ then „Destroy“.
E.g. Delete the slices first then Destroy the slices, Delete the GEOM label then Destroy the GEOM label (GEOM label is equivalent to partition table). :

Removing slice & index

- Delete the „slices“ (repeat as needed),
\$ gpart delete -i 1 da0s1
- Destroy „slices“ (repeat as needed),
\$ gpart destroy da0s1
- Delete the GEOM label,
\$ gpart delete -i 1 da0
- Destroy the GEOM label,
\$ gpart destroy da0
- On every step, verify the partition table :
\$ gpart show da0and
\$ gpart list
- Step 4 is optional.

Some notes :

1. Device id creation are done by kernel (automatically).
e.g. /dev/da0 ← this device file will be automatically created when the kernel recognize the inserted hard disk
2. The below label are destroyable through the „gpart“ utility :
 - /dev/da0s1 ← by gpart when creation (can be „destroy“)
 - /dev/da0s1a ← by gpart when creation (can be „destroy“)where /dev/da0 is NOT „destroyable“ as it is a device.
3. The MBR partitioning scheme has been around for too long and with limitation. It only allow 2 TB per partition and limit to 4 slices (or partition)
4. Use GPT (guid partition table) partition scheme instead.

It is new with big limits. The GPT partition scheme allows a partition or hard disk size of 9.4 ZB and up to 128 slices (aren't this too many :p)

5. By the way, you'd most likely run into problem if using GPT partition should you run windows. Check it out [here](#). Too bad :p

Further reference can refer to :

man gpart

or

search for [gpart manual pages](#) at [FreeBSD.org](#).

[/expand]