Linux Software RAID: growing filesystems and adding disks

Adding partitions

When new disks are added, existing raid partitions can be grown to use the new disks. After the new disk was partitioned, the RAID level 1/4/5/6 array can be grown for example using this command (assuming that before growing it contains three drives):

```
mdadm --add /dev/md1 /dev/sdb3
mdadm --grow --raid-devices=4 /dev/md1
```

The process can take even 10 hours. There is a critical section at start, which cannot be backed up. To allow recovery after unexpected power failure, an additional option `--backup-file=` can be specified.

Expanding existing partitions

It is possible to migrate the whole array to larger drives (e.g. 250 GB to 1 TB) by replacing one by one. In the end the number of devices will be the same, the data will remain intact, and you will have more space available to you.

Extending an existing RAID array

In order to increase the usable size of the array, you must increase the size of all disks in that array. Depending on the size of your disks, this may take days to complete. It is also important to note that while the array undergoes the resync process, it is vulnerable to irrecoverable failure if another drive were to fail. It would (of course) be a wise idea to completely back up your data before continuing.

First, choose a drive and completely remove it from the array

```
mdadm -f /dev/md0 /dev/sdd1
mdadm -r /dev/md0 /dev/sdd1
```

Next, partition the new drive so that you are using the amount of space you will eventually use on all new disks. For example, if you are going from 100 GB drives to 250 GB drives, you will want to

partition the new 250 GB drive to use 250 GB, not 100 GB. Also, remember to set the partition type to **0xDA** - Non-fs data (or **0xFD**, Linux raid autodetect if you are still using the deprecated autodetect).

```
fdisk /dev/sde
```

Now add the new disk to the array:

```
mdadm --add /dev/md0 /dev/sde1
```

Allow the resync to fully complete before continuing. You will now have to repeat the above steps for *each* disk in your array. Once all of the drives in your array have been replaced with larger drives, we can grow the space on the array by issuing:

```
mdadm --grow /dev/md0 --size=max
```

The array now represents one disk using all of the new available space.

Extending the filesystem

Now that you have expanded the underlying partition, you must now resize your filesystem to take advantage of it. For an ext2/ext3 filesystem:

```
resize2fs /dev/md0
```

For a reiserfs filesystem:

```
resize_reiserfs /dev/md0
```

Please see filesystem documentation for other filesystems.

LVM: Growing the PV

LVM (logical volume manager) abstracts a logical volume (that a filesystem sits on) from the physical disk. If you are used to LVM then you are likely used to growing LVs (logical volumes), but what we grow here is the PV (physical volume) that sits on the *md* device (RAID array).

For further LVM documentation, please see the Linux LVM HOWTO

Growing the physical volume is trivial:

```
pvresize /dev/md0
```

A before-and-after example is:

```
root@barcelona:~# pvdisplay
  --- Physical volume ---
```

```
  PV Name               /dev/md0
  VG Name               server1_vg
  PV Size               931.01 GB / not usable 558.43 GB
  Allocatable           yes
  PE Size (KByte)       4096
  Total PE              95379
  Free PE               42849
  Allocated PE          52530
  PV UUID               BV0mGK-FRtQ-KTLv-aW3I-TllW-Pkiz-3yVPd1

root@barcelona:~# pvresize /dev/md0
  Physical volume "/dev/md0" changed
  1 physical volume(s) resized / 0 physical volume(s) not resized

root@barcelona:~# pvdisplay
  --- Physical volume ---
  PV Name               /dev/md0
  VG Name               server1_vg
  PV Size               931.01 GB / not usable 1.19 MB
  Allocatable           yes
  PE Size (KByte)       4096
  Total PE              238337
  Free PE               185807
  Allocated PE          52530
  PV UUID               BV0mGK-FRtQ-KTLv-aW3I-TllW-Pkiz-3yVPd1
```

The above is the PV part after md0 was grown from ~400GB to ~930GB (a 400GB disk to a 1TB disk). Note the *PV Size* descriptions before and after.

Once the PV has been grown (and hence the size of the VG, volume group, will have increased), you can increase the size of an LV (logical volume), and then finally the filesystem, eg:

```
lvextend -L +50G -n home_lv server1_vg
resize2fs /dev/server1_vg/home_lv
```

The above grows the *home_lv* logical volume in the *server1_vg* volume group by 50GB. It then grows the ext2/ext3 filesystem on that LV to the full size of the LV, as per *Extending the filesystem* above.

Revision #3
Created 2026-04-01 17:11:29 CEST by Philip
Updated 2026-04-13 19:26:16 CEST by Philip