

# Linux 系统概论

天津医科大学  
生物医学工程与技术学院

2017-2018 学年下学期（春）  
2016 级生信班

# 第三章 文件系统 (集体备课)

伊现富 (Yi Xianfu)

天津医科大学 (TIJMU)  
生物医学工程与技术学院

2015 年 4 月 28 日



# 教学提纲

- 1 指导思想
- 2 引言
- 3 文件系统基础
  - 文件系统和分区
  - 目录结构
  - 路径
- 4 文件系统导航
  - 目录操作
  - 文件操作
  - 文件系统管理
- 5 文件类型
  - 命令详解
  - 类型简介
  - 链接
- 6 文件和目录权限
  - 权限简介
  - 修改权限
- 7 挂载文件系统
- 8 回顾与总结
  - 总结
  - 思考题

## 1 指导思想

## 2 引言

## 3 文件系统基础

- 文件系统和分区
- 目录结构
- 路径

## 4 文件系统导航

- 目录操作
- 文件操作
- 文件系统管理

## ● 命令详解

## 5 文件类型

- 类型简介
- 链接

## 6 文件和目录权限

- 权限简介
- 修改权限

## 7 挂载文件系统

## 8 回顾与总结

- 总结
- 思考题



顺序	授课内容	教材章节	学时	日期
1	Linux 基础	第 1、2 章	2	5.12
2	用户和组	第 3 章	2	5.19
3	文件系统	第 4 章	2	5.26
4	Linux 命令	第 6 章	2	6.02
5	高级 Linux 命令	第 8、9 章	2	6.09
6	软件安装	第 19 章	2	6.16
7	vi/Vim 编辑器	第 7 章	2	6.23
8	shell 脚本编程	第 13、14 章	2	6.30
9	Perl 语言简介	第 17 章	2	7.07



## 内容

- 引言与导入
- 文件系统基础
- 文件系统导航
- 文件类型
- 文件和目录权限
- 挂载文件系统
- 总结与答疑

## 学时（分钟）

- 5
- 20
- 25
- 20
- 20
- 5
- 5



## 内容特点

- 知识新颖
- 内容抽象
- 命令繁杂
- 缩写泛滥
- 学以致用

## 对应策略

- 温故知新 (熟悉  $\Rightarrow$  陌生)
- 类比举例 (死记  $\Rightarrow$  理解)
- 详略得当 (讲授  $\Rightarrow$  自学)
- 全称助记 (表象  $\Rightarrow$  本质)
- 学练结合 (理论  $\Rightarrow$  实践)



1 指导思想

2 引言

3 文件系统基础

- 文件系统和分区
- 目录结构
- 路径

4 文件系统导航

- 目录操作
- 文件操作
- 文件系统管理

● 命令详解

5 文件类型

- 类型简介
- 链接

6 文件和目录权限

- 权限简介
- 修改权限

7 挂载文件系统

8 回顾与总结

- 总结
- 思考题





计算机的**文件系统 (File system)** 是一种存储和组织计算机数据的方法，它使得对其访问和查找变得容易，文件系统使用**文件**和**树形目录**的抽象逻辑概念代替了硬盘和光盘等物理设备使用数据块的概念，用户使用文件系统来保存数据不必关心数据实际保存在硬盘（或者光盘）的地址为多少的数据块上，只需要记住这个文件的所属目录和文件名。在写入新数据之前，用户不必关心硬盘上的那个块地址有没有被使用，硬盘上的存储空间管理（分配和释放）功能由文件系统自动完成，用户只需要记住数据被写入到了哪个文件中。

严格地说，文件系统是一套实现了数据的存储、分级组织、访问和获取等操作的抽象数据类型 (Abstract data type) 。

文件系统通常使用硬盘和光盘这样的存储设备，并维护文件在设备中的物理位置。但是，实际上文件系统也可能仅仅是一种访问数据的界面而已，实际的数据是通过网络协议（如 NFS、SMB、9P 等）提供的或者存储在内存中，甚至可能根本没有对应的文件（如 proc 文件系统）



## 操作系统 vs. 文件系统

- 终端用户  $\longleftrightarrow$  操作系统  $\longleftrightarrow$  计算机硬件
- 终端用户  $\longleftrightarrow$  文件系统  $\longleftrightarrow$  硬盘等存储设备



## 引入

- 在 Windows 中，把硬盘/U 盘格式化为 FAT32 或 NTFS 格式。
- 16G 的 U 盘却装不下 4G 的电影！原因何在？

- 面向磁盘的文件系统（本地的文件系统）：位于硬盘、移动硬盘、光盘、U 盘或其他设备上的实际可访问的文件系统。
- 面向网络的文件系统（基于网络的文件系统）：可以远程访问的文件系统。如：NFS、Samba。
- 专用的或虚拟的文件系统：没有实际驻留在磁盘上的文件系统。如：TMPFS、PROXFS。



## 引入

- 在 Windows 中，把硬盘/U 盘格式化为 FAT32 或 NTFS 格式。
  - 16G 的 U 盘却装不下 4G 的电影！原因何在？
- ① 面向磁盘的文件系统（本地的文件系统）：位于硬盘、移动硬盘、光盘、U 盘或其他设备上的实际可访问的文件系统。
    - FAT、NTFS：Windows
    - EXT4、Btrfs、XFS：Linux
    - ISO9660：CD-ROM
    - UFS：Unix
  - ② 面向网络的文件系统（基于网络的文件系统）：可以远程访问的文件系统。如：NFS、Samba。
  - ③ 专用的或虚拟的文件系统：没有实际驻留在磁盘上的文件系统。如：TMPFS、PROXFS。



## 引入

- 在 Windows 中，把硬盘/U 盘格式化为 FAT32 或 NTFS 格式。
  - 16G 的 U 盘却装不下 4G 的电影！原因何在？
- ① 面向磁盘的文件系统（本地的文件系统）：位于硬盘、移动硬盘、光盘、U 盘或其他设备上的实际可访问的文件系统。
    - FAT、NTFS：Windows
    - EXT4、Btrfs、XFS：Linux
    - ISO9660：CD-ROM
    - UFS：Unix
  - ② 面向网络的文件系统（基于网络的文件系统）：可以远程访问的文件系统。如：NFS、Samba。
  - ③ 专用的或虚拟的文件系统：没有实际驻留在磁盘上的文件系统。如：TMPFS、PROXFS。



## 引入

- 在 Windows 中，把硬盘/U 盘格式化为 FAT32 或 NTFS 格式。
  - 16G 的 U 盘却装不下 4G 的电影！原因何在？
- ① 面向磁盘的文件系统（本地的文件系统）：位于硬盘、移动硬盘、光盘、U 盘或其他设备上的实际可访问的文件系统。
    - FAT、NTFS：Windows
    - EXT4、Btrfs、XFS：Linux
    - ISO9660：CD-ROM
    - UFS：Unix
  - ② 面向网络的文件系统（基于网络的文件系统）：可以远程访问的文件系统。如：NFS、Samba。
  - ③ 专用的或虚拟的文件系统：没有实际驻留在磁盘上的文件系统。如：TMPFS、PROXFS。



## 引入

- 在 Windows 中，把硬盘/U 盘格式化为 FAT32 或 NTFS 格式。
  - 16G 的 U 盘却装不下 4G 的电影！原因何在？
- ① 面向磁盘的文件系统（本地的文件系统）：位于硬盘、移动硬盘、光盘、U 盘或其他设备上的实际可访问的文件系统。
    - FAT、NTFS：Windows
    - **EXT4**、Btrfs、XFS：Linux
    - **ISO9660**：CD-ROM
    - UFS：Unix
  - ② 面向网络的文件系统（基于网络的文件系统）：可以远程访问的文件系统。如：NFS、Samba。
  - ③ 专用的或虚拟的文件系统：没有实际驻留在磁盘上的文件系统。如：TMPFS、PROXFS。



1 指导思想

2 引言

3 文件系统基础

- 文件系统和分区
- 目录结构
- 路径

4 文件系统导航

- 目录操作
- 文件操作
- 文件系统管理

● 命令详解

5 文件类型

- 类型简介
- 链接

6 文件和目录权限

- 权限简介
- 修改权限

7 挂载文件系统

8 回顾与总结

- 总结
- 思考题





1 指导思想

2 引言

3 文件系统基础

- 文件系统和分区
- 目录结构
- 路径

4 文件系统导航

- 目录操作
- 文件操作
- 文件系统管理

● 命令详解

5 文件类型

- 类型简介
- 链接

6 文件和目录权限

- 权限简介
- 修改权限

7 挂载文件系统

8 回顾与总结

- 总结
- 思考题



## 文件系统和分区

- 分区是信息的容器，包含整个硬盘或硬盘的一部分
- 文件系统是多个文件的逻辑集合，位于分区或磁盘上
- 一个分区通常只包含一个文件系统

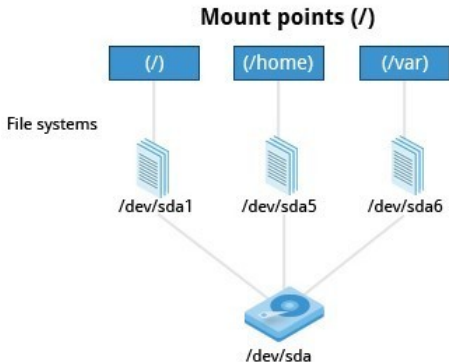
### 类比 Windows

- 一块硬盘
- C/D/E 多个分区
- FAT32、NTFS 不同文件系统



## 文件系统和分区

- 分区是信息的容器，包含整个硬盘或硬盘的一部分
- 文件系统是多个文件的逻辑集合，位于分区或磁盘上
- 一个分区通常只包含一个文件系统



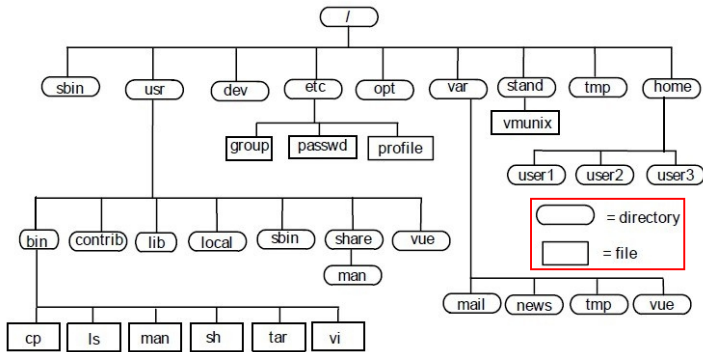
## 类比 Windows

- 一块硬盘
- C/D/E 多个分区
- FAT32、NTFS 不同文件系统



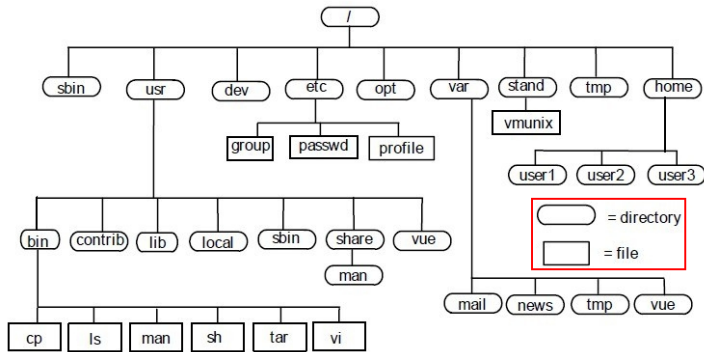
- 1 指导思想
- 2 引言
- 3 文件系统基础
  - 文件系统和分区
  - 目录结构
  - 路径
- 4 文件系统导航
  - 目录操作
  - 文件操作
  - 文件系统管理
- 5 命令详解
- 6 文件类型
  - 类型简介
  - 链接
- 7 文件和目录权限
  - 权限简介
  - 修改权限
- 8 挂载文件系统
- 9 回顾与总结
  - 总结
  - 思考题





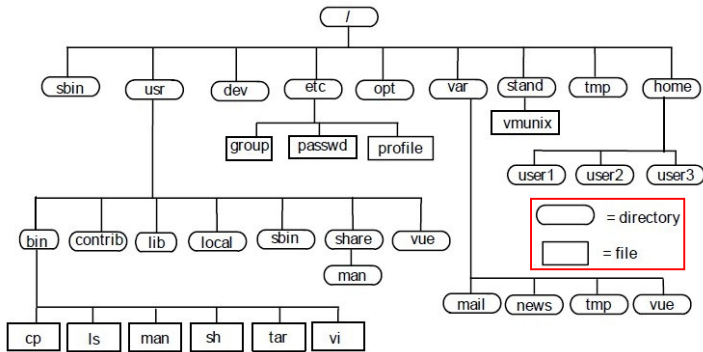
- Everything is a file. (一切皆文件。)
- 使用自顶而下的分层结构来组织文件
- 每个文件和目录都是从根目录 (/) (Root Directory) 开始的





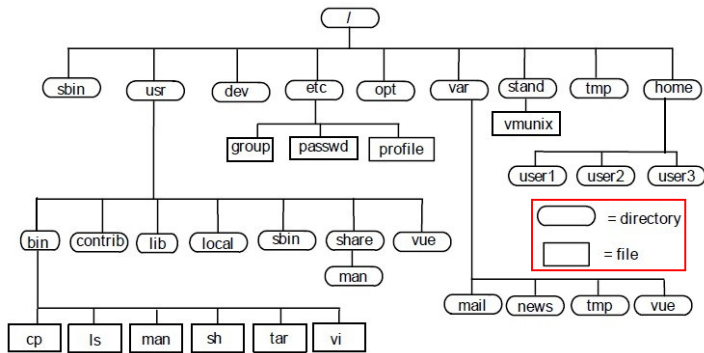
- Everything is a file. (一切皆文件。)
- 使用自顶而下的分层结构来组织文件
- 每个文件和目录都是从根目录 (/) (Root Directory) 开始的
- 文件和目录名的大小写是有区别的
- 定位文件：(根) 目录 ⇒ 子目录 ⇒ ... ⇒ 文件





- Everything is a file. (一切皆文件。)
- 使用自顶而下的分层结构来组织文件
- 每个文件和目录都是从根目录 (/) (Root Directory) 开始的
- 文件和目录名的大小写是有区别的
- 定位文件：(根) 目录 ⇒ 子目录 ⇒ ... ⇒ 文件

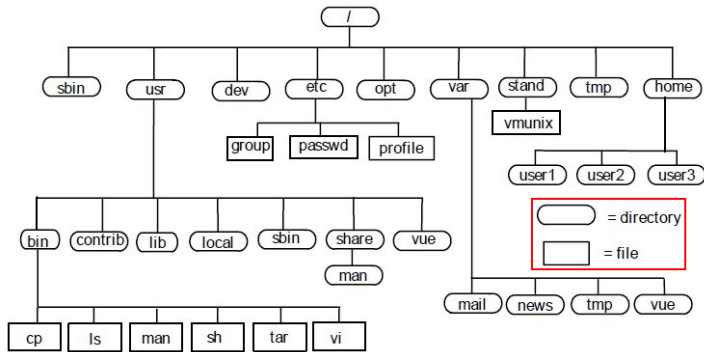




- Everything is a file. (一切皆文件。)
- 使用自顶而下的分层结构来组织文件
- 每个文件和目录都是从根目录 (/) (Root Directory) 开始的
- 文件和目录名的大小写是有区别的
- 定位文件：(根) 目录 ⇒ 子目录 ⇒ ... ⇒ 文件

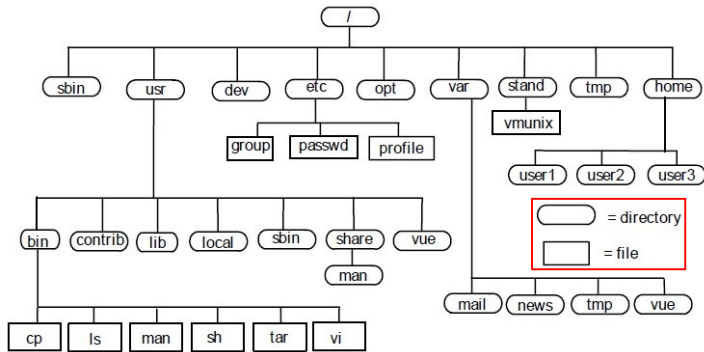






- Everything is a file. (一切皆文件。)
- 使用自顶而下的分层结构来组织文件
- 每个文件和目录都是从根目录 (/) (Root Directory) 开始的
- 文件和目录名的大小写是有区别的
- 定位文件：(根) 目录 ⇒ 子目录 ⇒ ... ⇒ 文件

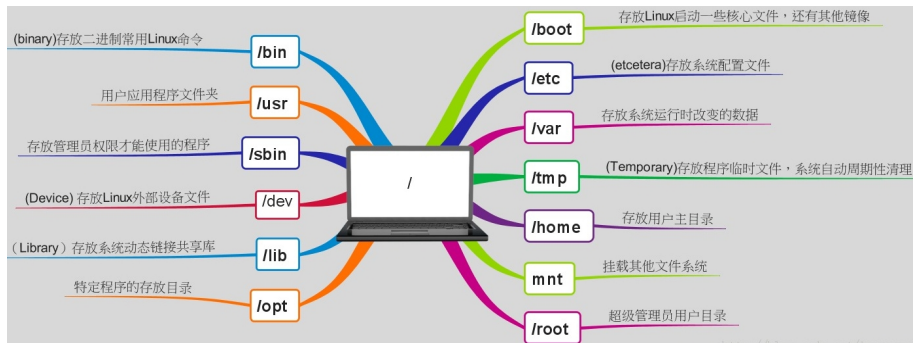




- Everything is a file. (一切皆文件。)
- 使用自顶而下的分层结构来组织文件
- 每个文件和目录都是从根目录 (/) (Root Directory) 开始的
- 文件和目录名的大小写是有区别的
- 定位文件：(根) 目录 ⇒ 子目录 ⇒ ... ⇒ 文件



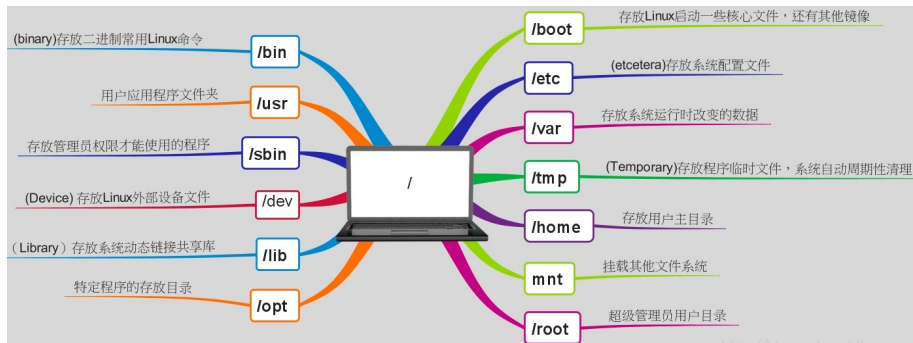
# 文件系统 | 基础 | 目录结构 | 基本目录



## 全称/助记

bin: binary; dev: device; lib: library; mnt: mount; proc: process;  
etc: etcetera => Extended Tool Chest, Editable Text Configuration;  
opt: optional; sbin: system binary; srv: service; tmp: temporary;  
usr: user => User System/Software Resources; var: variable.

# 文件系统 | 基础 | 目录结构 | 基本目录



## 全称/助记

bin: binary; dev: device; lib: library; mnt: mount; proc: process;  
etc: etcetera => Extended Tool Chest, Editable Text Configuration;  
opt: optional; sbin: system binary; srv: service; tmp: temporary;  
usr: user => User System/Software Resources; var: variable.

目录	内容
/	根目录
/bin	基本程序
/boot	启动系统时所需的文件
/dev	设备文件
/etc	配置文件
/home	用户的 home 目录
/lib	基本共享库, 内核模块
/lost+found	由 fsck 恢复的受损文件
/media	可移动介质的挂载点
/mnt	不能挂载在其他位置上的固定介质的挂载点



目录	内容
/opt	第三方应用程序（“可选软件”）
/proc	proc 文件
/root	根用户（超级用户）的 home 目录
/sbin	由超级用户运行的基本系统管理程序
/srv	本地系统所提供服务的的数据
/tmp	临时文件
/usr	静态数据使用的辅助文件系统
/var	可变数据使用的辅助文件系统



目录	内容
/usr/bin	非基本程序（大多数用户程序）
/usr/games	游戏等娱乐和教育程序
/usr/include	C 程序的头文件
/usr/lib	非基本共享库
/usr/local	本地安装程序
/usr/sbin	由超级用户运行的非基本系统管理程序
/usr/share	共享系统数据
/usr/src	源代码（只用于参考）



1 指导思想

2 引言

3 文件系统基础

- 文件系统和分区
- 目录结构
- 路径

4 文件系统导航

- 目录操作
- 文件操作
- 文件系统管理

● 命令详解

5 文件类型

- 类型简介
- 链接

6 文件和目录权限

- 权限简介
- 修改权限

7 挂载文件系统

8 回顾与总结

- 总结
- 思考题





## 绝对 vs. 相对（和现实生活中的定位方式相类比）

- 绝对路径（Absolute Path）：文件在文件系统中的精确位置，总是起始于 root (/)
- 相对路径（Relative Path）：相对于用户当前位置的一个文件或目录的位置

## 相对路径

- .：当前目录
- ..：上一层目录
- ~：当前用户的家目录（Home Directory）
- -：上一个工作目录



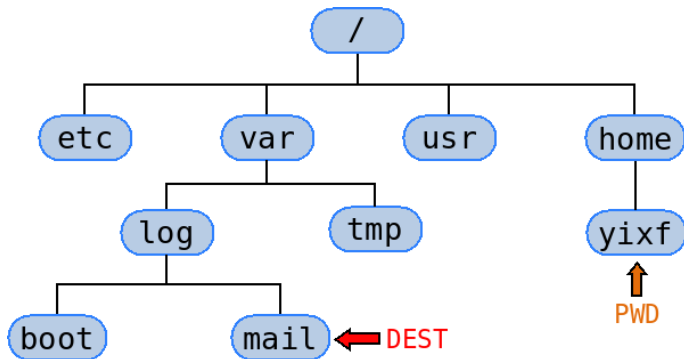
## 绝对 vs. 相对（和现实生活中的定位方式相类比）

- 绝对路径（Absolute Path）：文件在文件系统中的精确位置，总是起始于 root (/)
- 相对路径（Relative Path）：相对于用户当前位置的一个文件或目录的位置

## 相对路径

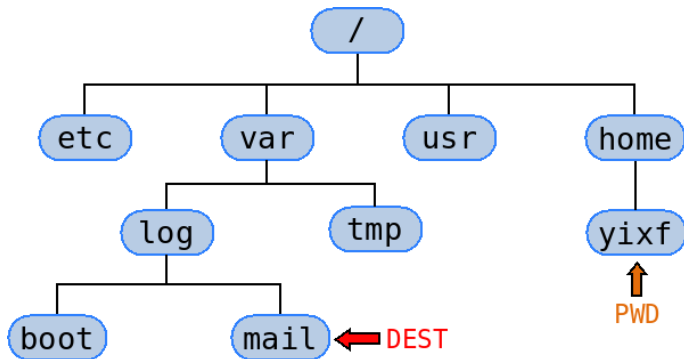
- .：当前目录
- ..：上一层目录
- ~：当前用户的家目录（Home Directory）
- -：上一个工作目录





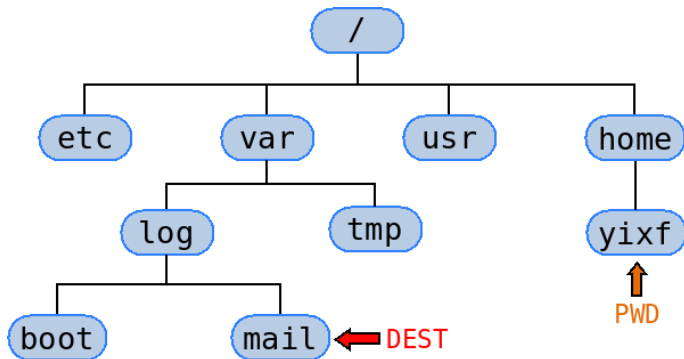
## 绝对 vs. 相对 (比较两者的优缺点)

- 绝对路径: `/var/log/mail`; 精确 vs. 冗长
- 相对路径: `../../../../var/log/mail`; (多数时候) 简短 vs. 隐患



## 绝对 vs. 相对 (比较两者的优缺点)

- 绝对路径：`/var/log/mail`；精确 vs. 冗长
- 相对路径：`../../../../var/log/mail`；(多数时候) 简短 vs. 隐患



## 绝对 vs. 相对 (比较两者的优缺点)

- 绝对路径：`/var/log/mail`；精确 vs. 冗长
- 相对路径：`../../../../var/log/mail`；(多数时候) 简短 vs. 隐患

## 和 Windows 进行比较

- 一切都源于根目录 (/)
- 文件名除了 / 之外，所有的字符都合法
- 有些字符最好不用，如空格符、制表符、退格符和 @ # \$ % & ( ) - 等字符
- 避免使用 . 作为普通文件名的第一个字符（隐藏文件）
- 大小写敏感，Linux 是区分大小写的操作系统
- real\_file、Real\_file、REAL\_FILE 是三个不同的文件名
- 按惯例文件名都是小写的



## 和 Windows 进行比较

- 一切都源于根目录 (/)
- 文件名除了 / 之外，所有的字符都合法
- 有些字符最好不用，如空格符、制表符、退格符和 @ # \$ % & ( ) - 等字符
- 避免使用 . 作为普通文件名的第一个字符（隐藏文件）
- 大小写敏感，Linux 是区分大小写的操作系统
- real\_file、Real\_file、REAL\_FILE 是三个不同的文件名
- 按惯例文件名都是小写的



## 和 Windows 进行比较

- 一切都源于根目录 (/)
- 文件名除了 / 之外，所有的字符都合法
- 有些字符最好不用，如空格符、制表符、退格符和 @ # \$ % & ( ) - 等字符
- 避免使用 . 作为普通文件名的第一个字符（隐藏文件）
- 大小写敏感，Linux 是区分大小写的操作系统
- real\_file、Real\_file、REAL\_FILE 是三个不同的文件名
- 按惯例文件名都是小写的





## 和 Windows 进行比较

- 一切都源于根目录 (/)
- 文件名除了 / 之外，所有的字符都合法
- 有些字符最好不用，如空格符、制表符、退格符和 @ # \$ % & ( ) - 等字符
- 避免使用 . 作为普通文件名的第一个字符（隐藏文件）
- 大小写敏感，Linux 是区分大小写的操作系统
- real\_file、Real\_file、REAL\_FILE 是三个不同的文件名
- 按惯例文件名都是小写的



## 和 Windows 进行比较

- 一切都源于根目录 (/)
- 文件名除了 / 之外，所有的字符都合法
- 有些字符最好不用，如空格符、制表符、退格符和 @ # \$ % & ( ) - 等字符
- 避免使用 . 作为普通文件名的第一个字符（隐藏文件）
- 大小写敏感，Linux 是区分大小写的操作系统
- real\_file、Real\_file、REAL\_FILE 是三个不同的文件名
- 按惯例文件名都是小写的



## 和 Windows 进行比较

- 一切都源于根目录 (/)
- 文件名除了 / 之外，所有的字符都合法
- 有些字符最好不用，如空格符、制表符、退格符和 @ # \$ % & ( ) - 等字符
- 避免使用 . 作为普通文件名的第一个字符（隐藏文件）
- 大小写敏感，Linux 是区分大小写的操作系统
- real\_file、Real\_file、REAL\_FILE 是三个不同的文件名
- 按惯例文件名都是小写的



## 和 Windows 进行比较

- 一切都源于根目录 (/)
- 文件名除了 / 之外，所有的字符都合法
- 有些字符最好不用，如空格符、制表符、退格符和 @ # \$ % & ( ) - 等字符
- 避免使用 . 作为普通文件名的第一个字符（隐藏文件）
- 大小写敏感，Linux 是区分大小写的操作系统
- real\_file、Real\_file、REAL\_FILE 是三个不同的文件名
- 按惯例文件名都是小写的



1 指导思想

2 引言

3 文件系统基础

- 文件系统和分区
- 目录结构
- 路径

4 文件系统导航

- 目录操作
- 文件操作
- 文件系统管理

● 命令详解

5 文件类型

- 类型简介
- 链接

6 文件和目录权限

- 权限简介
- 修改权限

7 挂载文件系统

8 回顾与总结

- 总结
- 思考题



## 引入

- 文件系统不同，但其中的操作大同小异
- 学生总结 Windows 中的常见操作，老师给出 Linux 中对应的命令
- 给出命令的全称辅助记忆，详细讲解个别重要的命令

### 目录

- 定位, pwd
- 切换, cd
- 列出, ls
- 创建, mkdir
- 删除, rmdir
- 树图, tree

### 文件

- 查看, cat
- 识别, file
- 创建, touch
- 复制, cp
- 移动, mv
- 重命名, mv
- 删除, rm

### 管理

- 查找, find
- 空间, df
- 大小, du



## 引入

- 文件系统不同，但其中的操作大同小异
- 学生总结 Windows 中的常见操作，老师给出 Linux 中对应的命令
- 给出命令的全称辅助记忆，详细讲解个别重要的命令

## 目录

- 定位, pwd
- 切换, cd
- 列出, ls
- 创建, mkdir
- 删除, rmdir
- 树图, tree

## 文件

- 查看, cat
- 识别, file
- 创建, touch
- 复制, cp
- 移动, mv
- 重命名, mv
- 删除, rm

## 管理

- 查找, find
- 空间, df
- 大小, du



## 引入

- 文件系统不同，但其中的操作大同小异
- 学生总结 Windows 中的常见操作，老师给出 Linux 中对应的命令
- 给出命令的全称辅助记忆，详细讲解个别重要的命令

## 目录

- 定位, pwd
- 切换, cd
- 列出, ls
- 创建, mkdir
- 删除, rmdir
- 树图, tree

## 文件

- 查看, cat
- 识别, file
- 创建, touch
- 复制, cp
- 移动, mv
- 重命名, mv
- 删除, rm

## 管理

- 查找, find
- 空间, df
- 大小, du





## 引入

- 文件系统不同，但其中的操作大同小异
- 学生总结 Windows 中的常见操作，老师给出 Linux 中对应的命令
- 给出命令的全称辅助记忆，详细讲解个别重要的命令

## 目录

- 定位, pwd
- 切换, cd
- 列出, ls
- 创建, mkdir
- 删除, rmdir
- 树图, tree

## 文件

- 查看, cat
- 识别, file
- 创建, touch
- 复制, cp
- 移动, mv
- 重命名, mv
- 删除, rm

## 管理

- 查找, find
- 空间, df
- 大小, du



## 1 指导思想

## 2 引言

## 3 文件系统基础

- 文件系统和分区
- 目录结构
- 路径

## 4 文件系统导航

- 目录操作
- 文件操作
- 文件系统管理

## ● 命令详解

## 5 文件类型

- 类型简介
- 链接

## 6 文件和目录权限

- 权限简介
- 修改权限

## 7 挂载文件系统

## 8 回顾与总结

- 总结
- 思考题



命令	助记	说明
pwd	Print Work Directory	显示用户的当前目录
ls	LiSt	列出指定目录的内容
cd	Change Directory	转到指定的目录
mkdir	MaKe DIRectory	创建指定的目录
rmdir	ReMove DIRectory	删除空目录
tree	—	以树状图列出目录的内容结构



- 1 指导思想
- 2 引言
- 3 文件系统基础
  - 文件系统和分区
  - 目录结构
  - 路径
- 4 文件系统导航
  - 目录操作
  - 文件操作
  - 文件系统管理
- 5 命令详解
- 6 文件类型
  - 类型简介
  - 链接
- 7 文件和目录权限
  - 权限简介
  - 修改权限
- 8 挂载文件系统
- 9 回顾与总结
  - 总结
  - 思考题



命令	助记	说明
file	—	识别文件类型（二进制、文本等）
cat	conCATenate	显示一个文件
touch	—	创建一个空文件或者修改一个现有文件的属性
cp	CoPy	把一个文件/目录复制到指定位置
mv	MoVe	移动文件/目录的位置或重命名一个文件/目录
rm	ReMove	删除文件
head	—	显示文件的开始部分
tail	—	显示文件的结尾部分
more	—	从头到尾浏览一个文件
less	—	从开头或结尾开始浏览整个文件



1 指导思想

2 引言

3 文件系统基础

- 文件系统和分区
- 目录结构
- 路径

4 文件系统导航

- 目录操作
- 文件操作
- 文件系统管理

● 命令详解

5 文件类型

- 类型简介
- 链接

6 文件和目录权限

- 权限简介
- 修改权限

7 挂载文件系统

8 回顾与总结

- 总结
- 思考题



命令	助记	说明
which	—	如果文件位于用户的 PATH 内，则显示文件位置
whereis	—	显示文件的位置
find	—	查找文件/目录
df	Disk Free	显示磁盘空间的使用情况
du	Disk Usage	显示目录空间占用情况



1 指导思想

2 引言

3 文件系统基础

- 文件系统和分区
- 目录结构
- 路径

4 文件系统导航

- 目录操作
- 文件操作
- 文件系统管理

● 命令详解

5 文件类型

- 类型简介
- 链接

6 文件和目录权限

- 权限简介
- 修改权限

7 挂载文件系统

8 回顾与总结

- 总结
- 思考题





## 选项（给出全称辅助记忆）

- -p : Preserve, 保持目录和文件的属性
- -R : Recursive, 递归
- -u : Update, 增量备份

cp 妙用：备份目录

cp -Rpu 待备份目录 目标目录



## 选项（给出全称辅助记忆）

- -p : Preserve, 保持目录和文件的属性
- -R : Recursive, 递归
- -u : Update, 增量备份

## cp 妙用：备份目录

cp -Rpu 待备份目录 目标目录



## cd

- `cd` : 返回家目录
- `cd ~` : 返回家目录
- `cd ..` : 返回上一层目录

## which vs. whereis

- `which` : 只在用户的 `PATH` 所指定的文件中查找
- `whereis` : 在系统的所有目录中定位要查找的命令

## find

```
find /usr/share -name lostfile -print
```



## cd

- cd : 返回家目录
- cd ~ : 返回家目录
- cd .. : 返回上一层目录

## which vs. whereis

- which : 只在用户的 PATH 所指定的文件中查找
- whereis : 在系统的所有目录中定位要查找的命令

## find

```
find /usr/share -name lostfile -print
```



## cd

- cd : 返回家目录
- cd ~ : 返回家目录
- cd .. : 返回上一层目录

## which vs. whereis

- which : 只在用户的 PATH 所指定的文件中查找
- whereis : 在系统的所有目录中定位要查找的命令

## find

```
find /usr/share -name lostfile -print
```



- `ls`：列出用户有权访问的任何目录的内容
- `ls -i` (Inode)：显示文件的 inode 信息 (埋伏笔：inode)
- `ls -a` (All)：显示所有的文件和目录，包括隐藏的文件和目录
- 在文件名的前面加一个 `.` (英文句号) 可以隐藏该文件或目录
- `ls -l` (Long)：显示目录内容的相关扩展信息 (埋伏笔：文件类型、链接、权限)



- `ls` : 列出用户有权访问的任何目录的内容
- `ls -i` (Inode) : 显示文件的 inode 信息 (埋伏笔: inode)
- `ls -a` (All) : 显示所有的文件和目录, 包括隐藏的文件和目录
- 在文件名的前面加一个 `.` (英文句号) 可以隐藏该文件或目录
- `ls -l` (Long) : 显示目录内容的相关扩展信息 (埋伏笔: 文件类型、链接、权限)



- `ls` : 列出用户有权访问的任何目录的内容
- `ls -i` (Inode) : 显示文件的 inode 信息 (埋伏笔: inode)
- `ls -a` (All) : 显示所有的文件和目录, 包括隐藏的文件和目录
- 在文件名的前面加一个 `.` (英文句号) 可以隐藏该文件或目录
- `ls -l` (Long) : 显示目录内容的相关扩展信息 (埋伏笔: 文件类型、链接、权限)





- `ls` : 列出用户有权访问的任何目录的内容
- `ls -i` (Inode) : 显示文件的 inode 信息 (埋伏笔: inode)
- `ls -a` (All) : 显示所有的文件和目录, 包括隐藏的文件和目录
- 在文件名的前面加一个 `.` (英文句号) 可以隐藏该文件或目录
- `ls -l` (Long) : 显示目录内容的相关扩展信息 (埋伏笔: 文件类型、链接、权限)



- `ls` : 列出用户有权访问的任何目录的内容
- `ls -i` (Inode) : 显示文件的 inode 信息 (埋伏笔: inode)
- `ls -a` (All) : 显示所有的文件和目录, 包括隐藏的文件和目录
- 在文件名的前面加一个 `.` (英文句号) 可以隐藏该文件或目录
- `ls -l` (Long) : 显示目录内容的相关扩展信息 (埋伏笔: 文件类型、链接、权限)

Permissions (3 for owner, 3 for group, 3 for other)	Owner	Group	Date and time of last modification			
- rw-r--r--	1 mdw	users	2321	Mar 15	1994	Fontmap
- rw-r--r--	1 mdw	users	139836	Aug 11	09:11	Index.whole
d rwxr-xr-x	2 mdw	users	1024	Jan 25	1994	Xfonts
d rwxr-xr-x	3 mdw	users	1024	Sep 20	07:40	bin
- rw-r--r--	1 mdw	users	124408	Nov 2	10:53	bitgif.tar.gz
d rwxr-xr-x	2 mdw	users	2048	Jan 21	1994	bitmaps

Type of file  
("d" means  
"directory")

Number of  
hard links

Size in bytes  
(for a directory, bytes used  
to store directory information)

Name



## cat vs. more vs. less

友好性：cat < more < less

## head vs. tail

- 默认显示文件的前/后 10 行
- `-n x`：指定查看文件的前/后 `x` 行
- `tail -f` (Follow)：监视文件内容的变化



## cat vs. more vs. less

友好性：cat < more < less

## head vs. tail

- 默认显示文件的前/后 10 行
- `-n x`：指定查看文件的前/后 `x` 行
- `tail -f (Follow)`：监视文件内容的变化



- `rmdir`：只能删除空目录
- `rm`：不能删除目录
- `-f` (Force)：强行删除文件
- `-r` (Recursive)：进入到目录中递归删除文件
- `-fr`：删除目录及其子目录，**谨慎使用**
- **切勿尝试（为什么？）**：`rm -rf /`, `rm -rf *`



- `rmdir`：只能删除空目录
- `rm`：不能删除目录
- `-f` (Force)：强行删除文件
- `-r` (Recursive)：进入到目录中递归删除文件
- `-fr`：删除目录及其子目录，**谨慎使用**
- **切勿尝试（为什么？）**：`rm -rf /`, `rm -rf *`



- `rmdir`：只能删除空目录
- `rm`：不能删除目录
- `-f` (Force)：强行删除文件
- `-r` (Recursive)：进入到目录中递归删除文件
- `-fr`：删除目录及其子目录，**谨慎使用**
- **切勿尝试（为什么？）**：`rm -rf /`, `rm -rf *`



- `rmdir`：只能删除空目录
- `rm`：不能删除目录
- `-f` (Force)：强行删除文件
- `-r` (Recursive)：进入到目录中递归删除文件
- `-fr`：删除目录及其子目录，**谨慎使用**
- **切勿尝试（为什么？）**：`rm -rf /`, `rm -rf *`





- `rmdir`：只能删除空目录
- `rm`：不能删除目录
- `-f` (Force)：强行删除文件
- `-r` (Recursive)：进入到目录中递归删除文件
- `-fr`：删除目录及其子目录，**谨慎使用**
- **切勿尝试（为什么？）**：`rm -rf /`, `rm -rf *`



- `rmdir`：只能删除空目录
- `rm`：不能删除目录
- `-f` (Force)：强行删除文件
- `-r` (Recursive)：进入到目录中递归删除文件
- `-fr`：删除目录及其子目录，**谨慎使用**
- **切勿尝试（为什么？）**：`rm -rf /`, `rm -rf *`



```

SUSE-LES-11:/usr/local/resin/conf # df -h
文件系统      容量  已用  可用  已用%  挂载点
/dev/xvda2     97G   7.3G   89G    8% /
devtmpfs       520M   92K   520M    1% /dev
tmpfs          520M   140K   520M    1% /dev/shm

root@bt:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1       19G   14G   4.1G   78% /
none            291M   240K   291M    1% /dev
none            298M    12K   298M    1% /dev/shm
none            298M    84K   298M    1% /var/run
none            298M     0   298M    0% /var/lock
none            298M     0   298M    0% /lib/init/rw

```

## 参数

- `df -h`, `du -h` (Human-readable) : K, M, G
- `du -s` (Summarize) : 目录的总大小

1 指导思想

2 引言

3 文件系统基础

- 文件系统和分区
- 目录结构
- 路径

4 文件系统导航

- 目录操作
- 文件操作
- 文件系统管理

● 命令详解

5 文件类型

- 类型简介
- 链接

6 文件和目录权限

- 权限简介
- 修改权限

7 挂载文件系统

8 回顾与总结

- 总结
- 思考题



# 教学提纲

1 指导思想

2 引言

3 文件系统基础

- 文件系统和分区
- 目录结构
- 路径

4 文件系统导航

- 目录操作
- 文件操作
- 文件系统管理

● 命令详解

5 文件类型

- 类型简介
- 链接

6 文件和目录权限

- 权限简介
- 修改权限

7 挂载文件系统

8 回顾与总结

- 总结
- 思考题



文件类型	说明
-	普通文件（文本文件、二进制可执行文件、硬链接）
d	目录文件
l	符号链接文件
b	块设备文件（块输入/输出设备文件）
c	字符设备文件（原始输入/输出设备文件）
p	命令管道（一种进程间通信的机制）
s	套接字（用于进程间通信）



## 1 指导思想

## 2 引言

## 3 文件系统基础

- 文件系统和分区
- 目录结构
- 路径

## 4 文件系统导航

- 目录操作
- 文件操作
- 文件系统管理

## ● 命令详解

## 5 文件类型

- 类型简介
- 链接

## 6 文件和目录权限

- 权限简介
- 修改权限

## 7 挂载文件系统

## 8 回顾与总结

- 总结
- 思考题



## 类比: 身份证号 vs. 姓名 vs. 曾用名/笔名/外号

- 在 Linux 中, 每一个文件都有一个相关联的数字: inode
- Linux 使用 inode 而不是文件名来引用文件
- 在一个分区中, inode 是唯一的
- 不同分区内的文件可以有相同的 inode





## 类比: 身份证号 vs. 姓名 vs. 曾用名/笔名/外号

- 在 Linux 中, 每一个文件都有一个相关联的数字: inode
- Linux 使用 inode 而不是文件名来引用文件
- 在一个分区中, inode 是唯一的
- 不同分区内的文件可以有相同的 inode



## 类比: 身份证号 vs. 姓名 vs. 曾用名/笔名/外号

- 在 Linux 中, 每一个文件都有一个相关联的数字: inode
- Linux 使用 inode 而不是文件名来引用文件
- 在一个分区中, inode 是唯一的
- 不同分区内的文件可以有相同的 inode



## 类比: 身份证号 vs. 姓名 vs. 曾用名/笔名/外号

- 在 Linux 中, 每一个文件都有一个相关联的数字: inode
- Linux 使用 inode 而不是文件名来引用文件
- 在一个分区中, inode 是唯一的
- 不同分区内的文件可以有相同的 inode



## 硬链接 (Hard Link) (孙悟空的分身)

- 硬链接与原文件具有相同的 inode, 两者本质上没有区别
- 对硬链接的修改会反映到原文件上, 反之亦然
- 如果删除硬链接, 原文件照样正常使用, 反之亦然
- 不能跨越文件系统, “等同于” 不占空间的复制 + 同步更新
- 只能对文件建立硬链接, 而不能对目录建立硬链接

## 软链接 (Soft Link, 符号链接, Symbolic Link)

## 硬链接 (Hard Link) (孙悟空的分身)

- 硬链接与原文件具有相同的 inode, 两者本质上没有区别
- 对硬链接的修改会反映到原文件上, 反之亦然
- 如果删除硬链接, 原文件照样正常使用, 反之亦然
- 不能跨越文件系统, “等同于” 不占空间的复制 + 同步更新
- 只能对文件建立硬链接, 而不能对目录建立硬链接

## 软链接 (Soft Link, 符号链接, Symbolic Link)

## 硬链接 (Hard Link) (孙悟空的分身)

- 硬链接与原文件具有相同的 inode, 两者本质上没有区别
- 对硬链接的修改会反映到原文件上, 反之亦然
- 如果删除硬链接, 原文件照样正常使用, 反之亦然
- 不能跨越文件系统, “等同于” 不占空间的复制 + 同步更新
- 只能对文件建立硬链接, 而不能对目录建立硬链接

## 软链接 (Soft Link, 符号链接, Symbolic Link)

## 硬链接 (Hard Link) (孙悟空的分身)

- 硬链接与原文件具有相同的 inode，两者本质上没有区别
- 对硬链接的修改会反映到原文件上，反之亦然
- 如果删除硬链接，原文件照样正常使用，反之亦然
- 不能跨越文件系统，“等同于”不占空间的复制 + 同步更新
- 只能对文件建立硬链接，而不能对目录建立硬链接

## 软链接 (Soft Link, 符号链接, Symbolic Link)

## 硬链接 (Hard Link) (孙悟空的分身)

- 硬链接与原文件具有相同的 inode，两者本质上没有区别
- 对硬链接的修改会反映到原文件上，反之亦然
- 如果删除硬链接，原文件照样正常使用，反之亦然
- 不能跨越文件系统，“等同于”不占空间的复制 + 同步更新
- 只能对文件建立硬链接，而不能对目录建立硬链接

## 软链接 (Soft Link, 符号链接, Symbolic Link)

能够跨越文件系统，相当于 Windows 中的快捷方式



## 硬链接 (Hard Link) (孙悟空的分身)

- 硬链接与原文件具有相同的 inode，两者本质上没有区别
- 对硬链接的修改会反映到原文件上，反之亦然
- 如果删除硬链接，原文件照样正常使用，反之亦然
- 不能跨越文件系统，“等同于”不占空间的复制 + 同步更新
- 只能对文件建立硬链接，而不能对目录建立硬链接

## 软链接 (Soft Link, 符号链接, Symbolic Link)

- 能够跨越文件系统，相当于Windows 中的快捷方式
- 软链接具有唯一的 inode，内部保存的是原文件的路径地址
- 删除原文件或修改软链接，原文件也会随之改变

## 硬链接 (Hard Link) (孙悟空的分身)

- 硬链接与原文件具有相同的 inode，两者本质上没有区别
- 对硬链接的修改会反映到原文件上，反之亦然
- 如果删除硬链接，原文件照样正常使用，反之亦然
- 不能跨越文件系统，“等同于”不占空间的复制 + 同步更新
- 只能对文件建立硬链接，而不能对目录建立硬链接

## 软链接 (Soft Link, 符号链接, Symbolic Link)

- 能够跨越文件系统，相当于Windows 中的快捷方式
- 软链接具有唯一的 inode，内部保存的是原文件的路径地址
- 如果打开并修改软链接，原文件也会随之改变
- 如果删除软链接，原文件并不会受到影响
- 如果删除原文件，软链接将失效

## 硬链接 (Hard Link) (孙悟空的分身)

- 硬链接与原文件具有相同的 inode，两者本质上没有区别
- 对硬链接的修改会反映到原文件上，反之亦然
- 如果删除硬链接，原文件照样正常使用，反之亦然
- 不能跨越文件系统，“等同于”不占空间的复制 + 同步更新
- 只能对文件建立硬链接，而不能对目录建立硬链接

## 软链接 (Soft Link, 符号链接, Symbolic Link)

- 能够跨越文件系统，相当于Windows 中的快捷方式
- 软链接具有唯一的 inode，内部保存的是原文件的路径地址
- 如果打开并修改软链接，原文件也会随之改变
- 如果删除软链接，原文件并不会受到影响
- 如果删除原文件，软链接将失效

## 硬链接 (Hard Link) (孙悟空的分身)

- 硬链接与原文件具有相同的 inode，两者本质上没有区别
- 对硬链接的修改会反映到原文件上，反之亦然
- 如果删除硬链接，原文件照样正常使用，反之亦然
- 不能跨越文件系统，“等同于”不占空间的复制 + 同步更新
- 只能对文件建立硬链接，而不能对目录建立硬链接

## 软链接 (Soft Link, 符号链接, Symbolic Link)

- 能够跨越文件系统，相当于Windows 中的快捷方式
- 软链接具有唯一的 inode，内部保存的是原文件的路径地址
- 如果打开并修改软链接，原文件也会随之改变
- 如果删除软链接，原文件并不会受到影响
- 如果删除原文件，软链接将失效

## 硬链接 (Hard Link) (孙悟空的分身)

- 硬链接与原文件具有相同的 inode，两者本质上没有区别
- 对硬链接的修改会反映到原文件上，反之亦然
- 如果删除硬链接，原文件照样正常使用，反之亦然
- 不能跨越文件系统，“等同于”不占空间的复制 + 同步更新
- 只能对文件建立硬链接，而不能对目录建立硬链接

## 软链接 (Soft Link, 符号链接, Symbolic Link)

- 能够跨越文件系统，相当于Windows 中的快捷方式
- 软链接具有唯一的 inode，内部保存的是原文件的路径地址
- 如果打开并修改软链接，原文件也会随之改变
- 如果删除软链接，原文件并不会受到影响
- 如果删除原文件，软链接将失效

## 硬链接 (Hard Link) (孙悟空的分身)

- 硬链接与原文件具有相同的 inode，两者本质上没有区别
- 对硬链接的修改会反映到原文件上，反之亦然
- 如果删除硬链接，原文件照样正常使用，反之亦然
- 不能跨越文件系统，“等同于”不占空间的复制 + 同步更新
- 只能对文件建立硬链接，而不能对目录建立硬链接

## 软链接 (Soft Link, 符号链接, Symbolic Link)

- 能够跨越文件系统，相当于Windows 中的快捷方式
- 软链接具有唯一的 inode，内部保存的是原文件的路径地址
- 如果打开并修改软链接，原文件也会随之改变
- 如果删除软链接，原文件并不会受到影响
- 如果删除原文件，软链接将失效

## Make a file:

```
$ touch name_A
```

name\_A  
(ordinary file)



## Hard link:

```
$ ln name_A name_B
```

name\_A

name\_B  
(Hard link)



## Symbolic link:

```
$ ln -s name_A name_B
```

name\_A

name\_B  
(Symbolic link)



项目	硬链接	软链接
语法	In source hardlink	ln -s source softlink
本质	与原文件没区别	保存原文件的路径
inode	与原文件相同	与原文件不同，唯一
类比	不占空间的复制 + 同步更新	快捷方式
文件系统	不能跨越	能跨越
删除原文件	不受影响	失效
使用对象	文件	文件和目录
修改链接	原文件随之改变	
删除链接	原文件不受影响	





- 为命令、程序或文件取别名
- 创建不占存储空间的文件副本
- 为文件创建方便的快捷方式
- 对文件进行分组



# 教学提纲

- 1 指导思想
- 2 引言
- 3 文件系统基础
  - 文件系统和分区
  - 目录结构
  - 路径
- 4 文件系统导航
  - 目录操作
  - 文件操作
  - 文件系统管理
- 5 文件类型
  - 命令详解
  - 类型简介
  - 链接
- 6 文件和目录权限
  - 权限简介
  - 修改权限
- 7 挂载文件系统
- 8 回顾与总结
  - 总结
  - 思考题



# 教学提纲

## 1 指导思想

## 2 引言

## 3 文件系统基础

- 文件系统和分区
- 目录结构
- 路径

## 4 文件系统导航

- 目录操作
- 文件操作
- 文件系统管理

## ● 命令详解

## 5 文件类型

- 类型简介
- 链接

## 6 文件和目录权限

- 权限简介
- 修改权限

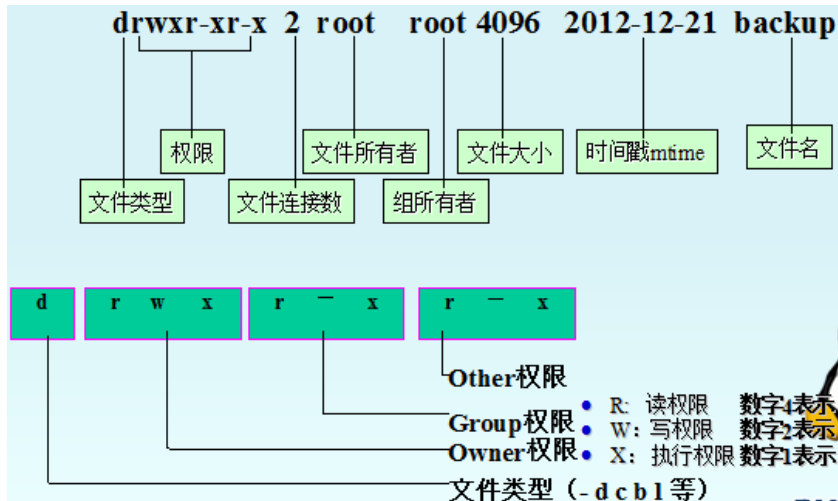
## 7 挂载文件系统

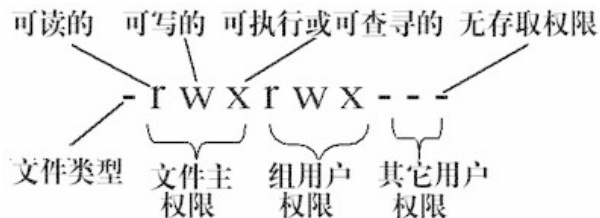
## 8 回顾与总结

- 总结
- 思考题



# 文件系统 | 权限 | 简介





字符位置	含义	助记
2~4	文件所有者	user
5~7	文件所属组	group
8~10	其他任何人	others



字符	助记	权限	对文件	对目录
r	Read	读	查看文件内容【cat】	读取/列出目录或子目录内容【ls】
w	Write	写	修改文件内容（添加文本或删除文件）【vim】	在目录中创建、修改、删除文件或子目录【touch】
x	eXecute	执行	执行/运行文件【sh】	进入目录搜索【cd】
-	-	-	无	无

注意：目录必须具有 x 权限，否则无法进入并查看其内容！



# 教学提纲

## 1 指导思想

## 2 引言

## 3 文件系统基础

- 文件系统和分区
- 目录结构
- 路径

## 4 文件系统导航

- 目录操作
- 文件操作
- 文件系统管理

## ● 命令详解

## 5 文件类型

- 类型简介
- 链接

## 6 文件和目录权限

- 权限简介
- 修改权限

## 7 挂载文件系统

## 8 回顾与总结

- 总结
- 思考题



## 修改权限

chmod (CHange MODe)

### 两种方式（板书）

#### ① 符号模式：容易理解

- 用户：u, g, o, a
- 操作：+, -, =
- 权限：r, w, x, -

#### ② 绝对模式：更加高效

- 0, 1, 2, 4
- $3 = 1 + 2$
- $5 = 1 + 4$
- $6 = 2 + 4$
- $7 = 1 + 2 + 4$



## 修改权限

chmod (CHange MODe)

## 两种方式 (板书)

### 1 符号模式：容易理解

- 用户：u, g, o, a
- 操作：+, -, =
- 权限：r, w, x, -

### 2 绝对模式：更加高效

- 0, 1, 2, 4
- $3 = 1 + 2$
- $5 = 1 + 4$
- $6 = 2 + 4$
- $7 = 1 + 2 + 4$

## 用户

- u : User, 用户
- g : Group, 组
- o : Other, 其他人
- a : All, 所有人

## 操作

- + : 添加
- - : 删除
- = : 指定

## 权限

- r : Read, 读
- w : Write, 写
- x : eXecute, 执行
- - : 无

## 实例 (一一解释具体含义)

- `chmod u-x testfile`
- `chmod g=r-x testfile`
- `chmod o+wx testfile`
- `chmod uo+x,g-w testfile`
- `chmod u-x,g=r-x,o+wx testfile`

## 用户

- u : User, 用户
- g : Group, 组
- o : Other, 其他人
- a : All, 所有人

## 操作

- + : 添加
- - : 删除
- = : 指定

## 权限

- r : Read, 读
- w : Write, 写
- x : eXecute, 执行
- - : 无

## 实例 (一一解释具体含义)

- `chmod u-x testfile`
- `chmod g=r-x testfile`
- `chmod o+wx testfile`
- `chmod uo+x,g-w testfile`
- `chmod u-x,g=r-x,o+wx testfile`

## 用户

- u : User, 用户
- g : Group, 组
- o : Other, 其他人
- a : All, 所有人

## 操作

- + : 添加
- - : 删除
- = : 指定

## 权限

- r : Read, 读
- w : Write, 写
- x : eXecute, 执行
- - : 无

## 实例 (一一解释具体含义)

- `chmod u-x testfile`
- `chmod g=r-x testfile`
- `chmod o+wx testfile`
- `chmod uo+x,g-w testfile`
- `chmod u-x,g=r-x,o+wx testfile`

## 用户

- u : User, 用户
- g : Group, 组
- o : Other, 其他人
- a : All, 所有人

## 操作

- + : 添加
- - : 删除
- = : 指定

## 权限

- r : Read, 读
- w : Write, 写
- x : eXecute, 执行
- - : 无

## 实例 (一一解释具体含义)

- `chmod u-x testfile`
- `chmod g=r-x testfile`
- `chmod o+wx testfile`
- `chmod uo+x,g-w testfile`
- `chmod u-x,g=r-x,o+wx testfile`

数字	符号	权限
0	---	无权限
1	--x	可执行
2	-w-	可写
3	-wx	可写、可执行 (2+1)
4	r--	可读
5	r-x	可读、可执行 (4+1)
6	rw-	可读、可写 (4+2)
7	rwx	可读、可写、可执行 (4+2+1)

## 实例 (一一解释具体含义)

- `chmod 740 testfile`
- `chmod 755 testfile`

数字	符号	权限
0	---	无权限
1	--x	可执行
2	-w-	可写
3	-wx	可写、可执行 (2+1)
4	r--	可读
5	r-x	可读、可执行 (4+1)
6	rw-	可读、可写 (4+2)
7	rwX	可读、可写、可执行 (4+2+1)

### 实例 (一一解释具体含义)

- `chmod 740 testfile`
- `chmod 755 testfile`

- 1 指导思想
- 2 引言
- 3 文件系统基础
  - 文件系统和分区
  - 目录结构
  - 路径
- 4 文件系统导航
  - 目录操作
  - 文件操作
  - 文件系统管理
- 5 命令详解
- 6 文件类型
  - 类型简介
  - 链接
- 7 文件和目录权限
  - 权限简介
  - 修改权限
- 8 挂载文件系统
  - 挂载文件系统
- 9 回顾与总结
  - 总结
  - 思考题





# 文件系统 | 挂载 | mount

```
[root@AY130424102830Z ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/xvda1       20G   5.8G   13G   32% /
tmpfs            498M    0   498M    0% /dev/shm
/dev/xvdb1       9.9G  151M   9.2G    2% /www
[root@AY130424102830Z ~]# mount
/dev/xvda1 on / type ext3 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
tmpfs on /dev/shm type tmpfs (rw)
xenfs on /proc/xen type xenfs (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
/dev/xvdb1 on /www type ext3 (rw)
```

## 语法（回顾文件系统的类型）

- `mount -t FILE.SYSTEM.TYPE DEVICE DIRECTORY`
- `mount -t iso9660 /dev/cdrom /mnt/cdrom`
- `umount DEVICE.TO.UNMOUNT`
- `umount /dev/cdrom`

```
[root@AY130424102830Z ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/xvda1       20G   5.8G   13G   32% /
tmpfs            498M    0   498M    0% /dev/shm
/dev/xvdb1       9.9G  151M   9.2G    2% /www
[root@AY130424102830Z ~]# mount
/dev/xvda1 on / type ext3 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
tmpfs on /dev/shm type tmpfs (rw)
xenfs on /proc/xen type xenfs (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
/dev/xvdb1 on /www type ext3 (rw)
```

## 语法（回顾文件系统的类型）

- `mount -t FILE.SYSTEM.TYPE DEVICE DIRECTORY`
- `mount -t iso9660 /dev/cdrom /mnt/cdrom`
- `umount DEVICE.TO.UNMOUNT`
- `umount /dev/cdrom`

## 1 指导思想

## 2 引言

## 3 文件系统基础

- 文件系统和分区
- 目录结构
- 路径

## 4 文件系统导航

- 目录操作
- 文件操作
- 文件系统管理

## ● 命令详解

## 5 文件类型

- 类型简介
- 链接

## 6 文件和目录权限

- 权限简介
- 修改权限

## 7 挂载文件系统

## 8 回顾与总结

- 总结
- 思考题



- 1 指导思想
- 2 引言
- 3 文件系统基础
  - 文件系统和分区
  - 目录结构
  - 路径
- 4 文件系统导航
  - 目录操作
  - 文件操作
  - 文件系统管理
- 5 文件类型
  - 类型简介
  - 链接
- 6 文件和目录权限
  - 权限简介
  - 修改权限
- 7 挂载文件系统
- 8 回顾与总结
  - 总结
  - 思考题



## 知识点

- Linux 的文件系统：目录结构，主要的基本目录
- Linux 中的路径：绝对路径和相对路径
- 文件系统导航的常见命令
- Linux 中的文件类型：常见类型，硬链接和软链接
- Linux 中的权限：文件和目录的权限，符号模式和绝对模式
- 文件系统的挂载与卸载

## 技能

- 在命令行中进行文件系统的导航
- 在命令行中创建硬链接和软链接
- 在命令行中修改文件的权限
- 在命令行中挂载、卸载文件系统

1 指导思想

2 引言

3 文件系统基础

- 文件系统和分区
- 目录结构
- 路径

4 文件系统导航

- 目录操作
- 文件操作
- 文件系统管理

● 命令详解

5 文件类型

- 类型简介
- 链接

6 文件和目录权限

- 权限简介
- 修改权限

7 挂载文件系统

8 回顾与总结

- 总结
- 思考题



- 1 列举 Linux 中的基本目录并解释其功能。
- 2 举例说明绝对路径和相对路径的区别。
- 3 列举几个进行文件系统导航的命令。
- 4 解释 ls -l 输出结果中每一列的含义。
- 5 比较 Linux 中的硬链接和软链接。
- 6 Linux 中的权限包括几种，针对哪些用户？
- 7 文件和目录的 rwx 权限有何异同？
- 8 举例说明如何使用符号模式修改权限？
- 9 举例说明如何使用绝对模式修改权限？



# 下节预告

总结日常使用 Windows 过程中的基本操作：目录操作、文件操作、系统管理、压缩解压、关机重启、……







TEX

LATEX

X<sub>Y</sub>TEX

Beamer

