1 CentOS7开机启动流程

1.1 硬件引导启动

1. Power On
2. 进入BIOS，先运行post初始化硬件，再查找启动介质，查找启动硬盘的0磁头、0磁道、1扇区，也就是Bootsector，定位后MBR即被装载到RAM中，BIOS将控制器交给MBR。
3. MBR，其中446字节是BootLoader，安装了grub2的boot.img

1.2 GRUB2引导阶段

加载装载程序的配置文件：/etc/grub.d/ /etc/default/grub /boot/grub2/grub.cfg
![[Pasted image 20220911234339.png]]
![[Pasted image 20220911181621.png]]

1.3 内核引导阶段

加载initramfs驱动模块和内核vmlinuz

1.4 systemd初始化阶段

systemd执行默认target配置，配置文件/etc/systemd/system/default.target

2 故障排查

2.1 主机故障排查

1. **系统启动故障**
   root文件系统被破坏，导致系统无法启动

- 进入emergency模式
- 检查/etc/fstab配置文件的正确性
- 使用fsck.ext4或xfs.repair命令检查并修复文件系统
  /etc/fstab文件里的配置出错，导致系统启动不了，进入单用户模式也没用，因为可能有些系统工具所在的磁盘没有挂载。需要进入营救模式，找到系统盘挂载的路径，修改fstab文件
  挂载时使用设备名不太好，因为设备名可能会变化，最好使用设备的uuid，查看uuid，tune2fs

2. **用户登录故障**
   忘记root密码

```
centos6：进入单用户模式（进入grub启动菜单项的编辑界面输入single然后ctrl+x执行）-->passwd命令修改密码
centos7：
# 1 进入grub启动菜单项的编辑界面中linux16行末添加init=bin/sh或rd.break，然后ctrl+x开始
# 2 解决乱码现象
locale # 查看当前编码
# 可以看到LANG=zh_CN.UTF-8
临时更改编码
export LANG=en_US
# 3 此时根文件系统还是只读的，需要先以读写方式重新挂载/
mount -o remount,rw /
```

```
# 4 更改密码
passwd root
# 5 重新标记selinux
touch /.autorelabel # 或者把selinux改为disabled状态
# 6 重启
exec /sbin/init
```

![[Pasted image 20220911224800.png]]
su命令切换用户带来的问题

```
# 故障现象：su:warning:cannot change directory to /home/oracle:Permission denied
# 看用户家目录的权限
# 看su命令的权限
ls -al /bin/su
# 看su命令的依赖系统库
ldd /bin/su
# 看selinux
cat /etc/selinux/config
# 查看一个文件或目录的详细信息
# 根目录权限
stat /
```

3. Read-only file system

```
思路：可能时服务器磁盘故障（磁盘空间满了或者磁盘无法写入了）
原因：磁盘分区出现了问题，导致文件系统结构不一致，文件系统关闭了写功能，需要修改文件系统
解决：umount /www/data;fsck -y /dev/sdb1
```

4. Too many open files

```
[root@localhost ~]# ulimit -a
core file size          (blocks, -c) 0    # 最大的core文件的大小，以blocks为单位
data seg size           (kbytes, -d) unlimited
scheduling priority             (-e) 0
file size               (blocks, -f) unlimited    # 进程可以创建文件的最大值，以blocks为单位
pending signals                 (-i) 3795
max locked memory       (kbytes, -l) 64   # 最大可加锁内存
max memory size         (kbytes, -m) unlimited   # 最大内存大小，以kbytes为单位
open files                      (-n) 1024    # 可以打开的最大文件描述符的数量
pipe size            (512 bytes, -p) 8
POSIX message queues     (bytes, -q) 819200
real-time priority              (-r) 0
stack size              (kbytes, -s) 8192
cpu time               (seconds, -t) unlimited   # 最大cpu占用时间
max user processes              (-u) 3795    # 用户最多可用的进程数
virtual memory          (kbytes, -v) unlimited   # 进程最大可用的虚拟内存，以kbytes为单位
file locks                      (-x) unlimited
# 配置文件
vi /etc/security/limits.conf
vi /etc/security/limits.d/20-nproc.conf
# 如果这两个文件都配置了会以20-nproc.conf为主
# 查看进程的启动时间
pgrep -f tomcat
```

```
24 ps -eo pid,lstart,etime|grep 3028
```

2.2 网络故障排查

1. 排除非自身因素
2. 查看本机ip地址
3. 检测与网关的连接
4. 检测与互联网的连接
5. 测试域名解析
6. 测试与特定站点的连接
7. 若本机的服务无法被访问

```
1 确定服务有没有启动，ps、top、netstat、lsof
2 服务监听的ip
3 iptables
4 selinux
```

3 Linux日志分析

3.1 日志分类

1. 内核及系统日志，这种日志数据由系统服务rsyslog统一管理，根据其主配置文件/etc/rsyslog.conf中的设置决定将内核消息及各种系统程序消息记录到什么位置。
2. 用户日志，这种日志数据用于记录Linux系统用户登录及退出系统的相关信息，包括用户名、登录的终端、登录时间、来源主机、正在使用的进程操作等。
3. 程序日志，有些应用程序会选择自己来独立管理一份日志文件，用于记录本程序运行过程中的各种实际信息。

3.2 系统日志文件

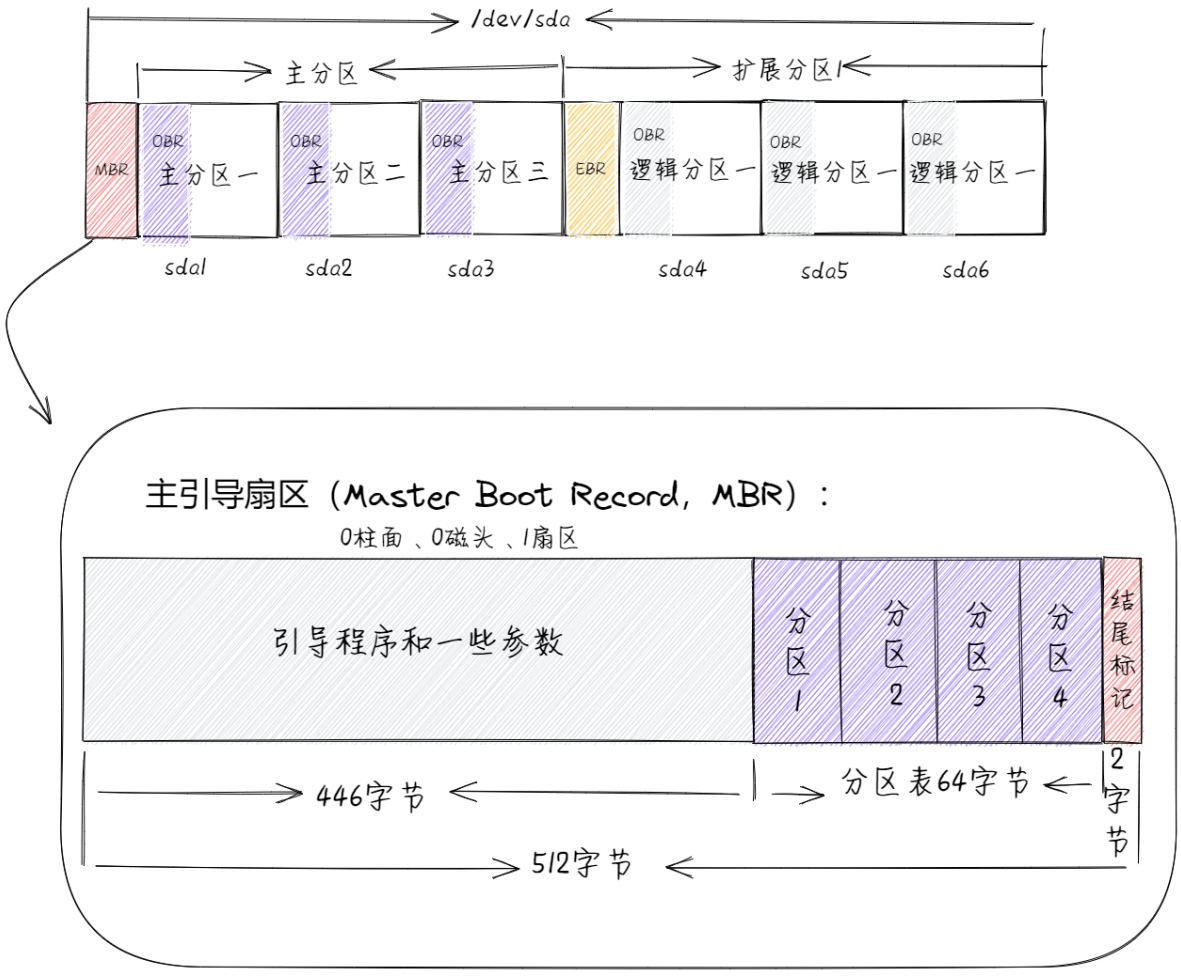
![[Pasted image 20220911220734.png]]
ssh远程登录和sudo的一些信息记录在/var/log/secure
wtmp、btmp、lastlog文件都是二进制文件，是为了防止非法修改，可以通过相应的命令查看。
dmesg系统启动时硬件相关信息，boot.log启动时软件的日志信息

3.3 日志级别

级别字段	说明
emerg	紧急，系统不可以
alert	需要立即引起注意的情况
crit	危险
err	报错
warning	警告
notice	需引起注意
info	值得报告
debug	debug模式下程序产生的消息
none	用于禁止任何消息

4 磁盘管理

4.1 磁盘分区



4.1.1 为什么要分区

- **方便管理和维护**：将不同类型的数据分别存放在不同的磁盘分区中，这样管理和维护就容易多了。
- **提升系统的效率**：对磁盘进行分区可以在系统读写磁盘时减小搜寻(Search)的范围，使磁头移动的距离缩短；硬盘分区也可以减轻碎片(文件不连续存放)所造成的系统效率下降的问题。
- **运用磁盘配额的功能限制用户运用的磁盘量**：因为限制用户运用磁盘配额(Quotas)的功能只能在分区一级上运用，所以为了限制用户运用磁盘的总量以防止用户浪费磁盘空间(甚至将磁盘空间耗光)，最好将磁盘先分区再分配给一般用户。
- **便于备份和恢复**：只需要备份所需的分区，而不用备份整个磁盘。

4.1.2 主分区与扩展分区

在MBR分区表中最多4个主分区或者3个主分区＋1个扩展分区，也就是说扩展分区只能有一个，然后可以再细分为多个逻辑分区。

4.1.3 分区表类型

	MBR磁盘（dos）	GPT磁盘
全称	Master Boot Record（主引导记录）	全局唯一标识分区表（GUID Partition Table，缩写：GPT）
最大分区数	最多4个主分区或者3个主分区＋1个扩展分区，64/16	最初规划硬盘分区时，留给分区表的空间决定了最多可以有多少个分区，受到操作系统限制，Windows最大仅支持128个GPT分区
最大分区	2TB，16字节中4字节表示最大扇区数，每个扇区512字节	18 EB
MBR位置	0柱面0磁头1扇区	0柱面0磁头1扇区（出于兼容性考虑）
DPT位置	分区信息直接存储于主引导记录(MBR)中	分区表的位置信息储存在GPT头中，在MBR后
备份		备份分区表，提高分区数据结构的完整性
BIOS启动方式	Legacy	UEFI（仅支持64位系统）
分区工具	fdisk	fdisk（不成熟）、gdisk、parted

```
# 查看磁盘分区表类型
[root@vs ~]# fdisk -l /dev/sdb
WARNING: fdisk GPT support is currently new, and therefore in an
experimental phase. Use at your own discretion.   # 说fdisk的GPT分区功能还在实验阶段

磁盘 /dev/sdb：3298.5 GB，3298534883328 字节，6442450944 个扇区
Units = 扇区 of 1 * 512 = 512 bytes
扇区大小(逻辑/物理)：512 字节 / 512 字节
I/O 大小(最小/最佳)：512 字节 / 512 字节
```

```
磁盘标签类型：gpt # 表示分区表类型位gpt
Disk identifier: BD7FEB7A-BFF4-4774-A426-569BACB3345A


#         Start          End    Size  Type            Name
 1          2048   6056577023    2.8T  Linux filesyste
 2    6056577024   6442450910    184G  Linux filesyste
```

4.1.4 怎么分区

```
# 磁盘分区步骤
# dos分区
fdisk /dev/sdb-->o(创建dos空分区表)-->n(创建新分区)-->p(选择主分区)-->1(分区号)-->第一个扇区-->+100G-->w(保存退出)
# gpt
gdisk /dev/sdb-->n(创建新分区，默认gpt)-->分区号-->第一扇区-->最后一个扇区-->p(查看信息)
# gdisk和fdisk用法类似
```

4.1.5 parted命令

参考parted百度文档

parted的操作都是实时的，也就是说你执行了一个分区的命令，他就实实在在地分区了，而不是像fdisk那样，需要执行w命令写入所做的修改，所以进行parted的测试千万注意不能在生产环境中！

命令行模式

```
parted [option] device [command]
parted /dev/sdb mkpart primary 1000GB 3299G
```

交互模式

```
parted [device]
```

命令	描述
select	选择磁盘
mklabel	创建分区表（msdos、gpt）,mklabel gpt或mklabel
mkpart	创建新分区，mkpart PART-TYPE [FS-TYPE] START END
print	输出分区信息
resize	重新调整分区大小
rescue	恢复被误删的分区

```
mklabel gpt
mkpart primary 0 1000G
mkpart primary 1000GB -1
# 分区类型primary、extended、logical，gpt分区表只有primary
rm 2
```

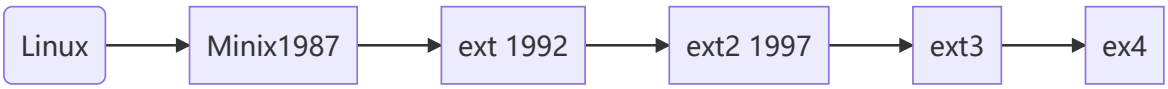
4.2 分区格式化

4.2.1 格式化的作用

格式化种类	作用
低级格式化（物理格式化）	对磁盘的物理表面进行处理，在磁盘上建立标准的磁盘记录格式，划分磁道（track）和扇区（sector），磁盘在出厂时就已经进行了物理格式化
高级格式化（逻辑格式化）	就是根据用户选定的文件系统（FAT32/NTFS/EXT2等等），在磁盘的特定区域写入特定数据，然后初始化磁盘或者某个分区，清除磁盘或者分区中的文件数据，标记出不可读和坏的扇区
快速格式化	仅仅是清楚掉表面数据而已，是可以通过某些手动恢复的。

4.2.2 常见文件系统比较

文件系统的最小分配单位:在Windows下叫做 **簇**；在Linux下叫做 **块（block)**



Linux和Windows都支持FAT系列文件系统

```
B[矩形] C(圆角矩形) D((圆形)) E{菱形} F>右向旗帜型]
```

ext4 VS xfs

	ext4	xfs
单个文件的大小	16GB到16TB	16TB到16EB
最大文件系统大小	1EB	8EB
可扩展性scalability	一般	好

xfs的块大小可以指定

```
mkfs.xfs
mkfs.ext{2,3,4}
```

- 设计文件系统时要考虑存储单元的大小，要分别考虑小文件和大文件的不同存储的需要；
- 文件系统设计也与需要管理的文件规模有关；
- 设计的文件文集系统还要考虑应用场景，根据应用场景的需要设计文件系统；

- 现在我们还要关心文件系统的安全性、可恢复性，文件系统要支持用户权限管理、日志管理等高级功能。
- 在网络环境下，对文件存储、检索、传输。销毁提出更高要求，一些分布式网络存储系统应运而生。

4.2.3 格式化命令

```
mkfs.xfs /dev/sdb1
mkfs.ext{2,3,4}
```

4.3 挂载

4.3.1 查看磁盘信息（uuid）

```
[root@vs ~]# blkid /dev/sdb1
/dev/sdb1: UUID="e643304d-e97d-4ec4-9052-3399ff71ae47" TYPE="xfs"
PARTLABEL="primary" PARTUUID="5b5d30c9-e059-4d7d-bf88-6854d4035033"
[root@vs ~]# blkid /dev/sdb
/dev/sdb: PTTYPE="gpt"

[root@vs ~]# blkid
/dev/sdb1: UUID="e643304d-e97d-4ec4-9052-3399ff71ae47" TYPE="xfs"
PARTLABEL="primary" PARTUUID="5b5d30c9-e059-4d7d-bf88-6854d4035033"
/dev/sda1: UUID="22c4fd57-685f-4de8-8a5c-5c56ab085a84" TYPE="xfs"
/dev/sda2: UUID="tCA2vt-ZvdF-V9jM-Bdu1-DdHZ-WAS4-LnLvN2" TYPE="LVM2_member"
/dev/sdb2: PARTLABEL="primary" PARTUUID="26a79450-a756-4327-90c5-
8378123efc86"
/dev/sr0: UUID="2020-11-04-11-36-43-00" LABEL="CentOS 7 x86_64"
TYPE="iso9660" PTTYPE="dos"
/dev/mapper/centos-root: UUID="678a4149-702a-4196-aa9e-c05933a0a792"
TYPE="xfs"
/dev/mapper/centos-swap: UUID="ee45a93e-c651-45bf-afec-274957beba9d"
TYPE="swap"
```

4.3.2 怎么挂载

mount命令

参数	参数说明
-a	挂载所有fstab中的文件系统
-l	查看挂载信息
-t	指定文件系统类型
-o	挂载选项列表，以英文逗号分隔
UUID=	按文件系统 UUID 指定设备
-U	同 UUID=
-o remount	将一个已经挂下的档案系统重新用不同的方式挂上。例如原先是唯读的系统，现在用可读写的模式重新挂上。
-o ro	用唯读模式挂上。
-o rw	用可读写模式挂上。
-o loop=	使用 loop 模式用来将一个档案当成硬盘分割挂上系统。

```
[root@vs ~]# blkid /dev/sdb2
/dev/sdb2: UUID="e088e85a-a188-4957-bd9e-8c44d96e7590" TYPE="ext4"
PARTLABEL="prim
ary" PARTUUID="26a79450-a756-4327-90c5-8378123efc86" [root@vs ~]# #mount -U
e088e85a-a188-4957-bd9e-8c44d96e7590
[root@vs ~]# mkdir /mnt/sdb2
[root@vs ~]# mount -U e088e85a-a188-4957-bd9e-8c44d96e7590 /mnt/sdb2 -t ext4
```

/etc/fstab

man 5 fstab

4.3.3 查看挂载信息

```
mount -l
# 或
df -Th
```

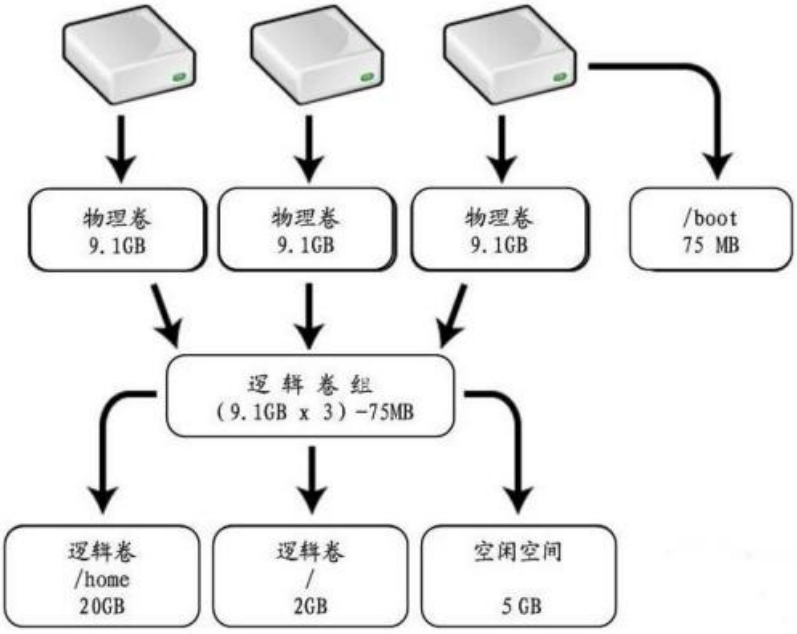
4.4 磁盘扩容

4.4.1 静态分区扩容

- 使用符号链接：将破坏Linux文件系统的标准结构
- 使用调整分区大小的工具，必须停机一段时间
- 重新分区，备份-->清除-->恢复

> 对于ext3/4文件系统可以单独使用resize2fs命令调整（扩展或缩减）文件系统的尺寸；
>
> 对于xfs文件系统，可以单独使用xfs_growfs命令扩展文件系统的尺寸
>
> xfs文件系统还不支持缩减文件系统的尺寸

4.4.2 LVM



已经被挂载的分区不能使用pvcreate命令变成物理卷(PV)

如果一个挂载点下的文件被进程占用，就不能直接取消挂载

使用已经进行高级格式化的分区**生成物理卷**会清除文件系统的数据，里面的数据也就没了。

```
yum install psmisc -y
# 查看挂载点被哪些进程占用
fuser -cu /mount_point
# 杀死这些进程
fuser -ck /mount_point
# 查看是否还有进程在访问挂载点
fuser -c /mount_point
```

创建物理卷

```
[root@vs ~]# pvcreate /dev/sdb1
WARNING: xfs signature detected on /dev/sdb1 at offset 0. Wipe it? [y/n]: y
  Wiping xfs signature on /dev/sdb1.
  Physical volume "/dev/sdb1" successfully created.
[root@vs ~]# pvscan
  PV /dev/sda2   VG centos          lvm2 [<19.00 GiB / 0    free]
  PV /dev/sdb1                      lvm2 [931.32 GiB]
  Total: 2 [<950.32 GiB] / in use: 1 [<19.00 GiB] / in no VG: 1 [931.32 GiB]
[root@vs ~]# pvcreate /dev/sdb2 /dev/sdc
WARNING: ext4 signature detected on /dev/sdb2 at offset 1080. Wipe it?
[y/n]: y
  Wiping ext4 signature on /dev/sdb2.
  Physical volume "/dev/sdb2" successfully created.
  Physical volume "/dev/sdc" successfully created.
[root@vs ~]# pvscan
  PV /dev/sda2   VG centos          lvm2 [<19.00 GiB / 0    free]
  PV /dev/sdc                       lvm2 [4.00 TiB]
  PV /dev/sdb1                      lvm2 [931.32 GiB]
```

```
  PV /dev/sdb2                      lvm2 [2.09 TiB]
  Total: 4 [<7.02 TiB] / in use: 1 [<19.00 GiB] / in no VG: 3 [<7.00 TiB]
```

创建卷组

```
[root@vs ~]# vgcreate data /dev/sdb1 /dev/sdb2
  Volume group "data" successfully created
[root@vs ~]# vgscan
  Reading volume groups from cache.
  Found volume group "centos" using metadata type lvm2
  Found volume group "data" using metadata type lvm2
```

创建逻辑卷

```
# 在data卷组上创建一个名字为video大小10G的逻辑卷
[root@vs ~]# lvcreate -L 10G -n video data
  Logical volume "video" created.
[root@vs ~]# lvscan
  ACTIVE            '/dev/centos/swap' [2.00 GiB] inherit
  ACTIVE            '/dev/centos/root' [<17.00 GiB] inherit
  ACTIVE            '/dev/data/video' [10.00 GiB] inherit
```

扩展卷组

```
[root@vs ~]# vgextend data /dev/sdc
  Volume group "data" successfully extended
[root@vs ~]# vgs
  VG     #PV #LV #SN Attr   VSize   VFree
  centos   1   2   0 wz--n- <19.00g    0
  data     3   1   0 wz--n-  <7.00t 6.99t
```

4.5 VFS