**Linux 文件系统概览**

| | | | |
|---|---|---|---|
| **Notebook:** | 微信 | | |
| **Created:** | 9/22/2017 13:06 | **Updated:** | 9/24/2017 13:26 |
| **Tags:** | 微信 | | |
| **URL:** | http://mp.weixin.qq.com/s?__biz=MiM5NiO4MiYwMO==&mid=2664609474&idx=1&... | | |

# Linux 文件系统概览

原创 2017-09-22 译者：ucasFL Linux中国


本文旨在高屋建瓴地来讨论 Linux 文件系统概念，而不是对某种特定的文件系统，比如 FXT4 是如何工作的进行具体的描述。另外，本文也不是一个文件系统命令的教程。

-- David Both

本文导航

编译自 | https://opensource.com/life/16/10/introduction-linux-filesystems
作者 | David Both
译者 | ucasFL

本文旨在高屋建瓴地来讨论 Linux 文件系统概念，而不是对某种特定的

文件系统，比如 EXT4 是如何工作的进行具体的描述。另外，本文也不是一个文件系统命令的教程。

每台通用计算机都需要将各种数据存储在硬盘驱动器（HDD）或其他类似设备上，比如 USB 存储器。这样做有两个原因。首先，当计算机关闭以后，内存（RAM）会失去存于它里面的内容。尽管存在非易失类型的 RAM，在计算机断电以后还能把数据存储下来（比如采用 USB 闪存和固态硬盘的闪存），但是，闪存和标准的、易失性的 RAM，比如 DDR3 以及其他相似类型的 RAM 相比，要贵很多。

数据需要存储在硬盘驱动上的另一个原因是，即使是标准的 RAM 也要比普通硬盘贵得多。尽管 RAM 和硬盘的价格都在迅速下降，但是 RAM 的价格依旧在以字节为单位来计算。让我们进行一个以字节为单位的快速计算：基于 16 GB 大的 RAM 的价格和 2 TB 大的硬盘驱动的价格。计算显示 RAM 的价格大约比硬盘驱动贵 71 倍。今天，一个典型的 RAM 的价格大约是 0.000000004373750 美元/每字节。

直观的展示一下在很久以前 RAM 的价格，在计算机发展的非常早的时期，其中一种类型的 RAM 是基于在 CRT 屏幕上的点。这种 RAM 非常昂贵，大约 1 美元/每字节。

## 定义

你可能听过其他人以各种不同和令人迷惑的方式谈论过文件系统。文件系统这个单词本身有多重含义，你需要从一个讨论或文件的上下文中理解它的正确含义。

我将根据我所观察到的在不同情况下使用"文件系统"这个词来定义它的不同含义。注意，尽管我试图遵循标准的"官方"含义，但是我打算基于它的不同用法来定义这个术语（如下）。这就是说我将在本文的后续章节中进

行更详细的探讨。

- 始于顶层 root（ / ）目录的整个 Linux 目录结构。
- 特定类型的数据存储格式，比如 EXT3、EXT4、BTRFS 以及 XFS 等等。Linux 支持近百种类型的文件系统，包括一些非常老的以及一些最新的。每一种文件系统类型都使用它自己独特的元数据结构来定义数据是如何存储和访问的。
- 用特定类型的文件系统格式化后的分区或逻辑卷，可以挂载到 Linux 文件系统的指定挂载点上。

# 文件系统的基本功能

磁盘存储是文件系统必须的功能，它与之伴生的有一些有趣而且不可或缺的细节。很明显，文件系统是用来为非易失数据的存储提供空间，这是它的基本功能。然而，它还有许多从需求出发的重要功能。

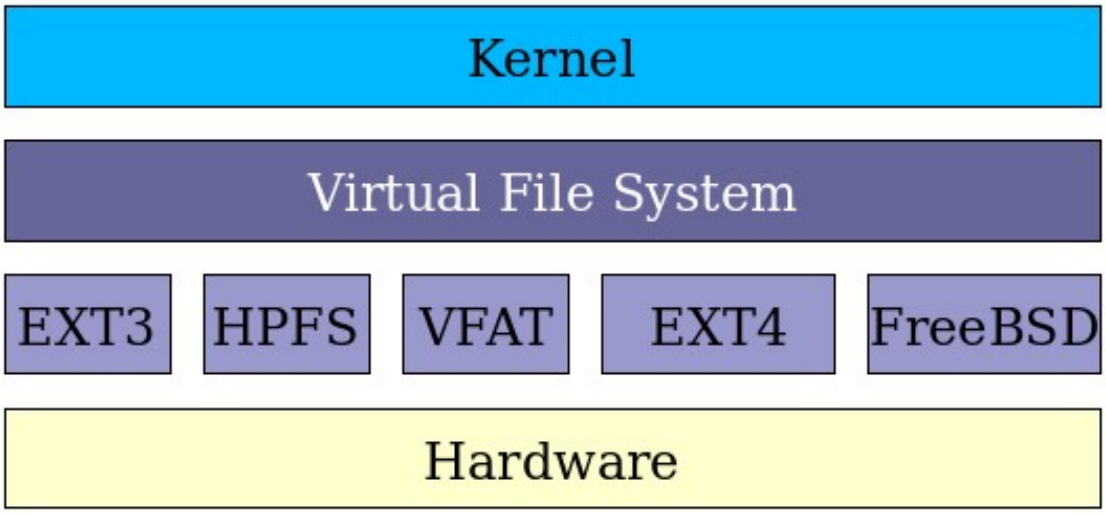
所有文件系统都需要提供一个名字空间，这是一种命名和组织方法。它定义了文件应该如何命名、文件名的最大长度，以及所有可用字符集中可用于文件名中字符集子集。它也定义了一个磁盘上数据的逻辑结构，比如使用目录来组织文件而不是把所有文件聚集成一个单一的、巨大的文件混合体。

定义名字空间以后，元数据结构是为该名字空间提供逻辑基础所必须的。这包括所需数据结构要能够支持分层目录结构，同时能够通过结构来确定硬盘空间中的块是已用的或可用的，支持修改文件或目录的名字，提供关于文件大小、创建时间、最后访问或修改时间等信息，以及位置或数据所属的文件在磁盘空间中的位置。其他的元数据用来存储关于磁盘细分的高级信息，比如逻辑卷和分区。这种更高层次的元数据以及它所代表的结构包含描述文件系统存储在驱动器或分区中的信息，但与文件系统元数据无关，与之独立。

文件系统也需要一个应用程序接口（API），从而提供了对文件系统对象，比如文件和目录进行操作的系统功能调用的访问。API 也提供了诸如创建、移动和删除文件的功能。它也提供了算法来确定某些信息，比如文件存于文件系统中的位置。这样的算法可以用来解释诸如磁盘速度和最小化磁盘碎片等术语。

现代文件系统还提供一个安全模型，这是一个定义文件和目录的访问权限的方案。Linux 文件系统安全模型确保用户只能访问自己的文件，而不能访问其他用户的文件或操作系统本身。

最后一块组成部分是实现这些所有功能所需要的软件。Linux 使用两层软件实现的方式来提高系统和程序员的效率。




*图片 1：Linux 两层文件系统软件实现。*

这两层中的第一层是 Linux 虚拟文件系统。虚拟文件系统提供了内核和开发者访问所有类型文件系统的的单一命令集。虚拟文件系统软件通过调用特殊设备驱动来和不同类型的文件系统进行交互。特定文件系统的设备驱动是第二层实现。设备驱动程序将文件系统命令的标准集解释为在分区或逻辑卷上的特定类型文件系统命令。

# 目录结构

作为一个通常来说非常有条理的处女座，我喜欢将东西存储在更小的、有组织的小容器中，而不是存于同一个大容器中。目录的使用使我能够存储文件并在我想要查看这些文件的时候也能够找到它们。目录也被称为文件夹，之所以被称为文件夹，是因为其中的文件被类比存放于物理桌面上。

在 Linux 和其他许多操作系统中，目录可以被组织成树状的分层结构。在 Linux 文件系统层次标准[1]中定义了 Linux 的目录结构（LCTT 译注：可参阅 这篇[2]）。当通过目录引用来访问目录时，更深层目录名字是通过正斜杠（/）来连接，从而形成一个序列，比如 /var/log 和 /var/spool/mail 。这些被称为路径。

下表提供了标准的、众所周知的、预定义的顶层 Linux 目录及其用途的简要清单。

< 如显示不全，请左右滑动 >

| 目录 | 描述 |
| --- | --- |
| **/ (root 文件系统)** | root 文件系统是文件系统的顶<br>统前需要用来启动 Linux 系统<br>剩余文件系统的全部可执行文<br>文件系统作为 root 文件系统的<br>载点上。 |
| **/bin** | /bin 目录包含用户的可执行 |
| /boot | 包含启动 Linux 系统所需要的<br>置文件。 |
| **/dev** | 该目录包含每一个连接到系统的<br>设备驱动，而是代表计算机上的 |
| **/etc** | 包含主机计算机的本地系统配置 |
| /home | 主目录存储用户文件，每一个用<br>子目录（作为其主目录）。 |
| **/lib** | 包含启动系统所需要的共享库文 |
| /media | 一个挂载外部可移动设备的地方<br>动器。 |
| /mnt | 一个普通文件系统的临时挂载点<br>一个文件系统进行修复或在其上 |
| /opt | 可选文件，比如供应商提供的应 |
| **/root** | 这不是 root（/）文件系统 |
| **/sbin** | 系统二进制文件。这些是用于系<br>临时目录。被操作系统和许多程 |

| | |
|---|---|
| /tmp | 时在这儿存储文件。注意，存<br>通知的情况下被删除。 |
| /usr | 该目录里面包含可共享的、只<br>库、man 文件以及其他类型的 |
| /var | 可变数据文件存储在这儿。这<br>数据库的文件、Web 服务器的 |

表 1：Linux 文件系统层次结构的顶层

这些目录以及它们的子目录如表 1 所示，在所有子目录中，粗体的目录组成了 root 文件系统的必需部分。也就是说，它们不能创建为一个分离的文件系统并且在开机时进行挂载。这是因为它们（特别是它们包含的内容）必须在系统启动的时候出现，从而系统才能正确启动。

/media 目录和 /mnt 目录是 root 文件系统的一部分，但是它们从来不包含任何数据，因为它们只是一个临时挂载点。

表 1 中剩下的非粗体的目录不需要在系统启动过程中出现，但会在之后挂载到 root 文件系统上，在开机阶段，它们为主机进行准备，从而执行有用的工作。

请参考官方 Linux 文件系统层次标准[3]（FHS）网页来了解这些每一个目录以及它们的子目录的更多细节。维基百科上也有关于 FHS[4] 的一个很好的介绍。应该尽可能的遵循这些标准，从而确保操作和功能的一致性。无论在主机上使用什么类型的文件系统，该层次目录结构都是相同的。

## Linux 统一目录结构

在一些非 Linux 操作系统的个人电脑上，如果有多个物理硬盘驱动器或多个分区，每一个硬盘或分区都会分配一个驱动器号。知道文件或程序位

于哪一个硬盘驱动器上是很有必要的，比如 C: 或 D: 。然后，你可以在命令中使用驱动器号，以 D: 为例，为了进入 D: 驱动器，你可以使用 cd 命令来更改工作目录为正确的目录，从而定位需要的文件。每一个硬盘驱动器都有自己单独的、完整的目录树。

Linux 文件系统将所有物理硬盘驱动器和分区统一为一个目录结构。它们均从顶层 root 目录（ / ）开始。所有其它目录以及它们的子目录均位于单一的 Linux 根目录下。这意味着只有一棵目录树来搜索文件和程序。

因为只有一个文件系统，所以 /home 、 /tmp 、 /var 、 /opt 或 /usr 能够创建在和 root（ / ）文件系统不同的物理硬盘驱动器、分区或逻辑分区上，然后挂载到一个挂载点（目录）上，从而作为 root 文件系统树的一部分。甚至可移动驱动器，比如 USB 驱动器或一个外接的 USB 或 ESATA 硬盘驱动器均可以挂载到 root 文件系统上，成为目录树不可或缺的部分。

当从 Linux 发行版的一个版本升级到另一个版本或从一个发行版更改到另一个发行版的时候，就会很清楚地看到这样创建到不同分区的好处。通常情况下，除了任何像 Fedora 中的 dnf-upgrade 之类的升级工具，会明智地在升级过程中偶尔重新格式化包含操作系统的硬盘驱动来删除那些长期积累的垃圾。如果 /home 目录是 root 文件系统的一部分（位于同一个硬盘驱动器），那么它也会被格式化，然后需要通过之前的备份恢复。如果 /home 目录作为一个分离的文件系统，那么安装程序将会识别到，并跳过它的格式化。对于存储数据库、邮箱、网页和其它可变的用户以及系统数据的 /var 目录也是这样的。

将 Linux 系统目录树的某些部分作为一个分离的文件系统还有一些其他原因。比如，在很久以前，我还不知道将所有需要的 Linux 目录均作为 root（ / ）文件系统的一部分可能存在的问题，于是，一些非常大的文件填满了 /home 目录。因为 /home 目录和 /tmp 目录均不是分离的文件系统，而是 root 文件系统的简单子目录，整个 root 文件系统就被填满了。于是就不再有剩余空间可以让操作系统用来存储临时文件或

扩展已存在数据文件。首先，应用程序开始抱怨没有空间来保存文件，然后，操作系统也开始异常行动。启动到单用户模式，并清除了 /home 目录中的多余文件之后，终于又能够重新工作了。然后，我使用非常标准的多重文件系统设置来重新安装 Linux 系统，从而避免了系统崩溃的再次发生。

我曾经遇到一个情况，Linux 主机还在运行，但是却不允许用户通过 GUI 桌面登录。我可以通过使用虚拟控制台[5]之一，通过命令行界面（CLI）本地登录，然后远程使用 SSH 。问题的原因是因为 /tmp 文件系统满了，因此 GUI 桌面登录时所需要的一些临时文件不能被创建。因为命令行界面登录不需要在 /tmp 目录中创建文件，所以无可用空间并不会阻止我使用命令行界面来登录。在这种情况下，/tmp 目录是一个分离的文件系统，在 /tmp 所位于的逻辑卷上还有大量的可用空间。我简单地扩展了 /tmp 逻辑卷[6]的容量到能够容纳主机所需要的临时文件，于是问题便解决了。注意，这个解决方法不需要重启，当 /tmp 文件系统扩大以后，用户就可以登录到桌面了。

当我在一家很大的科技公司当实验室管理员的时候，遇到过另外一个故障。开发者将一个应用程序安装到了一个错误的位置（ /var ）。结果该应用程序崩溃了，因为 /var 文件系统满了，由于缺乏空间，存储于 /var/log 中的日志文件无法附加新的日志消息。然而，系统仍然在运行，因为 root 文件系统和 /tmp 文件系统还没有被填满。删除了该应用程序并重新安装在 /opt 文件系统后，问题便解决了。

# 文件系统类型

Linux 系统支持大约 100 种分区类型的读取，但是只能对很少的一些进行创建和写操作。但是，可以挂载不同类型的文件系统在同一个 root 文件系统上，并且是很常见的。在这样的背景下，我们所说的文件系统一词

是指在硬盘驱动器或逻辑卷上的一个分区中存储和管理用户数据所需要的结构和元数据。能够被 Linux 系统的 fdisk 命令识别的文件系统类型的完整列表在此[7]，你可以感受一下 Linux 系统对许多类型的系统的高度兼容性。

Linux 支持读取这么多类型的分区系统的主要目的是为了提高兼容性，从而至少能够与一些其他计算机系统的文件系统进行交互。下面列出了在 Fedora 中创建一个新的文件系统时的所有可选类型：

- btrfs
- **cramfs**
- **ext2**
- **ext3**
- **ext4**
- fat
- gfs2
- hfsplus
- minix
- **msdos**
- ntfs
- reiserfs
- **vfat**
- xfs

其他发行版支持创建的文件系统类型不同。比如，CentOS 6 只支持创建上表中标为黑体的文件系统类型。

## 挂载

在 Linux 系统上""文件系统的术语是指在计算机发展的早期，磁带或可移动的磁盘组需要需要物理地挂载到一个合适的驱动器设备上。当通过物理的方式放置到驱动器上以后，操作系统会逻辑地挂载位于磁盘上的文件系

统，从而操作系统、应用程序和用户才能够访问文件系统中的内容。

一个挂载点简单的来说就是一个目录，就像任何其它目录一样，是作为 root 文件系统的一部分创建的。所以，比如，home 文件系统是挂载在目录 /home 下。文件系统可以被挂载到其他非 root 文件系统的挂载点上，但是这并不常见。

在 Linux 系统启动阶段的最初阶段，root 文件系统就会被挂载到 root 目录下（ / ）。其它文件系统在之后通过 SystemV 下的 rc 或更新一些的 Linux 发行版中的 systemd 等 Linux 启动程序挂载。在启动进程中文件系统的挂载是由 /etc/fstab 配置文件管理的。一个简单的记忆方法是，fstab 代表“”，它包含了需要挂载的文件系统的列表，这些文件系统均指定了挂载点，以及针对特定文件系统可能需要的选项。

使用 mount 命令可以把文件系统挂载到一个已有的目录/挂载点上。通常情况下，任何作为挂载点的目录都应该是空的且不包含任何其他文件。Linux 系统不会阻止用户挂载一个已被挂载了文件系统的目录或将文件系统挂载到一个包含文件的目录上。如果你将文件系统挂载到一个已有的目录或文件系统上，那么其原始内容将会被隐藏，只有新挂载的文件系统的内容是可见的。