



دانشگاه حکیم سبزواری

دانشکده مهندسی برق و کامپیوتر

مستند سازی پروژه

**RAID**

درس: سیستم عامل

دانشجویان: سیده بیتا امیری – عارفه عمیدیان

استاد مربوطه: خانم دکتر مینا ملک زاده

دی ماه 1402



## فهرست مطالب

|    |                                 |
|----|---------------------------------|
| 4  | مقدمه                           |
| 5  | تکنولوژی RAID چیست؟             |
| 6  | هدف از معماری های مختلف RAID    |
| 7  | انواع ذخیره سازی در RAID        |
| 7  | Mirroring                       |
| 7  | Striping                        |
| 8  | Error Correction                |
| 8  | انواع RAID                      |
| 8  | تکنولوژی RAID 0                 |
| 9  | ایجاد یک آرایه RAID 0           |
| 12 | تکنولوژی RAID 1                 |
| 15 | تکنولوژی RAID 2                 |
| 15 | تکنولوژی RAID 3                 |
| 16 | تکنولوژی RAID 4                 |
| 16 | تکنولوژی RAID 5                 |
| 21 | تکنولوژی RAID 6                 |
| 24 | تکنولوژی RAID 10                |
| 28 | تکنولوژی RAID 50                |
| 28 | ریدهای تودرتو (Nested RAIDs)    |
| 29 | مزایای استفاده از RAID          |
| 29 | معایب استفاده از RAID           |
| 29 | RAID Controller چیست؟           |
| 30 | بهترین نوع RAID کدام است؟       |
| 30 | ابزار مورد نیاز                 |
| 30 | پیش نیازهای لازم                |
| 31 | انجام Process RAID              |
| 33 | درست کردن RAID                  |
| 34 | ساخت Mount Point                |
| 34 | ایجاد فایل سیستم بر روی پارتیشن |
| 35 | Mount کردن Mount Point          |
| 35 | مشاهده نتیجه                    |
| 37 | برگرداندن به حالت اولیه         |
| 38 | منابع                           |

## مقدمه

در این مقاله به بررسی چگونگی RAID چند حافظه می پردازیم.

چه معنایی دارد؟

هدف از RAID چیست؟

انواع RAID کدامند؟

چه پیش نیازهایی میخواهد؟

چه ابزارهایی لازم دارد که باید برای کار استفاده کنیم؟

و در آخر به بررسی خروجی از انجام پروسه RAID Level 6 می پردازیم.



## تکنولوژی RAID چیست؟



اگر بخواهیم بصورت خلاصه Raid را توضیح دهیم باید بگوییم که این واژه مخفف عبارت Redundant Array of Independent Disks است و در بخش Storage ها و منبع ذخیره اطلاعات، مورد استفاده قرار می‌گیرد.

کاربرد RAID ایجاد یک کل واحد از مجموع چند هارد دیسک است. به عبارت دیگر با قرار دادن چند هارد دیسک در کنار هم و پیاده سازی RAID همه هارد دیسک ها به یک مجموعه واحد تبدیل شده و سیستم عامل، همه آن ها را به عنوان یک منبع واحد در نظر می‌گیرد. در تعریف رایید به زبان ساده می‌توان گفت Raid درواقع تکنولوژی مجازی سازی ذخیره دیتا است. برای شناسایی تعداد بسیاری هارد توسط سیستم، باید این هاردها به روشی به هم متصل شوند. این اتصال که به صورت نرم افزاری یا سخت افزاری صورت می‌گیرد، با نام RAID شناخته می‌شود. پس انواع RAID به صورت زیر می‌باشد:

**1- Hardware RAID:** که به صورت سخت افزاری و مستقل از سیستم عامل عمل می‌کند و تمامی عملیات توسط RAID Controller سیستم انجام می‌شود.

**2- Software RAID:** توسط سیستم عامل و نرم افزار پیکربندی و کنترل می‌شود.

تکنولوژی رایید (RAID) دارای سطح بندی های متفاوتی است که بسته به اینکه چه سطحی از RAID پیاده سازی شده باشد، می‌توان از مزایای آن که افزایش کارایی، امنیت اطلاعات و یا تلفیقی از این دو است بهره برداری نمود. از کاربرد های این تکنولوژی می‌توان در انواع وب

سرور، کامپیوتر های شخصی و مواردی از این قبیل نام برد که نیاز به افزایش کارایی و سرعت پردازش بالا دارند.

عملیات RAID کردن هاردها از حالت بسیار ساده و ابتدایی در محیط ویندوز و از طریق پنجره Disk Management تا مرحله حرفه ای و با استفاده از کارت های HBA می تواند صورت بپذیرد اما در تمام این روش ها، مسئله ساختار کلی RAID می باشد که در هاردها صورت می گیرد. اولین حالت ساده و ابتدایی RAID در ویندوز و در پنجره Disk Management، حالت چند هارد را به Dynamic تغییر داده و با انتخاب تمام این هاردها، می توان آنها را به صورت RAID تنظیم کرد که در این حالت تنها دو مدل Strip و Spanned را به عنوان حالت های RAID خواهیم داشت.

نکته بسیار مهم در این مرحله این است که در صورت تغییر هر هاردی به حالت Dynamic و در کل در صورت تغییر RAID هارد، تمام اطلاعات شما از روی آن پاک خواهد شد.

## هدف از معماری های مختلف RAID

### • افزایش امنیت

از آنجایی که اطلاعات کاربران بر روی دو یا بیشتر از دو هارد ذخیره می شود، اگر برای اطلاعات یا دیسک مشکلی ایجاد شود همچنان از طریق سایر دیسک ها به اطلاعات دسترسی خواهیم داشت و این قابلیت نشان دهنده ی امنیت در RAID است.

### • افزایش کارایی داده های ورودی و خروجی

در استفاده از RAID امکان این برای کنترلر وجود دارد که داده ها را به دو قسمت تقسیم کند و هر یک از بخش ها را در یک هارد قرار دهد و با این کار توانایی دسترسی به داده ها برای خواندن و نوشتن دو برابر می شود.

### • افزایش ظرفیت

در سیستم RAID این امکان وجود دارد که دو یا چند هارد دیسک با هم ترکیب شوند و یک حافظه ی قوی برای ذخیره سازی داده ها ایجاد کنند.



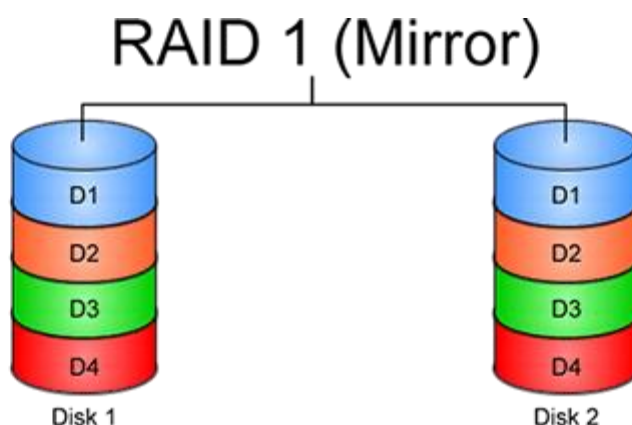
## انواع ذخیره سازی در RAID

RAID در واقع آرایه ای است که از دو دیسک سخت و یا بیشتر که در روی یک واحد قرار گرفته اند تشکیل یافته است. برای اینکه به آرایه RAID سه ویژگی بیشتر در مقایسه با JBOD داده شود سه مفهوم اساسی استفاده می شوند.

- Mirroring
- Striping
- Error correction

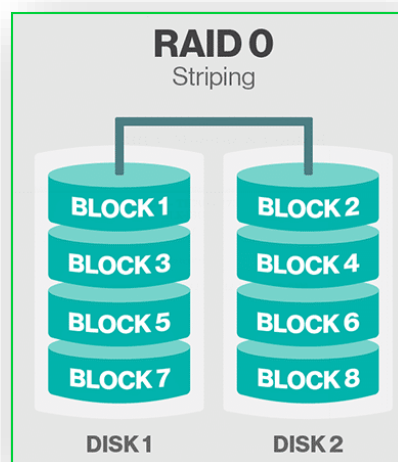
### Mirroring

مفهوم Mirroring یعنی داده ها در بیش از یک دیسک نوشته شوند. مثال اساسی Mirroring آرایه RAID1 با دو دیسک می باشد. هر دو دیسک محتوای یکسانی دارند. زمانی که برای درایو اولی مشکلی پیش بیاید، عملیات خواندن و نوشتن به شکل مستقیم در دیسک دوم انجام می شود. عملیات خواندن در آرایه های Mirroring در مقایسه با دیسک تنها به دلیل اینکه سیستم، داده ها را از چندین دیسک و در یک زمان می خواند سریع تر است. اگرچه عملیات نوشتن از آنجایی که داده ها باید در چند دیسک به جای یک دیسک نوشته شوند، کند تر می باشد. بازسازی آرایه دیسک mirror شده بسیار آسان است. داده ها از دیسک سالم به دیسک جدید کپی می شوند. در طول بازسازی کارایی خواندن کاهش پیدا می کند. چرا که تنها یک دیسک mirror قابل استفاده می باشد.



### Striping

عملیات جداسازی داده ها از چندین دیسک است. برای مثال آرایه RAID 0 داده ها را در دو دیسک قرار می دهد که این کار تنها کارایی را افزایش می دهد. عملیات خواندن و نوشتن در آرایه ی Striping در مقایسه با دیسک های تکی سریع تر می باشد.



## Error correction

داده های parity را با روشی که به آنها اجازه یافته شدن و احتمال تصحیح مشکلات را بدهد بر روی دیسک ذخیره می کند. RAID 5 مثال خوبی از مکانیزم تصحیح خطا است. مثلاً آرایه RAID 5 از سه نوار داده در دو دیسک اول تشکیل یافته و parity را در دیسک سوم قرار می دهد. مکانیزم تصحیح خطا باعث کاهش کارایی خصوصاً در عملیات نوشتن می شود چراکه هم داده و هم parity باید نوشته شود.

طراحی شمای raid شامل حفاظت از داده و کارایی آن می شود. با توجه به نیازهای سرور شما باید پیکربندی مناسب RAID را انتخاب نمایید.

## انواع RAID

همانطور که در بالا اشاره شد، تکنولوژی RAID انواع و سطح بندی متفاوتی دارد که در ادامه انواع رید را به شما معرفی خواهیم کرد:

- Raid استاندارد
- Raid ترکیبی
- Raid غیر استاندارد

در ادامه با معرفی هر raid با تفاوت raid ها آشنا می شوید:

### تکنولوژی RAID 0 چیست ؟

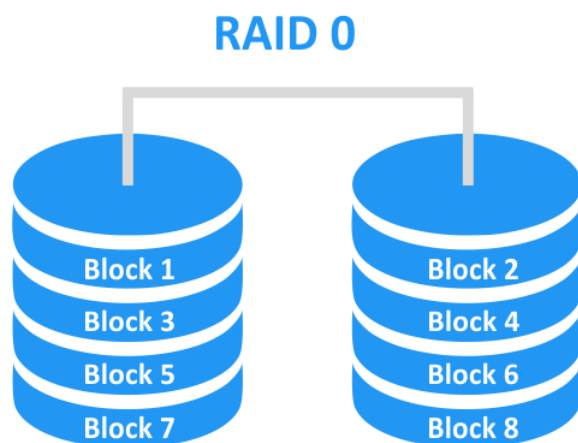
در تعریف raid و انواع آن ابتدا به بررسی raid 0 می پردازیم، RAID 0 اولین و سادهترین نوع RAID می باشد که به صورت معمول بر روی سیستم عامل ها وجود دارد. در صورتی که در



Disk Management سیستم خود دو هارد را با هم به صورت Dynamic و Strip متصل کرده باشید در اصل از این حالت RAID استفاده نموده اید. همانطور که گفته شد دو حالت Stripped و Spanned از انواع RAID 0 می باشد. در این حالت تمام هاردها بدون هیچ هارد رزرو و جایگزین، به صورت متوالی به هم متصل می شوند. سرعت در این مدل RAID بسیار بالا می باشد و تقریباً از مجموع تمام حجم هاردها می توان استفاده نمود. این مدل RAID باید حداقل دو عدد هارد بر روی سیستم یا Storage وجود داشته باشد.

در این روش امکان تحمل خطایذیری و بازیابی داده ها وجود ندارد و در صورت خرابی یکی از دیسک ها داده ها از بین خواهد رفت.

آرایه RAID 0 با تقسیم کردن داده ها به قطعات و جدا کردن آن در دیسک های موجود کار می کند. این بدان معنی است که هر دیسک حاوی بخشی از داده ها است و هنگام بازیابی اطلاعات به چندین دیسک ارجاع داده می شود.



### ایجاد یک آرایه RAID 0

شرایط مورد نیاز: حداقل 2 دستگاه ذخیره سازی.

مزیت اصلی: عملکرد از نظر خواندن/نوشتن و ظرفیت

مواردی که باید در نظر داشته باشید: مطمئن شوید که پشتیبان گیری کاربردی دارید. خرابی یک دستگاه تمام داده های آرایه را از بین می برد.

برای شروع، شناسه های دیسک های خامی را که استفاده می کنید پیدا کنید:

```
$ lsblk -o NAME,SIZE,FSTYPE,TYPE,MOUNTPOINT
```

```

trinity@kali: ~
File Actions Edit View Help
Processing triggers for kali-menu (2023.4.5) ...

(trinity@kali)~[~]
$ mdadm --create
mdadm: an md device must be given in this mode

(trinity@kali)~[~]
$ lsblk -o NAME,SIZE,FSTYPE,TYPE,MOUNTPOINT
NAME        SIZE  FSTYPE  TYPE  MOUNTPOINT
sda          953.9G disk
├─sda1       500M ntfs    part
├─sda2      136.2G ntfs    part
├─sda3      197.3G ntfs    part  /media/trinity/New Volume
├─sda4         1K      part
├─sda5      275.4G ntfs    part  /media/trinity/New Volume1
├─sda6      266.4G ntfs    part
├─sda7       77.2G ext4    part  /
├─sda8       971M swap    part  [SWAP]
sdb          58.6G disk
├─sdb1       58.6G exfat   part  /media/trinity/92A2-81CB
sr0          1024M  rom

```

در این مثال، شما دو دیسک بدون سیستم فایل دارید که اولی دارای 953.9 گیگ و دومی دارای 58.6 ظرفیت هستند. به این دستگاه ها شناسه های `dev/sda/` و `dev/sdb/` برای این جلسه داده شده است و اجزای خام مورد استفاده برای ساخت آرایه خواهند بود.

برای ایجاد یک آرایه RAID 0 با این اجزا، آنها را به دستور `mdadm --create` منتقل کنید. شما باید نام دستگاهی را که می خواهید ایجاد کنید، سطح RAID و تعداد دستگاه ها را مشخص کنید. در این دستور، نام دستگاه را `dev/md0/` است و دو دیسکی که آرایه را می سازند در کد زیر داریم:

```
$ sudo mdadm --create --verbose /dev/md0 --level=0 --raid-devices=2 /dev/sda /dev/sdb
```

با بررسی فایل `proc/mdstat/` تأیید کنید که RAID با موفقیت ایجاد شده است:

```
$ cat /proc/mdstat
```

خروجی به صورت زیر است:

```

Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md0 : active raid0 sdb[1] sda[0]
      209584128 blocks super 1.2 512k chunks

unused devices: <none>

```

این خروجی نشان می دهد که دستگاه `dev/md0/` در پیکربندی RAID 0 با استفاده از دستگاه های `dev/sda/` و `dev/sdb/` ایجاد شده است.

## ایجاد و نصب فایل سیستم

بعد، یک فایل سیستم روی آرایه ایجاد کنید:

```
$ sudo mkfs.ext4 -F /dev/md0
```

سپس، یک نقطه اتصال برای پیوست کردن فایل سیستم جدید ایجاد کنید:

```
$ sudo mkdir -p /mnt/md0
```

با دستور زیر می توانید فایل سیستم را Mount کنید:

```
$ sudo mount /dev/md0 /mnt/md0
```

پس از آن، بررسی کنید که آیا فضای جدید موجود است:

```
$ df -h -x devtmpfs -x tmpfs
```

خروجی به صورت زیر خواهد بود:

| Filesystem | Size | Used | Avail | Use% | Mounted on |
|------------|------|------|-------|------|------------|
| /dev/vda1  | 25G  | 1.4G | 23G   | 6%   | /          |
| /dev/vda15 | 105M | 3.4M | 102M  | 4%   | /boot/efi  |
| /dev/md0   | 196G | 61M  | 186G  | 1%   | /mnt/md0   |

فایل سیستم جدید اکنون نصب شده و قابل دسترسی است.

## ذخیره چیدمان آرایه

برای اطمینان از اینکه آرایه به طور خودکار در هنگام بوت جمع می شود، باید فایل `/etc/mdadm/mdadm.conf` را تنظیم کنید. شما می توانید به طور خودکار آرایه فعال را اسکن کرده و فایل را با موارد زیر اضافه کنید:

```
$ sudo mdadm --detail --scan | sudo tee -a /etc/mdadm/mdadm.conf
```

پس از آن، می توانید `initramfs` یا سیستم فایل `RAM` اولیه را به روزرسانی کنید تا آرایه در طول فرآیند بوت اولیه در دسترس باشد:

```
$ sudo update-initramfs -u
```

برای نصب خودکار در هنگام بوت، گزینه های نصب فایل سیستم جدید را به فایل `/etc/fstab` اضافه کنید:

```
$ echo '/dev/md0 /mnt/md0 ext4 defaults,nofail,discard 0 0' | sudo tee -a /etc/fstab
```

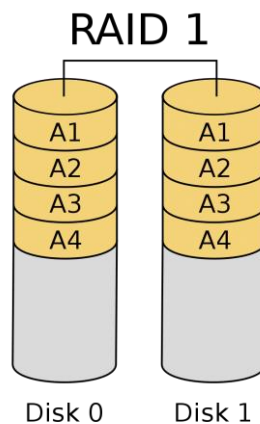
آرایه `RAID 0` شما اکنون به طور خودکار هر بوت را جمع و سوار می کند.

اکنون راه اندازی `RAID` خود را به پایان رسانده اید. اگر می خواهید یک `RAID` دیگر را امتحان کنید، دستورالعمل های بازنشانی در انتهای سطوح قرار دارد را دنبال کنید تا با ایجاد یک نوع آرایه `RAID` جدید ادامه دهید.

## تکنولوژی raid 1

این مدل تقریباً یکی از امن ترین انواع raid در سرورها می باشد. در مدل RAID 1 که حتماً تعداد زوج هارد باید بر روی سیستم یا Storage موجود باشد، نیمی از هاردها به صورت هاردهای رزرو آنلاین برای نیمی دیگر از آنها در نظر گرفته می شوند. بدین صورت که در هنگام نوشتن یک فایل بر روی هارد اصلی، یک کپی به عنوان پشتیبان نیز بر روی هارد رزرو نوشته می شود. در این حالت سرعت RAID نصف حالت معمول است زیرا عملیات نوشتن دو بار صورت می گیرد. اما در صورتی که به هر دلیلی هاردی کامل از بین برود، یک هارد پشتیبان از آن در دستگاه وجود داشته و می توان بدون هیچ مشکل یا قطعی در سیستم، از هارد رزرو استفاده کرد. در مراکز مهم که امنیت بسیار اهمیت بالاتری نسبت به سرعت دارد، از این نوع RAID در انواع هاردهای سیستم ها استفاده می شود (دو مدل اول معمولاً بر روی Workstation ها اعمال می شود و معمولاً در Storage ها به دلیل مشکلات موجود استفاده نمی شوند).

نوع آرایه RAID 1 با انعکاس داده ها در تمام دیسک های موجود پیاده سازی می شود. هر دیسک در یک آرایه RAID 1 یک کپی کامل از داده ها را دریافت می کند و در صورت خرابی دستگاه، افزونگی را ارائه می دهد.



شرایط مورد نیاز: حداقل 2 دستگاه ذخیره سازی.  
مزیت اصلی: افزونگی بین دو دستگاه ذخیره سازی.  
مواردی که باید در نظر داشته باشید: از آنجایی که دو نسخه از داده ها نگهداری می شود، تنها نیمی از فضای دیسک قابل استفاده خواهد بود.

برای شروع، شناسه های دیسک های خامی را که استفاده می کنید پیدا کنید:

```
$ lsblk -o NAME,SIZE,FSTYPE,TYPE,MOUNTPOINT
```

| NAME    | SIZE  | FSTYPE  | TYPE | MOUNTPOINT |
|---------|-------|---------|------|------------|
| sda     | 100G  | disk    |      |            |
| sdb     | 100G  | disk    |      |            |
| vda     | 25G   | disk    |      |            |
| └─vda1  | 24.9G | ext4    | part | /          |
| └─vda14 | 4M    |         | part |            |
| └─vda15 | 106M  | vfat    | part | /boot/efi  |
| vdb     | 466K  | iso9660 | disk |            |

در این مثال، شما دو دیسک بدون سیستم فایل دارید که هر کدام دارای 100 گیگ ظرفیت هستند. برای این جلسه داده شده است و اجزای `/dev/sda` و `/dev/sdb` به این دستگاه ها شناسه های خام مورد استفاده برای ساخت آرایه خواهند بود. برای ایجاد یک آرایه RAID 1 با این اجزا، آنها را به دستور `mdadm --create` منتقل کنید. شما باید نام دستگاهی را که می خواهید ایجاد کنید، سطح RAID و تعداد دستگاه ها را مشخص کنید. در این دستور، نام دستگاه را `/dev/md0` است و دو دیسکی که آرایه را می سازند در کد زیر داریم:

```
$ sudo mdadm --create --verbose /dev/md0 --level=1 --raid-devices=2 /dev/sda /dev/sdb
```

اگر دستگاه های مؤلفه ای که استفاده می کنید پارتیشن هایی با `boot flag` فعال نیستند، احتمالاً هشدار زیر را دریافت خواهید کرد. بی خطر است که با `y` پاسخ دهید و ادامه دهید:

```
mdadm: Note: this array has metadata at the start and
may not be suitable as a boot device. If you plan to
store '/boot' on this device please ensure that
your boot-loader understands md/v1.x metadata, or use
--metadata=0.90
mdadm: size set to 104792064K
Continue creating array? y
```

ابزار `mdadm` شروع به انعکاس درایوها می کند. این ممکن است کمی طول بکشد، اما آرایه را می توان در این مدت استفاده کرد. می توانید با بررسی فایل `/proc/mdstat`، پیشرفت آینه سازی را کنترل کنید:

```
$ cat /proc/mdstat
```

```
Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md0 : active raid1 sdb[1] sda[0]
104792064 blocks super 1.2 [2/2] [UU]
[====>.....] resync = 20.2% (21233216/104792064) finish=6.9min speed=199507K/sec

unused devices: <none>
```

در اولین هایلایت، دستگاه **/dev/md0** در پیکربندی RAID 1 با استفاده از دستگاه های **/dev/sdb** و **/dev/sda** ایجاد شد. دومین هایلایت پیشرفت در آینه کاری را نشان می دهد. تا زمانی که این فرآیند کامل شد، می توانید به مرحله بعدی ادامه دهید.

## ایجاد و نصب فایل سیستم

بعد، یک فایل سیستم روی آرایه ایجاد کنید:

```
$ sudo mkfs.ext4 -F /dev/md0
```

سپس، یک نقطه اتصال برای پیوست کردن فایل سیستم جدید ایجاد کنید:

```
$ sudo mkdir -p /mnt/md0
```

کنید: Mount با دستور زیر می توانید فایل سیستم را

```
$ sudo mount /dev/md0 /mnt/md0
```

پس از آن، بررسی کنید که آیا فضای جدید موجود است:

```
$ df -h -x devtmpfs -x tmpfs
```

خروجی به صورت زیر خواهد بود:

| Filesystem | Size | Used | Avail | Use% | Mounted on |
|------------|------|------|-------|------|------------|
| /dev/vda1  | 25G  | 1.4G | 23G   | 6%   | /          |
| /dev/vda15 | 105M | 3.4M | 102M  | 4%   | /boot/efi  |
| /dev/md0   | 196G | 61M  | 186G  | 1%   | /mnt/md0   |

فایل سیستم جدید اکنون نصب شده و قابل دسترسی است.

## ذخیره چیدمان آرایه

برای اطمینان از اینکه آرایه به طور خودکار در هنگام بوت جمع می شود، باید فایل **/etc/mdadm/mdadm.conf** را تنظیم کنید. شما می توانید به طور خودکار آرایه فعال را اسکن کرده و فایل را با موارد زیر اضافه کنید:

```
$ sudo mdadm --detail --scan | sudo tee -a /etc/mdadm/mdadm.conf
```

پس از آن، می توانید **initramfs** یا سیستم فایل **RAM** اولیه را به روزرسانی کنید تا آرایه در طول فرآیند بوت اولیه در دسترس باشد:

```
$ sudo update-initramfs -u
```

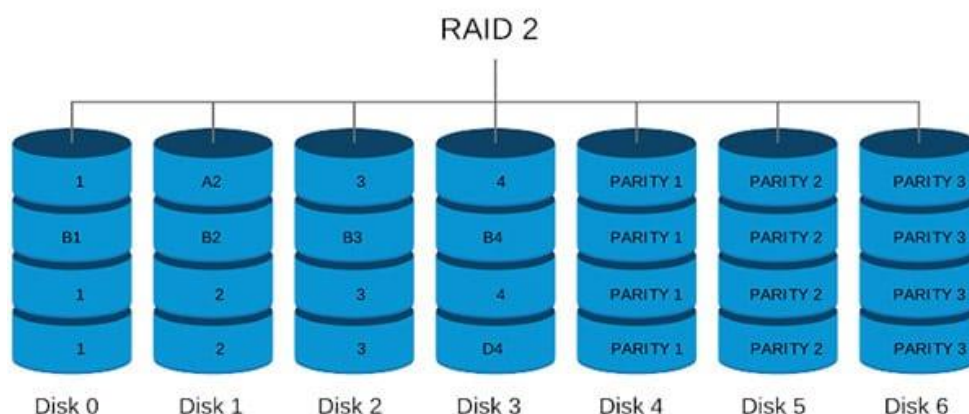
برای نصب خودکار در هنگام بوت، گزینه های نصب فایل سیستم جدید را به فایل **/etc/fstab** اضافه کنید:

```
$ echo '/dev/md0 /mnt/md0 ext4 defaults,nofail,discard 0 0' | sudo tee -a /etc/fstab
```

آرایه RAID 1 شما اکنون به طور خودکار هر بوت را مونتاژ و سوار می کند. اکنون راه اندازی RAID خود را به پایان رسانده اید. اگر می خواهید یک RAID دیگر را امتحان کنید، دستورالعمل های بازنشانی در انتهای سطوح قرار دارد را دنبال کنید تا با ایجاد یک نوع آرایه RAID جدید ادامه دهید.

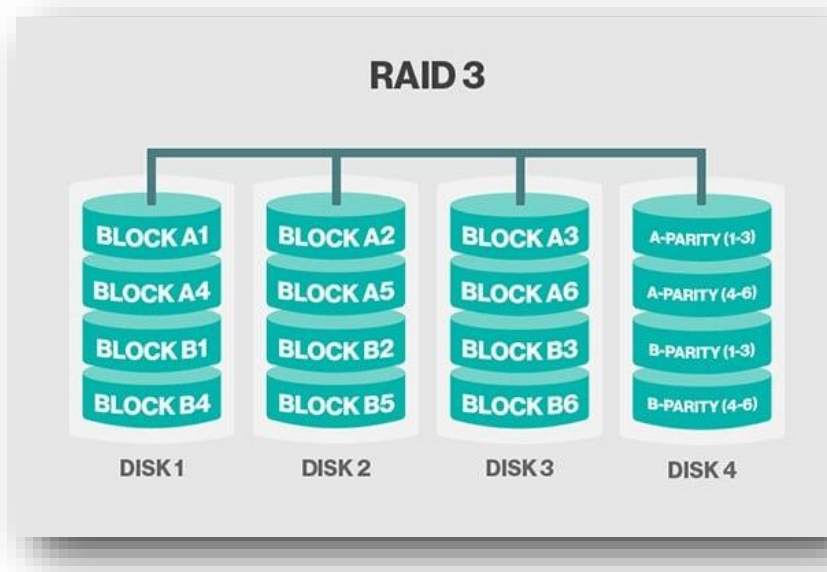
## تکنولوژی raid 2

تکنولوژی raid 2 تکنیکی است که از کد Hamming برای تصحیح خطاها استفاده می کند. Raid 2 از نواربندی در دیسک ها و برای تامین امنیت داده از ECC استفاده می کند. تکنولوژی RAID 2 به دلیل هزینه بالا و اجرای سخت هم اکنون منسوخ شده است. این رید از نوع ریدهای غیر استاندارد است.



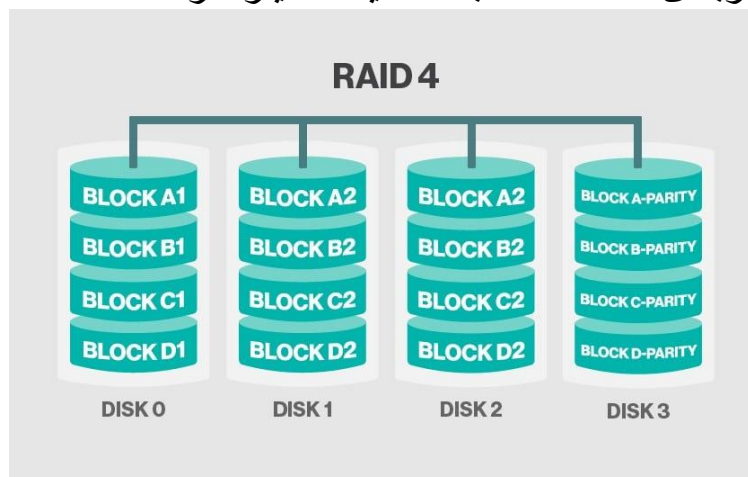
## تکنولوژی raid 3

Raid 3 یکی دیگر از انواع raid است که در آن از نواربندی اطلاعات استفاده می شود، و یک هارد دیسک را برای ذخیره اطلاعات parity اختصاص داده است. با استفاده از ECC امکان تشخیص خطاها در raid 3 امکان پذیر می باشد.



### تکنولوژی raid 4

RAID 4 شباهت زیادی به RAID 3 دارد. تفاوت اصلی در نحوه به اشتراک گذاری داده‌ها می‌باشد. این Raid از نوارهای بزرگ استفاده می‌کند. بدین معنی که شما می‌توانید هر رکوردی را از هر درایوی بخوانید. این توزیع به طور چشم‌گیری عملکرد را افزایش می‌دهد. RAID 4 برای پیاده سازی و پیکربندی کامل حداقل به سه دیسک نیاز دارد.



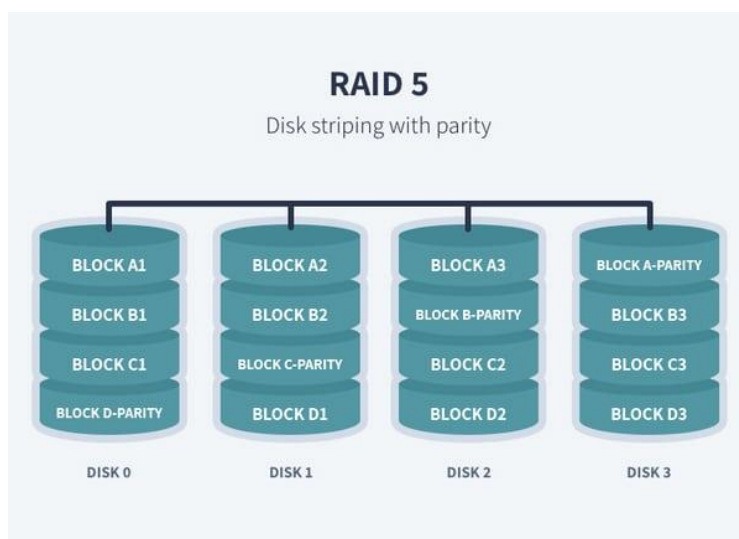
### تکنولوژی raid 5

در مدل RAID 5 حداقل به ۳ دیسک نیاز است. همانند RAID 1 اطلاعات به صورت Mirroring در دیسک‌ها ذخیره می‌شود. همچنین یک کد parity در بین دیسک‌ها پخش می‌شود که باعث افزایش کارایی می‌گردد و در هنگام بروز مشکل می‌توان اطلاعات از دست رفته را توسط آن بازیابی کرد. در مقایسه با RAID 1 این سطح، سرعت کمتری در نوشتن اطلاعات دارد. زیرا زمانی هم برای نوشتن اطلاعات parity مورد نیاز است. در حال حاضر یکی از پرکاربردترین مدل‌های RAID در انواع Storage ها همین حالت RAID 5 می‌باشد.



و اکثر قریب به اتفاق Storage ها و برخی از سیستم‌های Workstation خانگی یا اداری از این مدل برای شناسایی تعداد بالای هاردها استفاده می‌کند. در این روش داده‌ها همانند روش Striping به صورت تکه‌تکه شده بین دیسک‌ها تقسیم و ذخیره می‌شود با این تفاوت که علاوه بر داده‌های ذخیره شده نوعی داده تحت عنوان Parity نیز در دیسک‌ها ذخیره می‌گردد.

در این روش سرعت خواندن/نوشتن افزایش می‌یابد و همچنین بدلیل استفاده از Parity امکان تحمل خط‌پذیری و بازیابی داده‌ها در زمان خرابی یکی از دیسک‌ها وجود دارد. نوع آرایه RAID 5 با نوار کردن داده‌ها در دستگاه‌های موجود پیاده‌سازی می‌شود. یک جزء از هر نوار یک بلوک برابری محاسبه شده است. اگر دستگاهی از کار بیفتد، بلوک برابری و بلوک‌های باقی‌مانده می‌توانند برای محاسبه داده‌های از دست رفته استفاده شوند. دستگاهی که بلوک برابری را دریافت می‌کند به گونه‌ای می‌چرخد که هر دستگاه دارای مقدار متعادلی از اطلاعات برابری باشد.



گونه‌ای از دیسک‌ها به نام Hot spare نیز وجود دارد که به عنوان پشتیبانی برای آرایه RAID به کار می‌روند. این دیسک‌ها به صورت فیزیکی در آرایه نصب شده و تا زمانی که دیسک‌های فعال از کار بیفتند غیر فعال هستند. کنترلر RAID به صورت اتوماتیک درایوهای fail را با دیسک spare تعویض می‌کند و شروع به ایجاد مجدد فرآیند‌ها برای آرایه می‌نماید. برای مثال آرایه RAID 5 با یک دیسک Hot spare مساوی تعداد دیسک‌های RAID 6 بوده و از نظر حفاظت در هنگام به وجود آمدن خطا تسهیلات بیشتری را فراهم می‌کند.

شرایط مورد نیاز: حداقل 3 دستگاه ذخیره‌سازی.  
مزیت اصلی: افزونگی با ظرفیت قابل استفاده بیشتر.

مواردی که باید در نظر داشته باشید: در حالی که اطلاعات برابری (Parity) توزیع می شود، از ظرفیت یک دیسک برای برابری استفاده می شود. RAID 5 می تواند از عملکرد بسیار ضعیف در حالت تخریب رنج ببرد.

برای شروع، شناسه های دیسک های خامی را که استفاده می کنید پیدا کنید:

**\$ lsblk -o NAME,SIZE,FSTYPE,TYPE,MOUNTPOINT**

```
NAME    SIZE FSTYPE  TYPE MOUNTPOINT
sda     100G   disk
sdb     100G   disk
sdc     100G   disk
vda     25G    disk
├─vda1  24.9G ext4   part /
├─vda14  4M     part
└─vda15 106M vfat   part /boot/efi
vdb     466K iso9660 disk
```

در این مثال، شما سه دیسک بدون سیستم فایل دارید که هر کدام دارای 100 گیگ ظرفیت هستند. به این دستگاه ها شناسه های **/dev/sda** و **/dev/sdb** و **/dev/sdc** برای این جلسه داده شده است و اجزای خام مورد استفاده برای ساخت آرایه خواهند بود. برای ایجاد یک آرایه RAID 5 با این اجزا، آنها را به دستور **mdadm --create** منتقل کنید. شما باید نام دستگاهی را که می خواهید ایجاد کنید، سطح RAID و تعداد دستگاه ها را مشخص کنید. در این دستور، نام دستگاه را **/dev/md0** است و دو دیسکی که آرایه را می سازند در کد زیر داریم:

**\$ sudo mdadm --create --verbose /dev/md0 --level=5 --raid-devices=3  
/dev/sda /dev/sdb /dev/sdc**

ابزار **mdadm** شروع به پیکربندی آرایه می کند. از فرآیند بازیابی برای ساخت آرایه به دلایل عملکرد استفاده می کند. این ممکن است کمی طول بکشد، اما آرایه را می توان در این مدت استفاده کرد. می توانید با بررسی فایل **/proc/mdstat**، پیشرفت آینه سازی را کنترل کنید:

**\$ cat /proc/mdstat**

خروجی این دستور به صورت زیر است:

```
Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md0 : active raid5 sdc[3] sdb[1] sda[0]
      209582080 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/2] [UU_]
      [>.....] recovery = 0.9% (957244/104791040) finish=18.0min speed=95724K/sec
```

unused devices: <none>

در اولین هایلایت، دستگاه `/dev/md0` در پیکربندی RAID 5 با استفاده از دستگاه های `/dev/sda`، `/dev/sdb` و `/dev/sdc` ایجاد شد. دومین هایلایت پیشرفت ساخت را نشان می دهد.

**هشدار:** با توجه به روشی که `mdadm` آرایه های RAID 5 را می سازد، در حالی که آرایه هنوز در حال ساخت است، تعداد قطعات یدکی در آرایه به صورت نادرست گزارش می شود. این بدان معناست که قبل از به روزرسانی فایل `/etc/mdadm/mdadm.conf` باید منتظر بمانید تا آرایه مونتاژ شود. اگر فایل پیکربندی را در حالی که آرایه هنوز در حال ساخت است به روز کنید، سیستم اطلاعات نادرستی در مورد وضعیت آرایه خواهد داشت و نمی تواند آن را به طور خودکار در هنگام بوت با نام صحیح جمع کند. می توانید تا زمانی که این فرآیند کامل شد، راهنما را ادامه دهید.

### ایجاد و نصب فایل سیستم

بعد، یک فایل سیستم روی آرایه ایجاد کنید:

```
$ sudo mkfs.ext4 -F /dev/md0
```

سپس، یک نقطه اتصال برای پیوست کردن فایل سیستم جدید ایجاد کنید:

```
$ sudo mkdir -p /mnt/md0
```

با دستور زیر می توانید فایل سیستم را Mount کنید:

```
$ sudo mount /dev/md0 /mnt/md0
```

پس از آن، بررسی کنید که آیا فضای جدید موجود است:

```
$ df -h -x devtmpfs -x tmpfs
```

خروجی به صورت زیر خواهد بود:

| Filesystem | Size | Used | Avail | Use% | Mounted on |
|------------|------|------|-------|------|------------|
| /dev/vda1  | 25G  | 1.4G | 23G   | 6%   | /          |
| /dev/vda15 | 105M | 3.4M | 102M  | 4%   | /boot/efi  |
| /dev/md0   | 196G | 61M  | 186G  | 1%   | /mnt/md0   |

فایل سیستم جدید اکنون نصب شده و قابل دسترسی است.

### ذخیره چیدمان آرایه

برای اطمینان از اینکه آرایه به طور خودکار در هنگام بوت جمع می شود، باید فایل `/etc/mdadm/mdadm.conf` را تنظیم کنید.

**هشدار:** همانطور که قبلاً ذکر شد، قبل از اینکه پیکربندی را تنظیم کنید، دوباره بررسی کنید تا مطمئن شوید که آرایه مونتاژ شده است. انجام مراحل زیر قبل از ساخته شدن آرایه از مونتاژ صحیح آرایه سیستم در راه اندازی مجدد جلوگیری می کند.

می‌توانید با بررسی فایل `/proc/mdstat`، پیشرفت آینه‌سازی را کنترل کنید:

```
$ cat /proc/mdstat
```

خروجی به صورت زیر خواهد بود:

```
Personalities : [raid1] [linear] [multipath] [raid0] [raid6] [raid5] [raid4] [raid10]
md0 : active raid5 sdc[3] sdb[1] sda[0]
      209584128 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/3] [UUU]

unused devices: <none>
```

این خروجی نشان می‌دهد که بازسازی کامل شده است. اکنون، می‌توانید به طور خودکار آرایه فعال را اسکن کرده و فایل را اضافه کنید:

```
$ sudo mdadm --detail --scan | sudo tee -a /etc/mdadm/mdadm.conf
```

پس از آن، می‌توانید `initramfs` یا سیستم فایل `RAM` اولیه را به‌روزرسانی کنید تا آرایه در طول فرآیند بوت اولیه در دسترس باشد:

```
$ sudo update-initramfs -u
```

برای نصب خودکار در هنگام بوت، گزینه‌های نصب فایل سیستم جدید را به فایل `/etc/fstab` اضافه کنید:

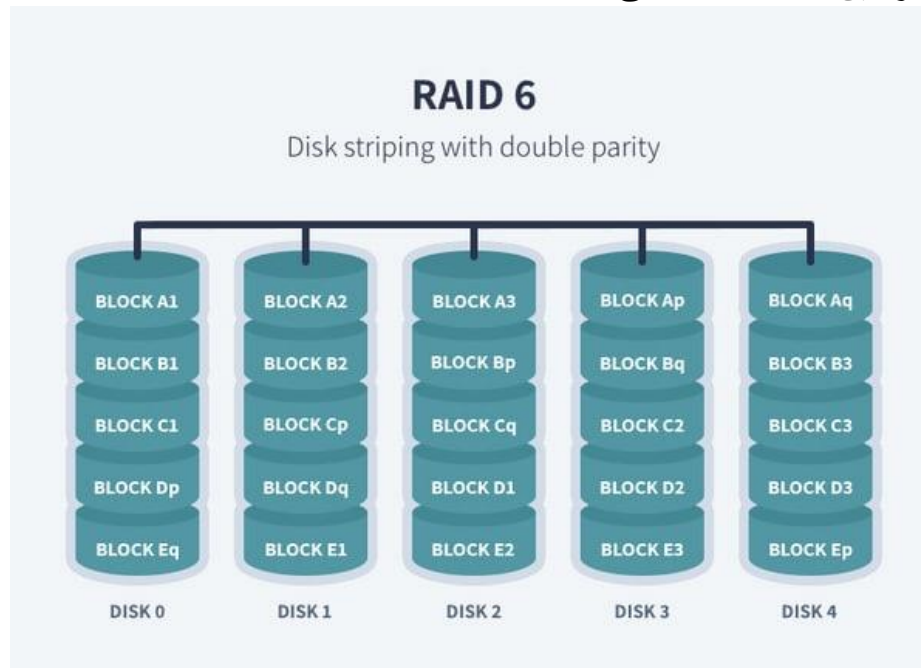
```
$ echo '/dev/md0 /mnt/md0 ext4 defaults,nofail,discard 0 0' | sudo tee
-a /etc/fstab
```

آرایه `RAID 5` شما اکنون به طور خودکار هر بوت را مونتاژ و سوار می‌کند.

اکنون راه اندازی `RAID` خود را به پایان رسانده اید. اگر می‌خواهید یک `RAID` دیگر را امتحان کنید، دستورالعمل‌های بازنشانی در انتهای سطوح قرار دارد را دنبال کنید تا با ایجاد یک نوع آرایه `RAID` جدید ادامه دهید.

## تکنولوژی raid 6

این مدل نیز مانند RAID 5 می‌باشد با این تفاوت که تعداد هارد رزرو آفلاین در مدل RAID 6 به دو عدد افزایش یافته است. این کار باعث افزایش ضریب امنیت در این مدل RAID نسبت به مدل RAID 5 شده است. این مدل از جدیدترین مدل‌های RAID می‌باشد و به همین خاطر با وجود امنیت و سرعت بالاتر نسبت به RAID 5، همچنان تعداد کمتری از Storageها در اقصی نقاط دنیا از این مدل استفاده می‌کنند.



شرایط مورد نیاز: حداقل 4 دستگاه ذخیره سازی.

مزیت اصلی: افزونگی مضاعف با ظرفیت قابل استفاده بیشتر.

مواردی که باید در نظر داشته باشید: در حالی که اطلاعات parity توزیع می‌شود، ظرفیت دو دیسک برای آن استفاده می‌شود. RAID 6 می‌تواند از عملکرد بسیار ضعیف در حالت تخریب رنج ببرد.

برای شروع، شناسه‌های دیسک‌های خامی را که استفاده می‌کنید پیدا کنید:

```
$ lsblk -o NAME,SIZE,FSTYPE,TYPE,MOUNTPOINT
```

| NAME | SIZE | FSTYPE | TYPE | MOUNTPOINT |
|------|------|--------|------|------------|
| sda  | 100G | disk   |      |            |
| sdb  | 100G | disk   |      |            |

```
sdc 100G disk
sdd 100G disk
vda 25G disk
├─vda1 24.9G ext4 part /
├─vda14 4M part
└─vda15 106M vfat part /boot/efi
vdb 466K iso9660 disk
```

در این مثال، شما چهار دیسک بدون سیستم فایل دارید که هر کدام دارای 100 گیگ ظرفیت هستند. به این دستگاه ها شناسه های `/dev/sda` و `/dev/sdb` و `/dev/sdc` و `/dev/sdd` برای این جلسه داده شده است و اجزای خام مورد استفاده برای ساخت آرایه خواهند بود. برای ایجاد یک آرایه RAID 6 با این اجزا، آنها را به دستور `mdadm --create` منتقل کنید. شما باید نام دستگاهی را که می خواهید ایجاد کنید، سطح RAID و تعداد دستگاه ها را مشخص کنید. در این دستور، نام دستگاه را `/dev/md0` است و دو دیسکی که آرایه را می سازند در کد زیر داریم:

```
$ sudo mdadm --create --verbose /dev/md0 --level=6 --raid-devices=4
/dev/sda /dev/sdb /dev/sdc /dev/sdd
```

ابزار `mdadm` شروع به پیکربندی آرایه می کند. از فرآیند بازیابی برای ساخت آرایه به دلایل عملکرد استفاده می کند. این ممکن است کمی طول بکشد، اما آرایه را می توان در این مدت استفاده کرد. می توانید با بررسی فایل `/proc/mdstat`، پیشرفت آینده سازی را کنترل کنید:

```
$ cat /proc/mdstat
```

خروجی دستور بالا به صورت زیر است:

```
Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md0 : active raid6 sdd[3] sdc[2] sdb[1] sda[0]
209584128 blocks super 1.2 level 6, 512k chunk, algorithm 2 [4/4] [UUUU]
[>.....] resync = 0.6% (668572/104792064) finish=10.3min speed=167143K/sec

unused devices: <none>
```

در اولین هایلایت، دستگاه `/dev/md0` در پیکربندی RAID 6 با استفاده از `/dev/sda`، `/dev/sdb`، `/dev/sdc` و `/dev/sdd` ایجاد شده است. دومین خط برجسته پیشرفت ساخت را نشان می دهد. می توانید تا زمانی که این فرآیند کامل شد، راهنما را ادامه دهید.

## ایجاد و نصب فایل سیستم

بعد، یک فایل سیستم روی آرایه ایجاد کنید:

```
$ sudo mkfs.ext4 -F /dev/md0
```

سپس، یک نقطه اتصال برای پیوست کردن فایل سیستم جدید ایجاد کنید:

```
$ sudo mkdir -p /mnt/md0
```

با دستور زیر می توانید فایل سیستم را Mount کنید:

```
$ sudo mount /dev/md0 /mnt/md0
```

پس از آن، بررسی کنید که آیا فضای جدید موجود است:

```
$ df -h -x devtmpfs -x tmpfs
```

خروجی به صورت زیر خواهد بود:

| Filesystem | Size | Used | Avail | Use% | Mounted on |
|------------|------|------|-------|------|------------|
| /dev/vda1  | 25G  | 1.4G | 23G   | 6%   | /          |
| /dev/vda15 | 105M | 3.4M | 102M  | 4%   | /boot/efi  |
| /dev/md0   | 197G | 60M  | 187G  | 1%   | /mnt/md0   |

فایل سیستم جدید اکنون نصب شده و قابل دسترسی است.

ذخیره چیدمان آرایه

برای اطمینان از اینکه آرایه به طور خودکار در هنگام بوت جمع می شود، باید فایل `/etc/mdadm/mdadm.conf` را تنظیم کنید. شما می توانید به طور خودکار آرایه فعال را اسکن کرده و فایل را با موارد زیر اضافه کنید:

```
$ sudo mdadm --detail --scan | sudo tee -a /etc/mdadm/mdadm.conf
```

پس از آن، می توانید `initramfs` یا سیستم فایل `RAM` اولیه را به روزرسانی کنید تا آرایه در طول فرآیند بوت اولیه در دسترس باشد:

```
$ sudo update-initramfs -u
```

برای نصب خودکار در هنگام بوت، گزینه های نصب فایل سیستم جدید را به فایل `/etc/fstab` اضافه کنید:

```
$ echo '/dev/md0 /mnt/md0 ext4 defaults,nofail,discard 0 0' | sudo tee -a /etc/fstab
```

آرایه `RAID 6` شما اکنون به طور خودکار هر بوت را مونتاژ و سوار می کند. اکنون راه اندازی `RAID` خود را به پایان رسانده اید. اگر می خواهید یک `RAID` دیگر را امتحان کنید، دستورالعمل های بازنشانی در انتهای سطوح قرار دارد را دنبال کنید تا با ایجاد یک نوع آرایه `RAID` جدید ادامه دهید.

## تکنولوژی raid 10

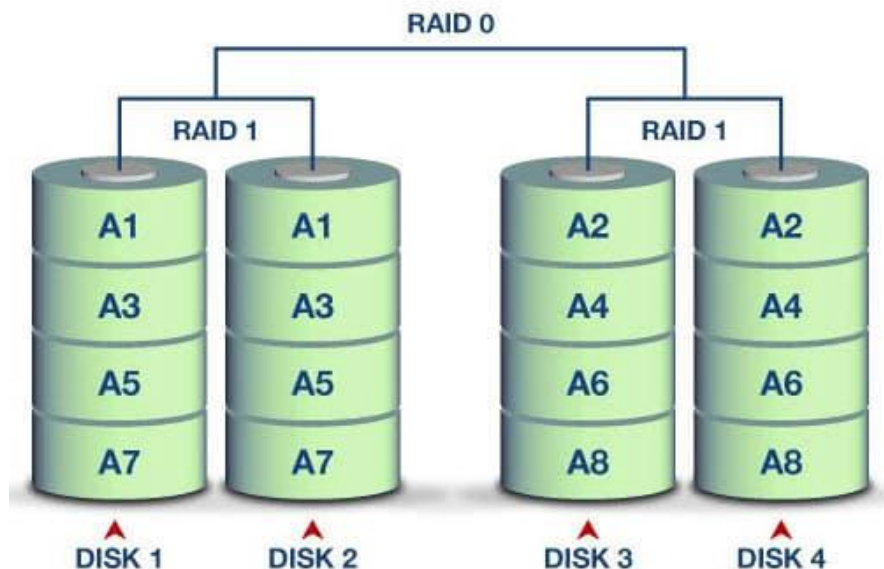
مدل RAID 10 یکی دیگر از انواع RAID می باشد که از ترکیب دو حالت RAID 0 و RAID 1 تشکیل می شود. برای مثال در صورتی که ۶ هارد داشته باشیم ، این هاردها با هم RAID 1 شده و امنیت بالایی را به وجود می آورند و تعداد ۳ هارد باقیمانده نیز به صورت RAID 0 شده تا سرعت نوشتن بر روی این هاردها نیز به نسبت افزایش یابد (لازم به ذکر است که سرعت در این روش نسبت به حالت RAID 5 , 6 کمتر است).

RAID 10 که با نام RAID 1+0 نیز شناخته می شود، یک پیکربندی RAID است که برای محافظت از داده ها، انعکاس دیسک و نوار دیسک را ترکیب می کند. حداقل به چهار دیسک و داده نواری در جفت های آینه شده نیاز دارد. تا زمانی که یک دیسک در هر جفت آینه کاری شده کارایی داشته باشد، داده ها قابل بازیابی هستند.

نوع آرایه RAID 10 به طور سنتی با ایجاد یک آرایه راه راه RAID 0 متشکل از مجموعه ای از آرایه های RAID 1 پیاده سازی می شود. این نوع آرایه تو در تو، هم افزونگی و هم کارایی بالا را به هزینه مقدار زیادی فضای دیسک می دهد. ابزار mdadm نوع RAID 10 خود را دارد که با افزایش انعطاف پذیری، مزایای مشابهی را ارائه می کند. توسط آرایه های تودرتو ایجاد نمی شود، اما بسیاری از ویژگی ها و تضمین های مشابه را دارد. در اینجا از mdadm RAID 10 استفاده خواهید کرد.

## Raid 1+0

### RAID 10





شرایط مورد نیاز: حداقل 3 دستگاه ذخیره سازی.

مزیت اصلی: عملکرد و افزونگی.

مواردی که باید در نظر داشته باشید: میزان کاهش ظرفیت برای آرایه با تعداد کپی های داده ای که انتخاب می کنید برای نگهداری تعریف می شود. تعداد کپی هایی که با RAID 10 به سبک mdadm ذخیره می شوند قابل تنظیم است.

به طور پیش فرض، دو نسخه از هر بلوک داده در چیزی که طرح نزدیک نامیده می شود ذخیره می شود. طرح بندی های ممکن که نحوه ذخیره هر بلوک داده را دیکته می کند به شرح زیر است:

➤ **نزدیک:** ترتیب پیش فرض. کپی های هر تکه به طور متوالی هنگام خط بندی نوشته می شوند، به این معنی که کپی های بلوک های داده در اطراف همان قسمت از چندین دیسک نوشته می شوند.

➤ **دور:** اولین و نسخه های بعدی در قسمت های مختلف دستگاه های ذخیره سازی آرایه نوشته می شوند. به عنوان مثال، تکه اول ممکن است در نزدیکی ابتدای یک دیسک نوشته شود، در حالی که تکه دوم در نیمه راه روی دیسک دیگری نوشته شود. این می تواند باعث افزایش عملکرد خواندن برای دیسک های چرخان سنتی به قیمت عملکرد نوشتن شود.

➤ **انحراف یا offset:** هر نوار کپی شده و توسط یک درایو آفست (منحرف) می شود. این بدان معنی است که کپی ها از یکدیگر آفست هستند، اما همچنان روی دیسک به هم نزدیک هستند. این کمک می کند تا جستجوی بیش از حد در طول برخی از بارهای کاری به حداقل برسد.

با بررسی بخش RAID10 در صفحه man می توانید درباره این طرح بندی ها اطلاعات بیشتری کسب کنید:

\$ man 4 md

برای شروع، شناسه های دیسک های خامی را که استفاده می کنید پیدا کنید:

\$ lsblk -o NAME,SIZE,FSTYPE,TYPE,MOUNTPOINT

| NAME    | SIZE  | FSTYPE | TYPE | MOUNTPOINT |
|---------|-------|--------|------|------------|
| sda     | 100G  | disk   |      |            |
| sdb     | 100G  | disk   |      |            |
| sd c    | 100G  | disk   |      |            |
| sdd     | 100G  | disk   |      |            |
| vda     | 25G   | disk   |      |            |
| └─vda1  | 24.9G | ext4   | part | /          |
| └─vda14 | 4M    |        | part |            |
| └─vda15 | 106M  | vfat   | part | /boot/efi  |

در این مثال، شما چهار دیسک بدون سیستم فایل دارید که هر کدام دارای 100 گیگابایت ظرفیت هستند. به این دستگاه ها شناسه های `/dev/sda` ، `/dev/sdb` ، `/dev/sdc` و `/dev/sdd` برای این جلسه داده شده است و اجزای خام مورد استفاده برای ساخت آرایه خواهند بود.

برای ایجاد یک آرایه RAID 10 با این اجزا، آنها را به دستور `mdadm --create` منتقل کنید. شما باید نام دستگاهی را که می خواهید ایجاد کنید، سطح RAID و تعداد دستگاه ها را مشخص کنید. در این دستور، نام دستگاه را `/dev/md0` می گذارید و دیسک هایی را که آرایه را می سازند شامل می شوید:

می توانید با استفاده از طرح بندی نزدیک، با تعیین نکردن شماره طرح و کپی، دو نسخه تنظیم کنید:

```
$ sudo mdadm --create --verbose /dev/md0 --level=10 --raid-
devices=4 /dev/sda /dev/sdb /dev/sdc /dev/sdd
```

اگر می خواهید از طرح بندی دیگری استفاده کنید یا تعداد کپی ها را تغییر دهید، باید از گزینه `--layout=` استفاده کنید که شناسه طرح و کپی را می گیرد. طرح بندی ها شامل `n` برای نزدیک، `f` برای دور و `o` برای افست هستند. تعداد نسخه های ذخیره شده پس از آن اضافه می شود. به عنوان مثال، برای ایجاد یک آرایه که دارای سه نسخه در طرح افست باشد، دستور شامل موارد زیر است:

```
$ sudo mdadm --create --verbose /dev/md0 --level=10 --layout=o3 --
raid-devices=4 /dev/sda /dev/sdb /dev/sdc /dev/sdd
```

ابزار `mdadm` شروع به پیکربندی آرایه می کند. از فرآیند بازیابی برای ساخت آرایه به دلایل عملکرد استفاده می کند. این ممکن است کمی طول بکشد، اما آرایه را می توان در این مدت استفاده کرد. می توانید با بررسی فایل `/proc/mdstat`، پیشرفت آینه سازی را کنترل کنید:

```
$ cat /proc/mdstat
```

خروجی دستور بالا به صورت زیر است:

```
Personalities : [raid6] [raid5] [raid4] [linear] [multipath] [raid0] [raid1] [raid10]
md0 : active raid10 sdd[3] sdc[2] sdb[1] sda[0]
      209584128 blocks super 1.2 512K chunks 2 near-copies [4/4] [UUUU]
      [==>.....] resync = 18.1% (37959424/209584128) finish=13.8min speed=206120K/sec

unused devices: <none>
```

در اولین هایلایت، دستگاه `/dev/md0` در پیکربندی RAID 10 با استفاده از `/dev/sda`، `/dev/sdb`، `/dev/sdc` و `/dev/sdd` ایجاد شده است. دومین هایلایت، طرح بندی مورد استفاده

برای این مثال را نشان می‌دهد (دو نسخه در پیکربندی نزدیک). سومین ناحیه برجسته پیشرفت در ساخت را نشان می‌دهد. می‌توانید تا زمانی که این فرآیند کامل شد، راهنما را ادامه دهید.

## ایجاد و نصب فایل سیستم

بعد، یک فایل سیستم روی آرایه ایجاد کنید:

```
$ sudo mkfs.ext4 -F /dev/md0
```

سپس، یک نقطه اتصال برای پیوست کردن فایل سیستم جدید ایجاد کنید:

```
$ sudo mkdir -p /mnt/md0
```

با دستور زیر می‌توانید فایل سیستم را Mount کنید:

```
$ sudo mount /dev/md0 /mnt/md0
```

پس از آن، بررسی کنید که آیا فضای جدید موجود است:

```
$ df -h -x devtmpfs -x tmpfs
```

خروجی به صورت زیر خواهد بود:

| Filesystem | Size | Used | Avail | Use% | Mounted on |
|------------|------|------|-------|------|------------|
| /dev/vda1  | 25G  | 1.4G | 23G   | 6%   | /          |
| /dev/vda15 | 105M | 3.4M | 102M  | 4%   | /boot/efi  |
| /dev/md0   | 197G | 60M  | 187G  | 1%   | /mnt/md0   |

فایل سیستم جدید اکنون نصب شده و قابل دسترسی است.

## ذخیره چیدمان آرایه

برای اطمینان از اینکه آرایه به طور خودکار در هنگام بوت جمع می‌شود، باید فایل `/etc/mdadm/mdadm.conf` را تنظیم کنید. شما می‌توانید به طور خودکار آرایه فعال را اسکن کرده و فایل را با موارد زیر اضافه کنید:

```
$ sudo mdadm --detail --scan | sudo tee -a /etc/mdadm/mdadm.conf
```

پس از آن، می‌توانید `initramfs` یا سیستم فایل `RAM` اولیه را به‌روزرسانی کنید تا آرایه در طول فرآیند بوت اولیه در دسترس باشد:

```
$ sudo update-initramfs -u
```

برای نصب خودکار در هنگام بوت، گزینه های نصب فایل سیستم جدید را به فایل `/etc/fstab` اضافه کنید:

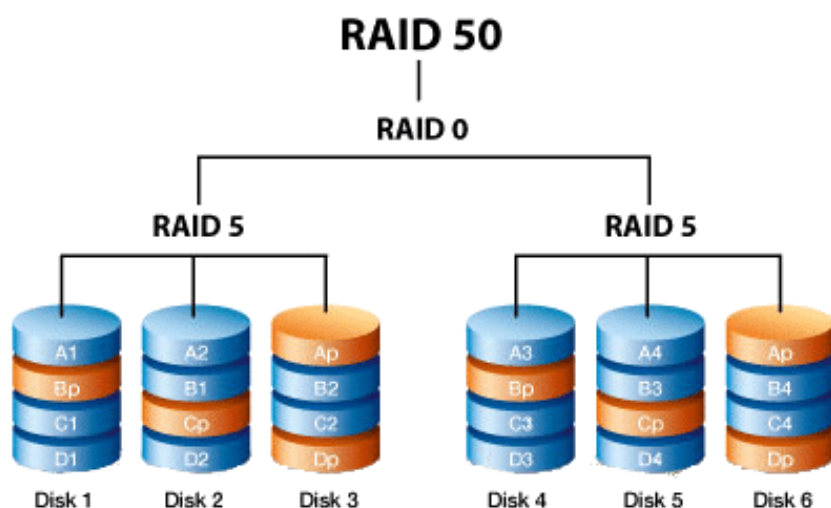
```
$ echo '/dev/md0 /mnt/md0 ext4 defaults,nofail,discard 0 0' | sudo tee -a /etc/fstab
```

آرایه **RAID 10** شما اکنون به طور خودکار هر بوت را مونتاژ و سوار می‌کند. اکنون راه اندازی **RAID** خود را به پایان رسانده اید. اگر می‌خواهید یک **RAID** دیگر را امتحان کنید، دستورالعمل‌های بازنشانی در انتهای سطوح قرار دارد را دنبال کنید تا با ایجاد یک نوع آرایه **RAID** جدید ادامه دهید.

## تکنولوژی raid 50

در مدل RAID 50 نیز از ترکیب دو حالت RAID 0 و RAID 5 استفاده شده است. در برخی از Storage های معمول ، تنها تعداد محدودی از هارد را می توان RAID 5 نمود. این مشکل با معرفی حالت RAID 50 تا حدی برطرف شده است. امروزه در این مدل Storage ها ، در ابتدا تمام هاردها به بخش های کوچک تر تقسیم شده و هر قسمت به صورت RAID 5 تنظیم می شوند. در نهایت تمام بسته های RAID 5 با هم به صورت RAID 0 تنظیم می شود تا امکان دسترسی به تمام هاردها و حجم آنها برای کاربران وجود داشته باشد.

لازم به ذکر است در حالت RAID 5 , 6 ، ظرفیت نهایی Storage به دلیل استفاده از هاردهای رزرو و ساختار داخلی این RAID ها پایین می باشد. برای مثال اگر شما Storage با ظرفیت ۲۴ عدد هارد داشته باشید و تمام هاردهای شما نیز ظرفیت ۲ ترابایت داشته باشند ، ظرفیت نهایی در حالت RAID 5 برابر با ۴۰ ترابایت خواهد شد (در حالت معمول باید این مقدار ۴۸ ترابایت باشد).



## ریدهای ترکیبی یا تو در تو (Nested RAIDs)

در زیر برخی از انواع raid را که ترکیبی از ریدهای استاندارد هستند معرفی خواهیم کرد:

- **Raid 10**
- **Raid 50**
- **Raid 01**: Raid 01 از ترکیب RAID 1 و RAID 0 به دست آمده است. یعنی سرعت بالای Raid 0 و امنیت raid 1 را دارد. RAID 0 + 1 به عنوان RAID 0 پیاده سازی می شود و عناصر آن RAID 1 است و برخلاف Raid 10 ابتدا اطلاعات

کپی می‌شوند و سپس نواربندی می‌شوند. همچنین اجرای تکنولوژی raid 01 بسیار آسان‌تر از raid 6 , raid 5 , raid 3 می‌باشد. یکی از معایب رید هزینه بالای آن است.

• **RAID 60:** Raid 60 نیز ترکیبی از raid 0 و raid 6 می‌باشد. پیکربندی در raid 60 به این شکل است که نیاز به ۸ هارد داریم و اطلاعات بین دو دسته هاردی که بصورت raid6 می‌باشند بصورت نواری توزیع می‌شود. یکی از مزایای رید ۶۰ این است که اگر سیستم نیز دچار مشکلی شود باز می‌تواند بدون وقفه به کار خود ادامه دهد.

## مزایای استفاده از RAID

- کاهش هزینه ها
- افزایش کارایی با استفاده از چند هارد دیسک
- بالا رفتن سرعت سرعت خواندن و نوشتن
- سرعت بازیابی بالا
- تأمین افزونگی برای هارد از دیگر مزایای رید است.

## معایب استفاده از RAID

- زمانی که درایوی خراب شود امکان خرابی در دیگر درایوهای آرایه نیز امکان پذیر است.
- با خراب شدن یک درایو امکان از دست رفتن اطلاعات وجود دارد.
- Raid نرم افزاری کندتر از Raid سخت افزاری عمل می‌کند.

## raid controller چیست؟

رید کنترلر یک قطعه‌ی سخت‌افزاری یا نرم‌افزاری می‌باشد. در اصل می‌توان گفت برای اینکه یک هارد دیسک بتواند وظیفه خود را به درستی انجام دهد، برای مدیریت و پشتیبانی، نیاز به یک raid controller دارد.

در واقع یک تراشه است که بین هارد دیسک و سیستم عامل قرار می‌گیرد تا عملکرد دیسک را بهبود ببخشد و حجم زیادی داده را مدیریت کنند. رید کنترلر برای بهبود کارایی استفاده می‌شود و در هنگام رخداد خرابی، از اطلاعات محافظت می‌کند. از مهمترین مزایای استفاده از رید کنترلر می‌توان به محافظت از هارد، افزایش طول عمر مفید HDD و افزایش عملکرد هارد اشاره کرد.

## بهترین نوع RAID کدام است؟

بهترین نوع raid کدام است؟ همانطور که می دانید سطح و نوع رید بر اساس نیازهای اپلیکیشن های در حال اجرا بر روی سرور باید مشخص شود. اما در حالت کلی RAID 0 سریع ترین رید، RAID 1 مطمئن ترین رید، و RAID 5 ترکیبی از سرعت و قابلیت اطمینان بودن را دارد.

## ابزار مورد نیاز

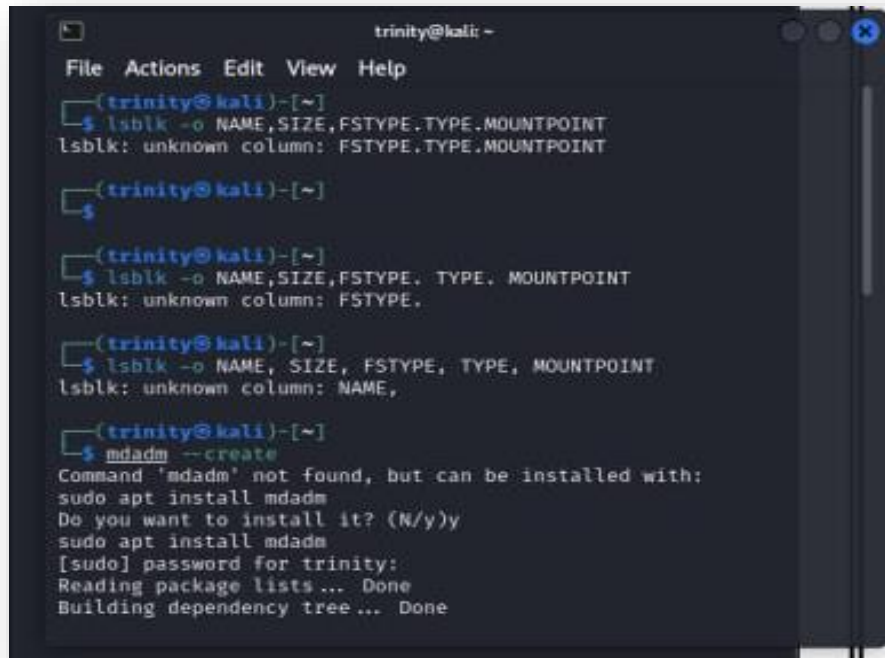
ابزار mdadm را می توان برای ایجاد و مدیریت آرایه های ذخیره سازی با استفاده از قابلیت های RAID در نرم افزار لینوکس استفاده کرد. مدیران از انعطاف پذیری بالایی در هماهنگ کردن دستگاه های ذخیره سازی فردی خود و ایجاد دستگاه های ذخیره سازی منطقی که دارای عملکرد یا ویژگی های افزونگی بیشتری هستند برخوردارند.

## پیش نیازهای لازم

بسته به نوع آرایه، شما به دو تا چهار دستگاه ذخیره سازی نیاز دارید. این درایوها قبل از دنبال کردن این مقاله نیازی به فرمت ندارند. همچنین می بایست یک نسخه لینوکس را داشته باشید.

## انجام Process Raid

ما برای انجام پروژه raiding از کالی لینوکس استفاده کردیم.  
برای اینکه بتوانیم اسامی دیسک‌ها یا همان حافظه‌های سیستم را ببینیم دستور lsblk را می‌زنیم.



```
trinity@kali: ~  
File Actions Edit View Help  
(trinity@kali)~  
$ lsblk -o NAME,SIZE,FSTYPE,TYPE,MOUNTPOINT  
lsblk: unknown column: FSTYPE,TYPE,MOUNTPOINT  
  
(trinity@kali)~  
$  
  
(trinity@kali)~  
$ lsblk -o NAME,SIZE,FSTYPE,TYPE,MOUNTPOINT  
lsblk: unknown column: FSTYPE.  
  
(trinity@kali)~  
$ lsblk -o NAME,SIZE,FSTYPE,TYPE,MOUNTPOINT  
lsblk: unknown column: NAME,  
  
(trinity@kali)~  
$ mdadm --create  
Command 'mdadm' not found, but can be installed with:  
sudo apt install mdadm  
Do you want to install it? (N/y)y  
sudo apt install mdadm  
[sudo] password for trinity:  
Reading package lists... Done  
Building dependency tree... Done
```

چون mdam نصب نیست به ارور برمیخوریم و پیشنهاد بهمون داده میشه که در صورت نیاز نصب شود.  
کمی صبر میکنیم.  
نصب که شد دوباره دستور lsblk رو وارد میکنیم.

```
(trinity@kali)-[~]
$ lsblk -o NAME,SIZE,FSTYPE,TYPE,MOUNTPOINT
NAME        SIZE FSTYPE TYPE MOUNTPOINT
sda          953.9G      disk
├─sda1       500M ntfs   part /media/trinity/5A42C53342C514A3
├─sda2      136.2G ntfs   part /media/trinity/124CCBD74CCBB42F
├─sda3      197.3G ntfs   part /media/trinity/New Volume
├─sda4         1K      part
├─sda5      275.4G ntfs   part /media/trinity/New Volume2
├─sda6      266.4G ntfs   part /media/trinity/New Volume1
├─sda7       77.2G ext4   part /
├─sda8       971M swap   part [SWAP]
sdb          29.4G      disk
├─sdb1       29.4G vfat   part
sdc           7.5G      disk
├─sdc1       7.5G vfat   part
sdd          58.6G      disk
├─sdd1       58.6G vfat   part
sde           7.2G      disk
├─sde1       7.2G vfat   part
sr0          1024M      rom
```

```
(trinity@kali)-[~]
$ sudo mdadm -C -l6 -n4 /dev/dd0 /dev/sd{b,c,d,e}
[sudo] password for trinity:
mdadm: Value "/dev/dd0" cannot be set as devname. Reason: Not POSIX compatible.
```

در اینجا ما ۴ فلش به سیستم خود وصل کرده‌ایم. (با Hub)

در اینجا می‌خواهیم ۴ تا فلش (دو تا 8 گیگ و 64 , 32) را با هم raid کنیم. نام این دیسک‌ها همانطور که در لیست بالا قرار دارند sdb و sdc و sdd و sde است.

به دلیل اینکه برای اولین بار پس از باز کردن ترمینال از دستور **sudo** استفاده می‌کنیم می‌بایست رمز یوزر خود را وارد کنیم.



## درست کردن RAID

برای اینکار می‌توانیم از دستور زیر استفاده کنیم.

```
(trinity@kali)-[~]
$ sudo mdadm -C -l6 -n4 /dev/dd0 /dev/sd{b,c,d,e}
[sudo] password for trinity:
mdadm: Value "/dev/dd0" cannot be set as devname. Reason: Not POSIX compatible.

(trinity@kali)-[~]
$ sudo mdadm -C -l6 -n4 /dev/md0 /dev/sd{b,c,d,e}
mdadm: partition table exists on /dev/sdb
mdadm: partition table exists on /dev/sdb but will be lost or
      meaningless after creating array
mdadm: partition table exists on /dev/sdc
mdadm: partition table exists on /dev/sdc but will be lost or
      meaningless after creating array
mdadm: partition table exists on /dev/sdd
mdadm: partition table exists on /dev/sdd but will be lost or
      meaningless after creating array
mdadm: partition table exists on /dev/sde
mdadm: partition table exists on /dev/sde but will be lost or
      meaningless after creating array
mdadm: largest drive (/dev/sdd) exceeds size (7560192K) by more than 1%
Continue creating array? y
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.

(trinity@kali)-[~]
$
```

همانطور که مشخص است این دستور را با `sudo` زدیم چرا که نیاز به دسترسی در سطح روت دارد.

پس از عبارت `mdadm` که در واقع ابزار `raid` برای ماست از سوییچ `--create` استفاده می‌کنیم تا یک آرایه `raid` بسازیم. عبارت `/dev/md0` هم نام حافظه ایجاد شده است. پس از آن هم از سوییچ `لول` استفاده می‌کنیم. (`--level`). پس از آن می‌بایست تعداد دیوایس‌هایی که در `raid` شرکت می‌کنند را مشخص کنیم. برای این منظور از سوییچ `--raid-devices` استفاده می‌کنیم.

به این نکته توجه کنید که در `raid` سطح ۶ می‌بایست حداقل ۴ دیوایس داشته باشیم. پس از آن هم باید آدرس حافظه‌ها را به دستور بدهیم. در واقع همانطور که بالاتر اشاره کردیم آدرس‌ها در `lsblk` مشخص است.

همینطور می‌توانیم به جای `--create` از `-C` استفاده کنیم. به طور کلی می‌توان برای ساخت این `raid` از دستور زیر هم استفاده کنیم. این دستور دقیقاً کار دستور بالا را می‌کند:

```
$ sudo mdadm -C -l6 -n4 /dev/md0 /dev/sd{a,b,c,d}
```

## ساخت Mount Point

در مرحله بعدی لازم است که یک دایرکتوری بسازیم.

```
File Actions Edit View Help
(trinity@kali)-[~]
$ sudo mkdir /home/raid
[sudo] password for trinity:
mkdir: cannot create directory '/home/raid': File exists
```

با این دستور یک دایرکتوری در پوشه home خود به اسم raid می‌سازیم. این پوشه باید با حافظه raid ساخته شده بعدا mount شوند.

این دستور هم چون دستور ساخت پوشه در home است نیاز به sudo دارد. بعد هم پسورد یوزر باید وارد شود. (ما چون از قبل ساختیم وجود دارد)

## ایجاد سیستم فایل بر روی پارتیشن

پس از این مرحله می‌بایست حافظه ایجاد شده را فرمت کنیم و یک سیستم فایل بر روی پارتیشن ایجاد شده درست کنیم. این کار با دستور را با mkfs انجام می‌دهیم.

```
(trinity@kali)-[~]
$ sudo mkfs.ext4 /dev/md0
mke2fs 1.47.0 (5-Feb-2023)
Creating filesystem with 3780096 4k blocks and 946560 inodes
Filesystem UUID: 6206c14c-43cd-43d1-8a09-43fb84134c6a
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done
```

اگر به جای ext4 از ntfs استفاده کنیم سیستم فایل ایجاد شده از نوع ntfs خواهد بود.

## Mount کردن mount point

```
(trinity@kali)-[~]  
$ sudo mount /dev/md0 /home/raid  
  
(trinity@kali)-[~]  
$
```

زمانیکه شما یک پارتیشن را ایجاد می کنید تا زمانیکه پارتیشن مورد نظر را به یک directory یا mount point در سیستم متصل یا mount نکنید قادر به استفاده کردن از آن پارتیشن نخواهید بود. اینکار با استفاده از دستوری به نام mount در سیستم عامل لینوکس استفاده می شود.

با اینکار عملیات raid تمام می شود. حال اگر lsblk را دوباره بزنیم می توانیم ببینیم که نوع حافظه ها از part به raid6 تغییر کرده است.

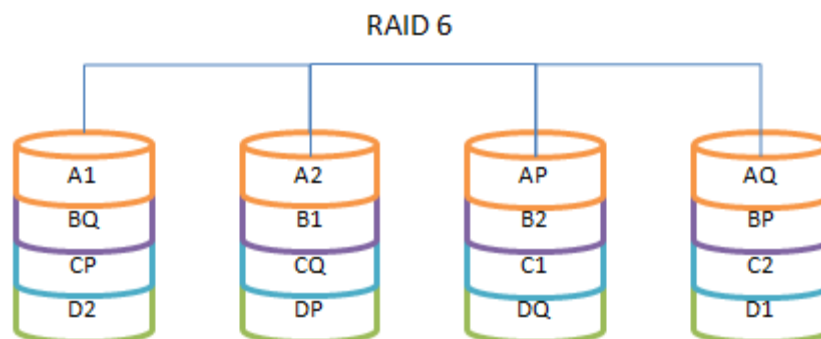
### مشاهده نتیجه

دستور زیر اطلاعات دیسک های ما را نشان می دهد:

```
trinity@kali: ~  
File Actions Edit View Help  
(trinity@kali)-[~]  
$ lsblk  
NAME        MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINTS  
sda         8:0    0 953.9G  0 disk  
├─sda1      8:1    0   500M  0 part  /media/trinity/5A42C53342C514A3  
├─sda2      8:2    0 136.2G  0 part  /media/trinity/124CCBD74CCBB42F  
├─sda3      8:3    0 197.3G  0 part  /media/trinity/New Volume  
├─sda4      8:4    0    1K    0 part  
├─sda5      8:5    0 275.4G  0 part  /media/trinity/New Volume2  
├─sda6      8:6    0 266.4G  0 part  /media/trinity/New Volume1  
├─sda7      8:7    0   77.2G  0 part  /  
└─sda8      8:8    0   971M  0 part  [SWAP]  
sdb         8:16    1  29.4G  0 disk  
├─sdb1      8:17    1  29.4G  0 part  
└─md0       9:0    0  14.4G  0 raid6 /home/raid  
sdc         8:32    1   7.5G  0 disk  
├─sdc1      8:33    1   7.5G  0 part  
└─md0       9:0    0  14.4G  0 raid6 /home/raid  
sdd         8:48    1  58.6G  0 disk  
├─sdd1      8:49    1  58.6G  0 part  
└─md0       9:0    0  14.4G  0 raid6 /home/raid  
sde         8:64    1   7.2G  0 disk  
├─sde1      8:65    1   7.2G  0 part  
└─md0       9:0    0  14.4G  0 raid6 /home/raid  
sr0         11:0    1 1024M  0 rom
```

همانطور که در تصویر می بینید mount point که بالاتر به آن اشاره شد در حافظه های فعال در عمل raid به /home/raid یا همان دایرکتوری که ساختیم تغییر کرده اند.

اکنون اگر ما دیتایی را به حافظه‌ی ایجاد شده منتقل کنیم به صورت زیر منتقل می‌شود:



همانطور که مشخص است حداقل نیاز به ۴ حافظه داریم. همینطور اگر یک دیتا انتقال داده شود با توجه به اینکه فایل تکه تکه منتقل می‌شود و دو حافظه هم تحت Parity داریم هم سرعت خواندن و نوشتن در این شیوه بیشتر از یک حافظه است و هم اطمینان بیشتری نسبت به دیتای منتقل شده از لحاظ از دست رفتن دیتا داریم.

تفاوت اصلی بین RAID 5 و RAID 6 این است که آرایه RAID 5 می‌تواند پس از یک disk failure به کار خود ادامه دهد، اما آرایه RAID 6 می‌تواند دو disk failure همزمان را حفظ کند و همچنان به کار خود ادامه دهد. آرایه های RAID 6 نیز کمتر در معرض خطا در فرآیند بازسازی دیسک هستند.

خرابی هارد دیسک یا disk failure زمانی اتفاق می‌افتد که یک درایو دیسک سخت کار نمی‌کند و نمی‌توان با رایانه‌ای که به درستی پیکربندی شده است به اطلاعات ذخیره‌شده دسترسی پیدا کرد.

خرابی هارد دیسک ممکن است در طول کارکرد معمولی یا به دلیل یک عامل خارجی مانند قرار گرفتن در معرض آتش یا آب یا میدان‌های مغناطیسی زیاد یا ضربه شدید یا آلودگی محیطی رخ دهد که می‌تواند منجر به تصادف شود.

اطلاعات ذخیره شده روی هارد دیسک نیز ممکن است در نتیجه خرابی داده‌ها، اختلال یا از بین رفتن رکورد اصلی راه اندازی هارد دیسک، یا بدافزاری که به طور عمدی محتویات دیسک را از بین می‌برد، غیرقابل دسترسی باشد.

## برگرداندن به حالت اول

برای اینکار ابتدا باید mount point ای که آن را mount کرده بودیم را unmount کنیم.

این کار را با دستور umount انجام می‌دهیم:

```
$ umount /dev/md0
```

سپس باید mdadm را در حافظه ایجاد شده را متوقف کنیم.

```
$ mdadm --stop /dev/md0
```

با دستور زیر هم تک تک حافظه‌های استفاده شده در فرایند raid را از آن جدا می‌کنیم:

```
$ mdadm --zero-superblock /dev/sd{a,b,c,d}
```

پس از انجام سه مرحله بالا می‌بینیم که حافظه‌ها (فلش‌ها) به حالت اول برگشته‌اند.

```
(trinity@kali)-[~]
$ sudo umount /dev/md0
[sudo] password for trinity:

(trinity@kali)-[~]
$ mdadm --stop /dev/md0
mdadm: must be super-user to perform this action

(trinity@kali)-[~]
$ sudo mdadm --stop /dev/md0
mdadm: stopped /dev/md0

(trinity@kali)-[~]
$ sudo mdadm --zero-superblock /dev/sd{b,c,d,e}

(trinity@kali)-[~]
$ lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS
sda 8:0 0 953.9G 0 disk
├─sda1 8:1 0 500M 0 part /media/trinity/5A42C53342C514A3
├─sda2 8:2 0 136.2G 0 part /media/trinity/124CCBD74CCBB42F
└─sda3 8:3 0 197.3G 0 part /media/trinity/New Volume
```



```
trinity@kali: ~  
$ sudo mdadm --zero-superblock /dev/sd{b,c,d,e}  
  
trinity@kali: ~  
$ lsblk  
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS  
sda          8:0    0 953.9G  0 disk  
├─sda1       8:1    0   500M  0 part /media/trinity/5A42C53342C514A3  
├─sda2       8:2    0  136.2G  0 part /media/trinity/124CCBD74CCBB42F  
├─sda3       8:3    0  197.3G  0 part /media/trinity/New Volume  
├─sda4       8:4    0    1K    0 part  
├─sda5       8:5    0  275.4G  0 part /media/trinity/New Volume2  
├─sda6       8:6    0  266.4G  0 part /media/trinity/New Volume1  
├─sda7       8:7    0   77.2G  0 part /  
└─sda8       8:8    0   971M  0 part [SWAP]  
sdb          8:16   1  29.4G  0 disk  
└─sdb1       8:17   1  29.4G  0 part  
sdc          8:32   1   7.5G  0 disk  
└─sdc1       8:33   1   7.5G  0 part  
sdd          8:48   1  58.6G  0 disk  
└─sdd1       8:49   1  58.6G  0 part  
sde          8:64   1   7.2G  0 disk  
└─sde1       8:65   1   7.2G  0 part  
sr0         11:0    1 1024M  0 rom
```

## منابع

- <https://iranhost.com/blog/raid-%D9%88-%D8%A7%D9%86%D9%88%D8%A7%D8%B9-%D8%A2%D9%86/#gref>
- [https://en.wikipedia.org/wiki/Hard\\_disk\\_drive\\_failure](https://en.wikipedia.org/wiki/Hard_disk_drive_failure)
- <https://novablog.ir/%D8%A2%D9%85%D9%88%D8%B2%D8%B4-%D9%86%D8%B5%D8%A8-%D8%B3%DB%8C%D8%B3%D8%AA%D9%85-%D8%B9%D8%A7%D9%85%D9%84-ubuntu-14-04/>
- <https://digiato.com/article/2022/09/28/install-linux-on-windows>
- <https://linuxlearn.org/raid-in-linux/>
- <https://www.digitalocean.com/community/tutorials/how-to-create-raid-arrays-with-mdadm-on-ubuntu-18-04>
- <https://www.techtarget.com/searchdatabackup/tip/RAID-5-vs-RAID-6-Capacity-performance-durability#:~:text=The%20primary%20difference%20between%20RAID,during%20the%20disk%20rebuilding%20process>

```
min=0.01, max=1000.0,  
default=1.0,  
)
```

```
execute(self, context):
```

```
    # get the folder
```

```
    folder_path = (os.path.dirname(self.filepath))
```

```
    # get objects selected in viewport
```

```
    viewport_selection = bpy.context.selected_objects
```

```
    # get export objects
```

```
    export_list = []  
    self.use_selection_setting == False:
```

```
    obj_export_list = [i for i in bpy.context.scene.objects]
```

```
    # select all objects
```

```
    ops.object.select_all(action='DESELECT')
```

```
    for item in obj_export_list:
```

```
        item.select = True
```

```
        if item.type == 'MESH':
```

```
            file_path = os.path.join(folder_path, "{}.obj".format(item.name))  
            bpy.ops.export_scene.obj(filepath=file_path, use_selection=True,
```

```
                                    axis_forward=self.axis_forward_setting,
```

```
                                    axis_up=self.axis_up_setting,
```

```
                                    use_animation=self.use_animation_setting,
```

```
                                    use_mesh_modifiers=self.use_mesh_modifiers_setting,
```

```
                                    use_edges=self.use_edges_setting,
```

```
                                    use_faces=self.use_faces_setting,
```

```
                                    use_normals=self.use_normals_setting,
```

```
                                    use_tessellation=self.use_tessellation_setting,
```

```
                                    use_uv=self.use_uv_setting,
```

```
                                    use_wireframe=self.use_wireframe_setting,
```

```
                                    use_x_ray=self.use_x_ray_setting,
```

```
                                    use_z_axis=self.use_z_axis_setting,
```

```
                                    use_z_axis_reverse=self.use_z_axis_reverse_setting,
```

```
                                    use_z_axis_reverse_reverse=self.use_z_axis_reverse_reverse_setting,
```

```
                                    use_z_axis_reverse_reverse_reverse=self.use_z_axis_reverse_reverse_reverse_setting,
```

```
                                    use_z_axis_reverse_reverse_reverse_reverse=self.use_z_axis_reverse_reverse_reverse_reverse_setting,
```

```
                                    use_z_axis_reverse_reverse_reverse_reverse_reverse=self.use_z_axis_reverse_reverse_reverse_reverse_reverse_setting,
```

```
                                    use_z_axis_reverse_reverse_reverse_reverse_reverse_reverse=self.use_z_axis_reverse_reverse_reverse_reverse_reverse_reverse_setting,
```

```
                                    use_z_axis_reverse_reverse_reverse_reverse_reverse_reverse_reverse=self.use_z_axis_reverse_reverse_reverse_reverse_reverse_reverse_reverse_setting,
```

```
                                    use_z_axis_reverse_reverse_reverse_reverse_reverse_reverse_reverse_reverse=self.use_z_axis_reverse_reverse_reverse_reverse_reverse_reverse_reverse_reverse_setting,
```

```
                                    use_z_axis_reverse_reverse_reverse_reverse_reverse_reverse_reverse_reverse_reverse=self.use_z_axis_reverse_reverse_reverse_reverse_reverse_reverse_reverse_reverse_reverse_setting,
```

```
                                    use_z_axis_reverse_reverse_reverse_reverse_reverse_reverse_reverse_reverse_reverse_reverse=self.use_z_axis_reverse_reverse_reverse_reverse_reverse_reverse_reverse_reverse_reverse_reverse_setting,
```

**Any fool can write code that a  
computer can understand. Good  
programmers write code that  
humans can understand.**