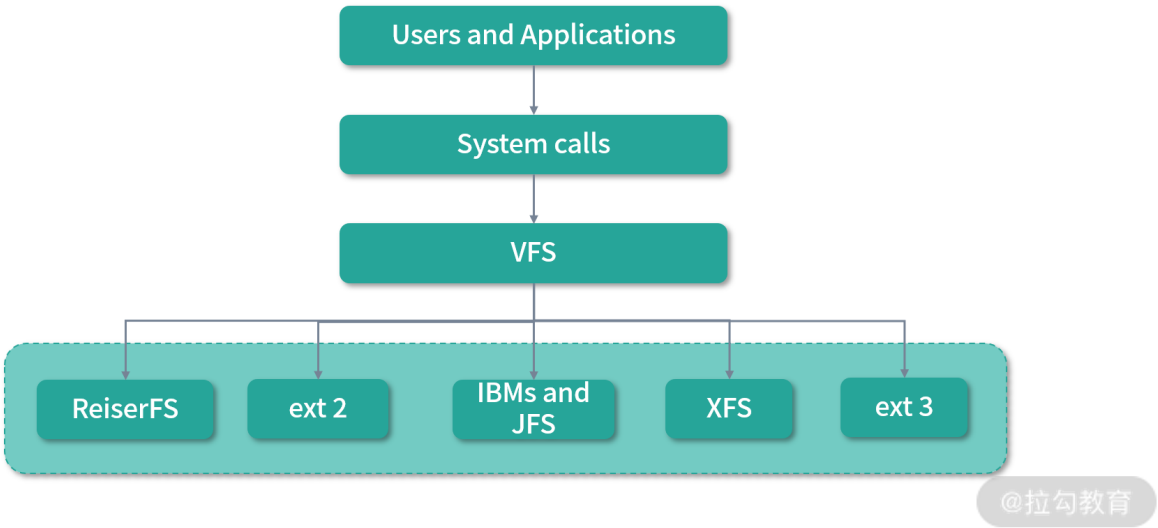
29 Linux 下的各个目录有什么作用?

今天我们开始学习模块六：文件系统。学习文件系统的意义在于文件系统有很多设计思路可以迁移到实际的工作场景中，比如：

- MySQL 的 binlog 和 Redis AOF 都像极了日志文件系统的设计；
- B Tree 用于加速磁盘数据访问的设计，对于索引设计也有通用的意义。

特别是近年来分布式系统的普及，学习分布式文件系统，也是理解分布式架构最核心的一个环节。其实文件系统最精彩的还是虚拟文件系统的设计，比如 Linux 可以支持每个目录用不同的文件系统。这些文件看上去是一个个目录和文件，实际上可能是磁盘、内存、网络文件系统、远程磁盘、网卡、随机数产生器、输入输出设备等，这样虚拟文件系统就成了整合一切设备资源的平台。大量的操作都可以抽象成对文件的操作，程序的书写就会完整而统一，且扩展性强。

这一讲，我会从 Linux 的目录结构和用途开始，带你认识 Linux 的文件系统。Linux 所有的文件都建立在虚拟文件系统（Virtual File System ，VFS）之上，如下图所示：



当你访问一个目录或者文件，虽然用的是 Linux 标准的文件 API 对文件进行操作，但实际操作的可能是磁盘、内存、网络或者数据库等。**因此，Linux 上不同的目录可能是不同的磁盘，不同的文件可能是不同的设备。**

分区结构

在 Linux 中，`/` 是根目录。之前我们在“**08 讲**”提到过，每个目录可以是不同的文件系统（不同的磁盘或者设备）。你可能会问我，`/` 是对应一个磁盘还是多个磁盘呢？在 `/` 创建目录的时候，目录属于哪个磁盘呢？

```
ramroll@u1:/$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            7.8G     0  7.8G   0% /dev
tmpfs           1.6G  1.9M  1.6G   1% /run
/dev/sda5        29G   17G   11G  61% /
tmpfs           7.9G     0  7.9G   0% /dev/shm
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
tmpfs           7.9G     0  7.9G   0% /sys/fs/cgroup
```


你可以用 `df -h` 查看上面两个问题的答案，在上图中我的 `/` 挂载到了 `/dev/sda5` 上。如果你想要看到更多信息，可以使用 `df -T`，如下图所示：

```
ramroll@u1:/$ df -T
Filesystem     Type     1K-blocks     Used Available Use% Mounted on
udev           devtmpfs   8156656        0   8156656   0% /dev
tmpfs          tmpfs      1637016     1904   1635112   1% /run
/dev/sda5      ext4      30313412 17498920  11251612  61% /
```


`/` 的文件系统类型是 `ext4`。这是一种常用的日志文件系统。关于日志文件系统，我会在“**30 讲”**为你介绍。然后你可能还会有一个疑问，`/dev/sda5` 究竟是一块磁盘还是别的什么？这个时候你可以用 `fdisk -l` 查看，结果如下图：

```
Disk /dev/sda: 30 GiB, 32212254720 bytes, 62914560 sectors
Disk model: VMware Virtual S
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x88d993c4

Device     Boot   Start      End  Sectors  Size Id Type
/dev/sda1  *       2048  1050623  1048576  512M  b W95 FAT32
/dev/sda2       1052670 62912511 61859842 29.5G  5 Extended
/dev/sda5       1052672 62912511 61859840 29.5G 83 Linux
```


你可以看到我的 Linux 虚拟机上，有一块 30G 的硬盘（当然是虚拟的）。然后这块硬盘下有 3 个设备（Device）：/dev/sda1, /dev/sda2 和 /dev/sda5。在 Linux 中，数字 1~4 结尾的是主分区，通常一块磁盘最多只能有 4 个主分区用于系统启动。主分区之下，还可以再

分成若干个逻辑分区，4 以上的数字都是逻辑分区。因此 `/dev/sda2` 和 `/dev/sda5` 是主分区包含逻辑分区的关系。

挂载

分区结构最终需要最终挂载到目录上。上面例子中 `/dev/sda5` 分区被挂载到了 `/` 下。 这样在 `/` 创建的文件都属于这个 `/dev/sda5` 分区。 另外， `/dev/sda5` 采用 `ext4` 文件系统。可见**不同的目录可以采用不同的文件系统。**

将一个文件系统映射到某个目录的过程叫作挂载（Mount）。当然这里的文件系统可以是某个分区、某个 USB 设备，也可以是某个读卡器等。你可以用 `mount -l` 查看已经挂载的文件系统。

```
ramroll@u1:/$ mount -l
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
udev on /dev type devtmpfs (rw,nosuid,noexec,relatime,size=8156656k,nr_inode
s=2039164,mode=755)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptm
xmode=000)
tmpfs on /run type tmpfs (rw,nosuid,nodev,noexec,relatime,size=1637016k,mode
=755)
/dev/sda5 on / type ext4 (rw,relatime,errors=remount-ro)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,r
elatime)
```


上图中的 `sysfs``proc``devtmpfs``tmpfs``ext4` 都是不同的文件系统，下面我们来说说它们的作用。

- `sysfs` 让用户通过文件访问和设置设备驱动信息。
- `proc` 是一个虚拟文件系统，让用户可以通过文件访问内核中的进程信息。
- `devtmpfs` 在内存中创造设备文件节点。
- `tmpfs` 用内存模拟磁盘文件。
- `ext4` 是一个通常意义上我们认为的文件系统，也是管理磁盘上文件用的系统。

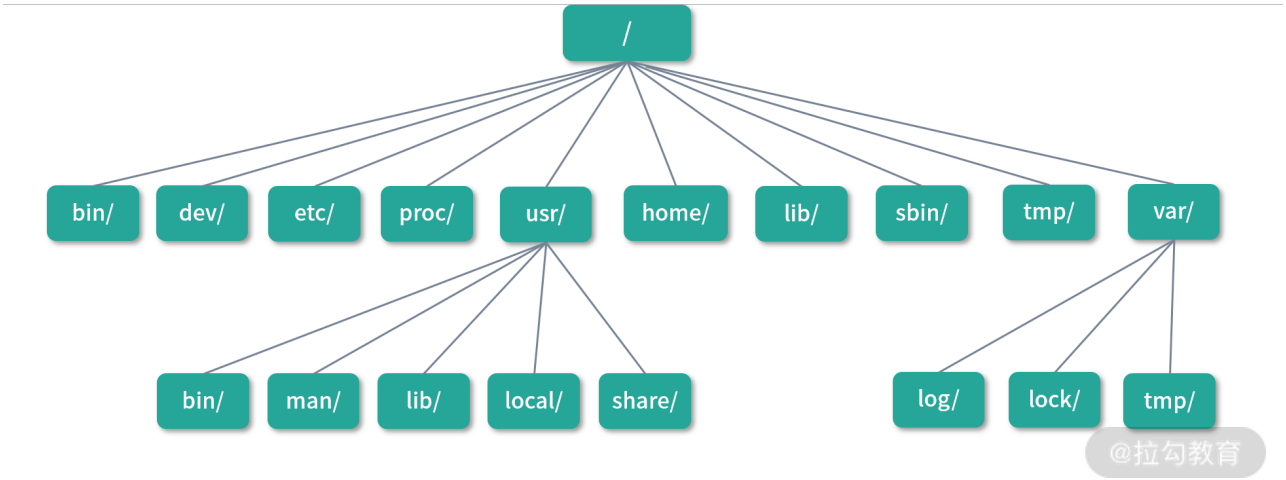
你可以看到挂载记录中不仅有文件系统类型，挂载的目录（on 后面部分），还有读写的权限等。你也可以用 `mount` 指令挂载一个文件系统到某个目录，比如说：

```
mount /dev/sda6 /abc
```

上面这个命令将 `/dev/sda6` 挂载到目录 `abc` 。

目录结构

因为 Linux 内文件系统较多，用途繁杂，Linux 对文件系统中的目录进行了一定的归类，如下图所示：



最顶层的目录称作根目录， 用 `/` 表示。 `/` 目录下用户可以再创建目录，但是有一些目录随着系统创建就已经存在，接下来我会和你一起讨论下它们的用途。

/bin（二进制）包含了许多所有用户都可以访问的可执行文件，如 ls, cp, cd 等。这里的大多数程序都是二进制格式的，因此称作 `bin` 目录。 `bin` 是一个命名习惯，比如说 `nginx` 中的可执行文件会在 Nginx 安装目录的 bin 文件夹下面。

/dev（设备文件） 通常挂载在 `devtmpfs` 文件系统上，里面存放的是设备文件节点。通常直接和内存进行映射，而不是存在物理磁盘上。

值得一提的是其中有几个有趣的文件，它们是虚拟设备。

/dev/null是可以用来销毁任何输出的虚拟设备。你可以用 `>` 重定向符号将任何输出流重定向到 `/dev/null` 来忽略输出的结果。

/dev/zero是一个产生数字 0 的虚拟设备。无论你对它进行多少次读取，都会读到 0。

/dev/ramdom是一个产生随机数的虚拟设备。读取这个文件中数据，你会得到一个随机数。你不停地读取这个文件，就会得到一个随机数的序列。

/etc（配置文件）， `/etc` 名字的含义是 `and so on……`，也就是“等等及其他”，Linux 用它来保管程序的配置。比如说 `mysql` 通常会在 `/etc/mysql` 下创建配置。再比如说 `/etc/passwd` 是系统的用户配置，存储了用户信息。

/proc（进程和内核文件） 存储了执行中进程和内核的信息。比如你可以通过 `/proc/1122` 目

录找到和进程 `1122` 关联的全部信息。还可以在 `/proc/cpuinfo` 下找到和 CPU 相关的全部信息。

/sbin（系统二进制） 和 `/bin` 类似，通常是系统启动必需的指令，也可以包括管理员才会使用的指令。

/tmp（临时文件） 用于存放应用的临时文件，通常用的是 `tmpfs` 文件系统。因为 `tmpfs` 是一个内存文件系统，系统重启的时候清除 `/tmp` 文件，所以这个目录不能放应用和重要的数据。

/var （Variable data file,，可变数据文件） 用于存储运行时的数据，比如日志通常会存放在 `/var/log` 目录下面。再比如应用的缓存文件、用户的登录行为等，都可以放到 `/var` 目录下， `/var` 下的文件会长期保存。

/boot（启动） 目录下存放了 Linux 的内核文件和启动镜像，通常这个目录会写入磁盘最头部的分区，启动的时候需要加载目录内的文件。

/opt（Optional Software，可选软件） 通常会把第三方软件安装到这个目录。以后你安装软件的时候，可以考虑在这个目录下创建。

/root（root 用户家目录） 为了防止误操作，Linux 设计中 root 用户的家目录没有设计在 `/home/root` 下，而是放到了 `/root` 目录。

/home（家目录） 用于存放用户的个人数据，比如用户 `lagou` 的个人数据会存放到 `/home/lagou` 下面。并且通常在用户登录，或者执行 `cd` 指令后，都会在家目录下工作。 用户通常会对自己的家目录拥有管理权限，而无法访问其他用户的家目录。

/media（媒体） 自动挂载的设备通常会出现在 `/media` 目录下。比如你插入 U 盘，通常较新版本的 Linux 都会帮你自动完成挂载，也就是在 `/media` 下创建一个目录代表 U 盘。

/mnt（Mount，挂载） 我们习惯把手动挂载的设备放到这个目录。比如你插入 U 盘后，如果 Linux 没有帮你完成自动挂载，可以用 `mount` 命令手动将 U 盘内容挂载到 `/mnt` 目录下。

/svr（Service Data,，服务数据） 通常用来存放服务数据，比如说你开发的网站资源文件（脚本、网页等）。不过现在很多团队的习惯发生了变化， 有的团队会把网站相关的资源放到 `/www` 目录下，也有的团队会放到 `/data` 下。总之，在存放资源的角度，还是比较灵活的。

/usr（Unix System Resource） 包含系统需要的资源文件，通常应用程序会把后来安装的可执行文件也放到这个目录下，比如说

- `vim` 编辑器的可执行文件通常会在 `/usr/bin` 目录下，区别于 `ls` 会在 `/bin` 目录下
- `/usr/sbin` 中会包含有通常系统管理员才会使用的指令。
- `/usr/lib` 目录中存放系统的库文件，比如一些重要的对象和动态链接库文件。
- `/usr/lib` 目录下会有大量的 `.so` 文件，这些叫作 `Shared Object` ，类似 `windows` 下的 `dll` 文件。
- `/usr/share` 目录下主要是文档，比如说 man 的文档都在 `/usr/share/man` 下面。

总结

这一讲我们了解了 Linux 虚拟文件系统的设计，并且熟悉了 Linux 的目录结构。我曾经看到不少程序员把程序装到了 `/home` 目录，也看到过不少程序员将数据放到了 `/root` 目录。这样做并不会带来致命性问题，但是会给其他和你一起工作的同事带来困扰。

今天我们讲到的这些规范是整个世界通用的，如果每个人都能遵循规范的原则，工作起来就会有很好的默契。登录一台 `linux` 服务器，你可以通过目录结构快速熟悉。你可以查阅 `/etc` 下的配置，看看 `/opt` 下装了什么软件，这就是规范的好处。

那么通过这节课的学习，你现在可以尝试来回答本节标题中的试题目：Linux下各个目录有什么作用了吗?

【解析】通常面试官会挑选其中一部分对你进行抽查，如果你快要面试了，再 Review 一下本讲的内容吧。

上一页 下一页