

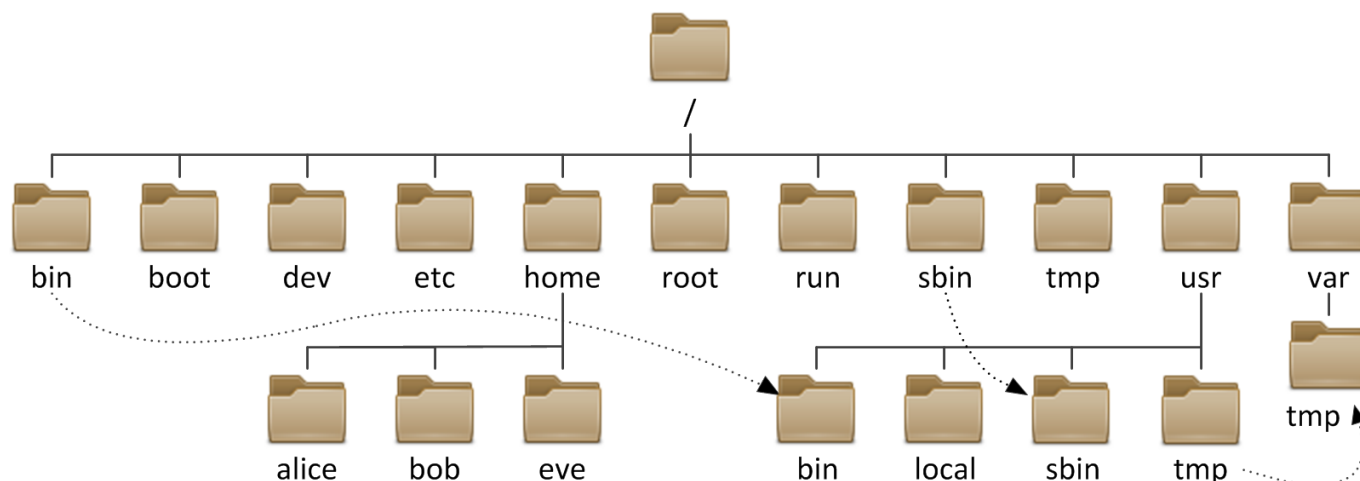
@[TOC](

linux文件管理

)

一.文件系统结构及组织细节

1.linux文件目录结构(centos7)



2.linux文件系统简述

- linux系统下文件和目录被组织成一个单根倒置的树状结构 文件系统从根目录下开始，用“/”表示
- 根文件系统(rootfs): root filesystem
- linux文件系统中的文件名称区分大小写，其中以点(.)开头的文件为隐藏文件
- 文件有两类数据： 元数据： metadata 数据： data
- 文件系统分层结构： LSB Linux Standard Base
FHS: (Filesystem Hierarchy Standard) 该标准由文件系统层级标准组定制,可直接在下面网站获得其pdf版本
- FHS:该标准提供了一些在类unix系统下如何安排文件和目录的一些指导和要求；目的在于支持应用程序、系统管理工具、开发工具和脚本的互操作性及为这些系统所提供的文档的一致性。

[FHS-pdf](#)

3.linux下的文件命令规则

- 文件名最长255个字节
- 包括路径在内文件名称最长4095个字节
- 蓝色-->目录 绿色-->可执行文件 红色-->压缩文件 浅蓝色-->链接文件 灰色-->其他文件 (可以自定义)
- 除了斜杠和NUL,所有字符都有效.但使用特殊字符的目录名和文件不推荐使用，有些字符需要用引号来引用它们

- 标准Linux文件系统（如ext4），文件名称大小写敏感
例如：MAIL, Mail, mail, mAiL

4.文件系统结构细节

目录	功能
/boot	引导文件存放目录，内核文件(vmlinuz)、引导加载器(bootloader, grub)都存放于此目录
/bin	所有用户使用的基本命令；不能关联至独立分区，OS启动即会用到的程序
/sbin	管理类的基本命令；不能关联至独立分区，OS启动即会用到的程序
/lib	启动时程序依赖的基本共享库文件以及内核模块文件(/lib/modules)
/lib64	专用于x86_64系统上的辅助共享库文件存放位置
/etc	配置文件目录
/home/USERNAME	普通用户家目录
/root	管理员的家目录
/media	便携式移动设备挂载点
/mnt	临时文件系统挂载点
/dev	设备文件及特殊文件存储位置
	b: block device，块设备，随机访问
	c: character device，字符设备，线性访问
/opt	第三方应用程序的安装位置
/srv	系统上运行的服务用到的数据
/tmp	临时文件存储位置
/usr	universal shared, read-only data 全局共享的只读文件存放地
	/usr/bin: 保证系统拥有完整功能而提供的应用程序
	/usr/sbin: centos7上访问/sbin实质是访问 /usr/sbin
	/usr/lib: 32位使用
	/usr/lib64: 只存在64位系统
	/usr/include: C程序的头文件(header files)
	/usr/share: 结构化独立的数据，例如doc, man等
	/usr/local: 第三方应用程序的安装位置: bin, sbin, lib, lib64, etc, share
/var	variable data files 可变文件存放地点
	/var/cache: 应用程序缓存数据目录

目录	功能
	/var/lib: 应用程序状态信息数据
	/var/local: 专用于为/usr/local下的应用程序存储可变数据
	/var/lock: 锁文件
	/var/log: 日志目录及文件
	/var/opt: 专用于为/opt下的应用程序存储可变数据
	/var/run: 运行中的进程相关数据,通常用于存储进程pid文件
	/var/spool: 应用程序数据池
	/var/tmp: 保存系统两次重启之间产生的临时数据
/proc	用于输出内核与进程信息相关的虚拟文件系统
/sys	用于输出当前系统上硬件设备相关信息虚拟文件系统
/selinux	security enhanced Linux，selinux相关的安全策略等信息的存储位置

5.Linux下文件种类及各类文件存放地

- 文件类型

符号	文件类型
-	普通文件
d	目录文件
b	块设备
c	字符设备
l	符号链接文件
p	管道文件pipe
s	套接字文件socket

- 下面例子中第一列第一个字符表示文件类型，如"-rw-----"中开头的"-"表示普通字符

```
[root@centos7 ~]$ll
total 112
-rw-----. 1 root root 2026 Sep 20 14:31 anaconda-ks.cfg
-rw-r--r--. 1 root root 77 Sep 22 15:42 bash-scripting.sh
drwxr-xr-x. 2 root root 6 Sep 20 15:08 Desktop
drwxr-xr-x. 2 root root 6 Sep 20 15:05 Documents
drwxr-xr-x. 2 root root 6 Sep 20 15:05 Downloads
-rw-r--r--. 1 root root 2056 Sep 20 14:45 initial-setup-ks.cfg
-rw-r--r--. 1 root root 0 Sep 23 11:25 lslfdjsljD
drwxr-xr-x. 2 root root 6 Sep 20 15:05 Music
```

```
drwxr-xr-x. 2 root root      6 Sep 20 15:05 Pictures
drwxr-xr-x. 2 root root      6 Sep 20 15:05 Public
drwxr-xr-x. 3 root root     21 Sep 21 18:39 scripts
drwxr-xr-x. 2 root root      6 Sep 20 15:05 Templates
-rw-r--r--. 1 root root 15502 Sep 22 20:18 user-home.png
-rw-r--r--. 1 root root 83424 Sep 22 20:30 user-trash.png
drwxr-xr-x. 2 root root      6 Sep 20 15:05 Videos
```

- 各类型的文件存放地

文件类型	存放的文件夹
二进制程序	/bin, /sbin, /usr/bin, /usr/sbin, /usr/local/bin, /usr/local/sbin
库文件	/lib, /lib64, /usr/lib, /usr/lib64, /usr/local/lib, /usr/local/lib64
配置文件	/etc, /etc/DIRECTORY, /usr/local/etc
帮助文件	/usr/share/man, /usr/share/doc, /usr/local/share/man, /usr/local/share/doc

二.linux系统下文件的创建和目录的导航等操作

1.当前工作目录相关的操作

- 每个shell和系统进程都有一个当前的工作目录，用户登录后默认在自己的家目录，如:/home/alice
- 术语CWD:current work directory 当前工作路径
- 显示当前shell CWD的绝对路径使用： `pwd: printing working directory`
 - P 显示真实物理路径
 - L 显示链接路径（默认）

2.绝对和相对路径表示和相关操作

- 绝对路径表示从根开始完整文件的位置路径，以正斜杠开始，可用于任何想指定一个文件名的时候
 - 相对路径名指定相对于当前工作目录或某目录的位置，不以斜线开始，可以作为一个简短的形式指定一个文件名
- 查看路径的基名： `basename /path/to/somefile` 查看某路径的目录名： `dirname /path/to/somefile` 如：

```
[root@centos7 ~]$basename /etc/sysconfig/network-scripts
network-scripts
[root@centos7 ~]$dirname /etc/sysconfig/network-scripts
/etc/sysconfig
```

3.更改目录

- `cd` 命令用于改变工作目录
 - `cd /home/wang/`
 - `cd home/wang`

切换至父目录: `cd ..`

切换至当前用户主目录: `cd`

切换至以前的工作目录: `cd -`

选项: `-P` #使用真实的物理文件路径, 不跟随符号链接文件 相关的环境变量:

`PWD`: 当前目录路径

`OLDPWD`: 上一次目录路径

4.使用ls命令列出目录内容

列出当前目录的内容或指定目录

用法: `ls [options] [files_or_dirs]`

示例

```
ls -a 包含隐藏文件
ls -l 显示额外的信息
ls -R 目录递归
ls -ld 目录和符号链接信息
ls -1 文件分行显示
ls -S 按从大到小排序
ls -t 按mtime排序
ls -u 配合-t选项, 显示并按atime从新到旧排序
ls -U 按目录存放顺序显示
ls -X 按文件后缀排序
```

5.使用stat命令查看文件

- 默认stat列出所给的文件的大小、类型和三个时间戳

三个时间戳:

access time 访问时间, atime, 读取文件内容

modify time 修改时间, mtime, 改变文件内容(数据)

change time 改变时间, ctime, 元数据发生改变

```
[root@centos7 ~]$stat .bashrc
File: '.bashrc'
Size: 759          Blocks: 8          IO Block: 4096   regular file
Device: 802h/2050d Inode: 201510615   Links: 1
Access: (0644/-rw-r--r--)  Uid: (  0/   root)   Gid: (  0/   root)
Context: system_u:object_r:admin_home_t:s0
Access: 2019-09-23 21:58:28.695883200 +0800
Modify: 2019-09-23 17:42:43.856543586 +0800
Change: 2019-09-23 17:42:43.857543588 +0800
Birth: -
[root@centos7 ~]$stat -c %x .bashrc
2019-09-23 21:58:28.695883200 +0800
[root@centos7 ~]$stat -c %y .bashrc
2019-09-23 17:42:43.856543586 +0800
[root@centos7 ~]$stat -c %z .bashrc
2019-09-23 17:42:43.857543588 +0800
```

6.通配符模式 wild-card pattern

- 使用ls等命令时可以使用文件名通配开筛选需要的文件，使用特定的字符代表需要匹配的字符：

字符	代表的意思
*	匹配零个或多个字符
?	匹配任何单个字符
~	当前用户家目录
~mage	用户mage家目录
~+	当前工作目录
~-	前一个工作目录
[0-9]	匹配数字范围
[a-z]	字母
[A-Z]	字母
[wang]	匹配列表中的任何一个字符
[^wang]	匹配列表中的所有字符以外的字符

- 另外，linux系统中有定义好的字符类表示入下：

字符	代表的意思
[:digit:]	任意数字，相当于0-9
[:lower:]	任意小写字母
[:upper:]	任意大写字母
[:alpha:]	任意大小写字母
[:alnum:]	任意数字或字母
[:blank:]	水平空白字符
[:space:]	水平或垂直空白字符
[:punct:]	标点符号
[:print:]	可打印字符
[:cntrl:]	控制（非打印）字符
[:graph:]	图形字符
[:xdigit:]	十六进制字符

7.通配符练习

- 1、显示/var目录下所有以l开头，以一个小写字母结尾，且中间出现至少一位数字的文件或目录
`ls -d /var/l*[0-9]*[[:lower:]]`
- 2、显示/etc目录下以任意一位数字开头，且以非数字结尾的文件或目录
`ls -d /etc/[[:digit:]]*[^[:digit:]]`
- 3、显示/etc/目录下以非字母开头，后面跟了一个字母及其它任意长度任意字符的文件或目录
`ls -d /etc/[^[[:alpha:]]][[:alpha:]]*`
- 4、显示/etc/目录下所有以rc开头，并后面是0-6之间的数字，其它为任意字符的文件或目录
`ls -d /etc/rc[0-6]*`
- 5、显示/etc目录下，所有以.d结尾的文件或目录
`ls /etc/*.d -d`
- 6、显示/etc目录下，所有.conf结尾，且以m,n,r,p开头的文件或目录
`ls /etc/[mnpr]*.conf -d`
- 7、只显示/root下的隐藏文件和目录
`ls -d /root/.* (ls -d .[^.]*)`
- 8、只显示/etc下的非隐藏目录
`ls /etc/*/ -d`

8.使用touch命令创建文件和更改文件的时间戳

- touch命令
 格式: touch [OPTION]... FILE...
 -a 仅改变 atime和ctime
 -m 仅改变 mtime和ctime
 -t [[CC]YY]MMDDhhmm[ss]
 指定atime和mtime的时间戳时加-c选项表示如果文件不存在则不新建文件

三.复制、转移和删除文件及目录

1.使用cp命令复制文件、目录和改名

- cp命令使用简述

```
cp [OPTION]... [-T] SOURCE DEST
cp [OPTION]... SOURCE... DIRECTORY
cp [OPTION]... -t DIRECTORY SOURCE...
cp SRC DEST
```

SRC是文件:

如果目标不存在: 新建DEST, 并将SRC中内容填充至DEST中

如果目标存在:

如果DEST是文件: 将SRC中的内容覆盖至DEST中

基于安全, 建议为cp命令使用-i选项

如果DEST是目录: 在DEST下新建与原文件同名的文件, 并将SRC中内容填充至新文件中

```
cp SRC... DEST
```

SRC...

多个文件

DEST 必须存在，且为目录，其它情形均会出错

cp SRC DEST

SRC是目录：此时使用选项：-r

如果DEST不存在：则创建指定目录，复制SRC目录中所有文件至DEST中

如果DEST存在：

 如果DEST是文件：报错

 如果DEST是目录：

- cp命令常用选项

选项	功能
-i	覆盖前提示
-n	不覆盖，注意两者顺序
-r, -R	递归复制目录及内部的所有内容
-a	归档，相当于-dR --preserve=all
-d --no-dereference -- preserve=links	不复制原文件，只复制链接名
--preserve=[ATTR_LIST]	mode: 权限 ownership: 属主属组 timestamp: links ,xattr, context, all
-p	等同--preserve=mode,ownership,timestamp
-v --verbose	显示详细信息
-f --force	删除目标文件夹的同名文件，再创建新文件
-u --update	只复制源比目标更新文件或目标不存在的文件
-b	目标存在，覆盖前先备份，形式为 filename~
--backup=numbered	目标存在，覆盖前先备份加数字后缀:cp -av --backup=numbered /etc/issue ./

- cp命令各种情况

源\目标	不存在	存在且为文件	存在且为目录
一个文件	新建DEST，并将SRC中内容填充至DEST中	将SRC中的内容覆盖至DEST中，注意：数据丢失风险！建议用 -i 选项	在DEST下新建与原文件同名的文件，并将SRC中内容填充至新文件中
多个文件	提示错误	提示错误	在DEST下新建与原文件同名的文件，并将原文件内容复制进新文件中

源\目标	不存在	存在且为文件	存在且为目录
目录	创建指定DEST同名目录，		在DEST下新建与原目录同名的
须使	复制SRC目录中所有文件至	提示错误	目录，并将SRC中内容复制至新
用-r选	DEST下		目录中
项			

2.使用mv命令移动和重命名文件

```
mv [OPTION]... [-T] SOURCE DEST
mv [OPTION]... SOURCE... DIRECTORY
mv [OPTION]... -t DIRECTORY SOURCE...
常用选项:
  -i 交互式
  -f 强制
  -b 目标存在，覆盖前先备份
```

3.使用rm命令删除文件

```
rm [OPTION]... FILE...
常用选项:
  -i 交互式
  -f 强制删除
  -r 递归
  --no-preserve-root 删除/
示例:
  rm -rf /*
```

4.目录操作

```
tree 显示目录树
  -d: 只显示目录
  -L level: 指定显示的层级数目
  -P pattern: 只显示由指定wild-card pattern匹配到的路径
mkdir 创建目录
  -p: 存在于不报错，且可自动创建所需的各目录
  -v: 显示详细信息
  -m MODE: 创建目录时直接指定权限
rmdir 删除空目录
  -p: 递归删除父空目录x
  -v: 显示详细信息
rm -r 递归删除目录树
```

5.练习

- (1)每天将/etc/目录下所有文件，备份到/data独立的子目录下，并要求子目录格式为 backupYYYY-mm-dd，备份过程可见

```
cp -a -v /etc /data/backup`date +%Y`
```
- (2)创建/data/rootdir目录，并复制/root下所有文件到该目录内，要求保留原有限权 mkdir /data/rootdir

```
cp -av /root /data/rootdir
```
- (3) 如何创建/testdir/dir1/x, /testdir/dir1/y, /testdir/dir1/x/a, /testdir/dir1/x/b, /testdir/dir1/y/a, /testdir/dir1/y/b

```
mkdir -pv /testdir/dir1/{x,y}/{a,b}
```
- (4) 如何创建/testdir/dir2/x, /testdir/dir2/y, /testdir/dir2/x/a,/testdir/dir2/x/b

```
mkdir -pv /testdir/dir2/{x/{a,b},y}
```
- (5) 如何创建/testdir/dir3, /testdir/dir4, /testdir/dir5, /testdir/dir5/dir6, /testdir/dir5/dir7

```
mkdir -pv /testdir/dir{3,4,5/dir{6,7}}
```

四.linux下的inode解释

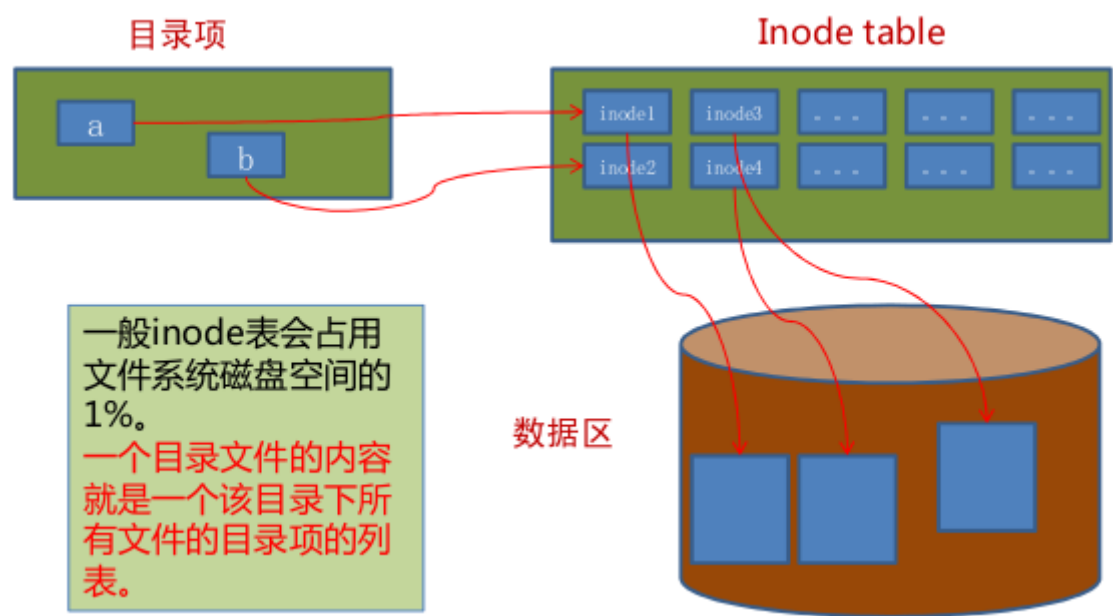
1.inode简述

- 几乎每个文件系统都会需要大量不同的数据结构来保证其底层对各种文件存储目的的支持;在linux系统中（ext3或者ext4或者xfs文件系统）就有一个很重要的数据结构叫**inode**(index node),一个inode包含某个文件或者某个目录的以下信息（也叫元数据）：

信息项	英文术语
文件类型（可执行文件，块文件，字符文件等）	File types (executable, block special etc)
文件权限（读，写，执行等）	Permissions (read, write etc)
文件属主全局唯一标识符	UID (Owner)
文件属组全局唯一标识符	GID (Group)
文件大小	FileSize
时间戳：最后访问时间、最后修改时间、最后改变时间和inode号变化信息	Time stamps including last access, last modification and last inode number change.
文件链接数（硬链接和软链接）	Number of links (soft/hard)
文件在磁盘的位置	Location of ile on harddisk.
其他信息	Some other metadata about file.

- inode**数据结构被存储在**inode**表中：由于每个**inode**代表某个文件的所有属性信息，所以**inode**表就记录了整个文件系统上的所有文件的信息（元数据）

- linux文件系统中每个目录下的文件被存储成目录项，每一项对应其inode号，通过inode号就可以访问到inode表的某一项，该项就记录了该文件的元数据。如下图：

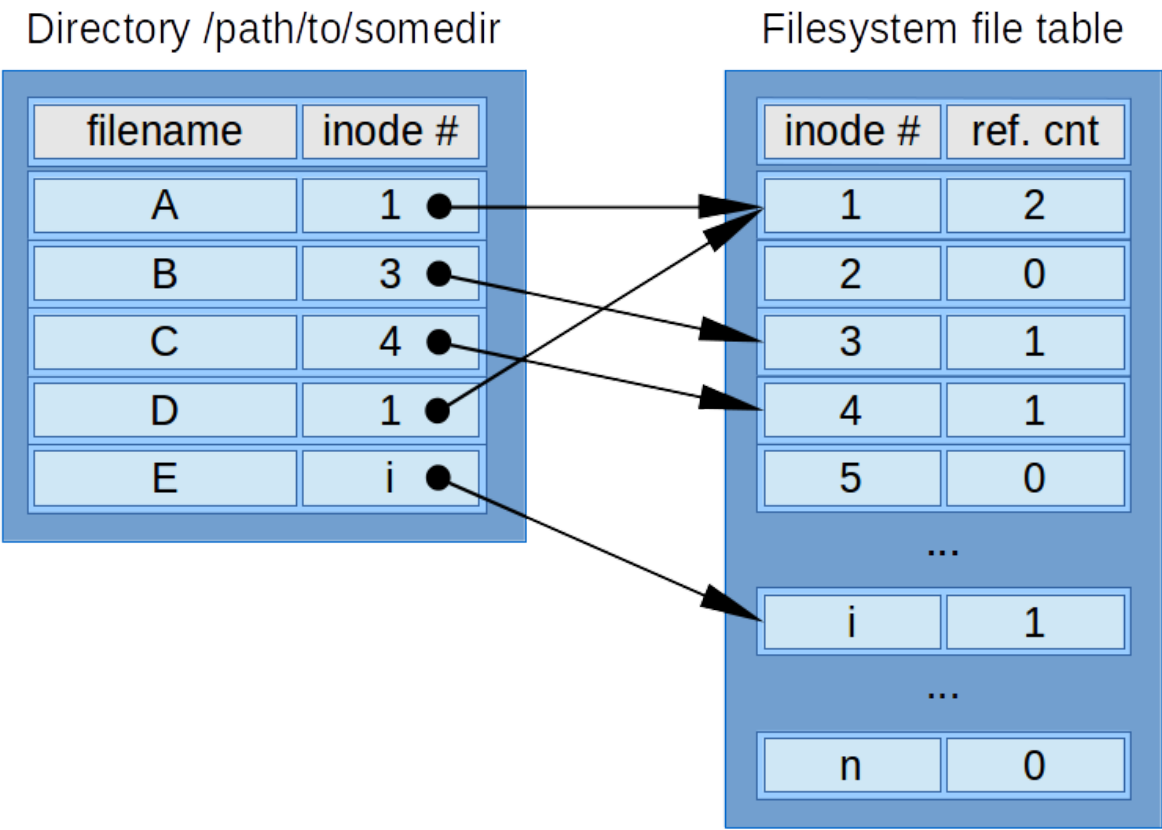


目录项和inode表及inode表某项所指向的数据交互图

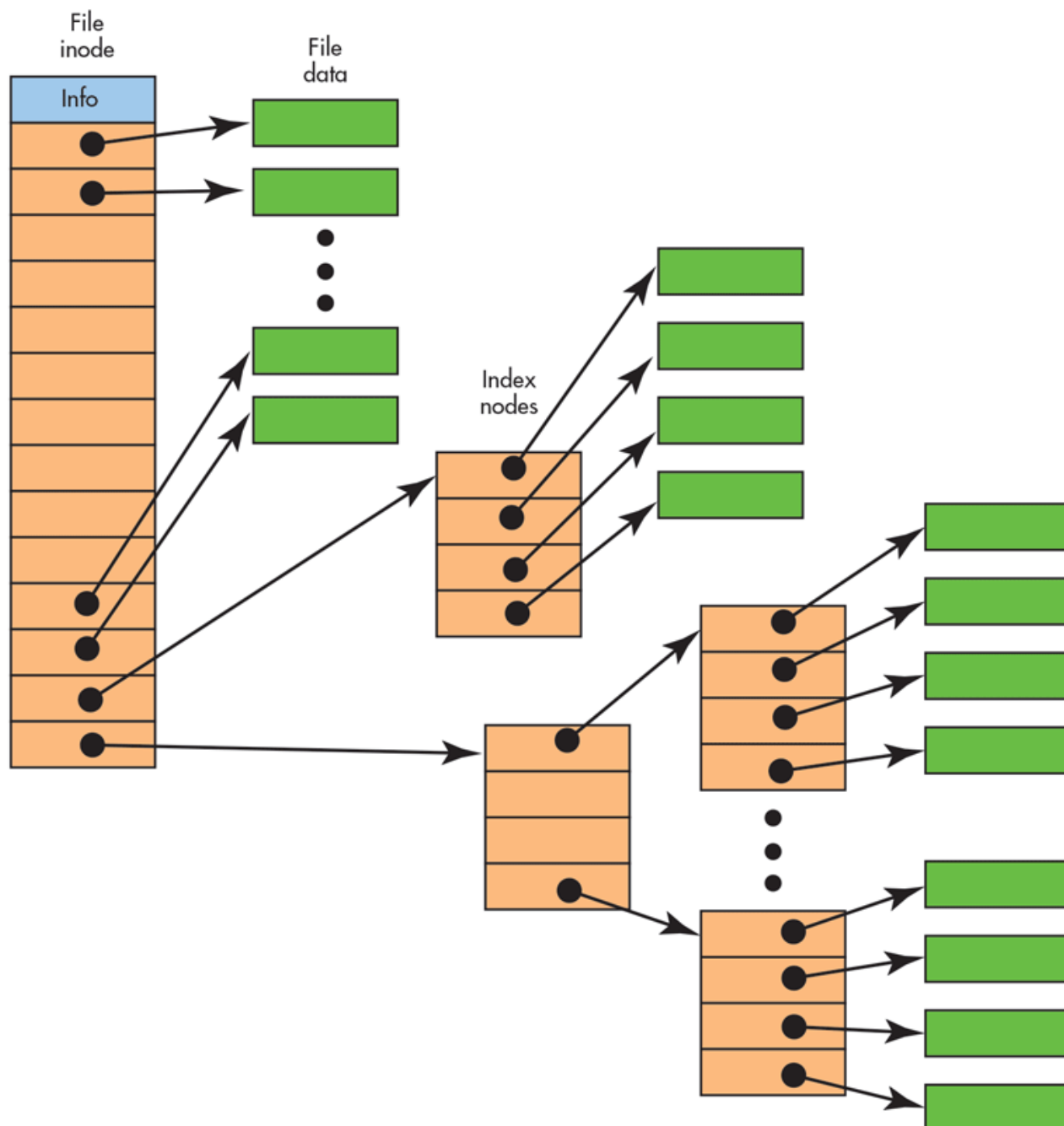
- 上图中目录项通过inode号找到其在inode table中的元数据，linux系统下查看inode号使用ls -li:

```
[root@centos8 /data]$ls -li my_file_1.txt
144 my_file_1.txt                                # 144就是文件my_file_1.txt的inode号
```

- 如下图，左边的表格是文件名和inode表格，右边是inode表项和inode号码的对应表格，通过ref.cnt就可以找到inode项



- 找到inode项后，该项存储有下图的元数据和指向文件实际数据的指针（箭头的起点所在的数据块就存储了指针：其中有12个一级指针，当指针不够用时，会使用间接指针和三次间接及更多次的间接指针）



2. cp、rm、mv等命令都和inode紧密相关

3.cp命令与inode

分配一个空闲的inode号，在inode表中生成新条目；在目录中创建一个目录项，将名称与inode编号关联；拷贝数据生成新的文件

4.rm命令和inode

链接数递减，从而释放的inode号可以被重用 把数据块放在空闲列表中 删除目录项 数据实际上不会马上被删除，但当另一个文件使用数据块时将被覆盖

5.mv命令和inode

如果mv命令的目标和源在相同的文件系统，作为mv 命令

用新的文件名创建对应新的目录项
删除旧目录条目对应的旧的文件名
不影响inode表（除时间戳）或磁盘上的数据位置：没有数据被移动！

如果目标和源在一个不同的文件系统，`mv`相当于`cp`和`rm`

五.硬链接和软链接

1.硬链接

创建硬链接会增加额外的记录项以引用文件
对应于同一文件系统上一个物理文件
每个目录引用相同的inode号
创建时链接数递增

删除文件时：

`rm`命令递减计数的链接
文件要存在，至少有一个链接数
当链接数为零时，该文件被删除
不能跨越驱动器或分区

为文件创建硬链接语法：

```
ln filename [linkname]
```

2.符号（或软）链接

一个符号链接指向另一个文件
`ls -l`的 显示链接的名称和引用的文件
一个符号链接的内容是它引用文件的名称
可以对目录进行
可以跨分区
指向的是另一个文件的路径；其大小为指向的路径字符串的长度；不增加或减少目标文件inode的引用计数
语法：

```
ln -s filename [linkname]
```

3.确定文件内容

文件可以包含多种类型的数据
检查文件的类型，然后确定适当的打开命令或应用程序使用

```
file [options] <filename>...
```

常用选项：

`-b` 列出文件辨识结果时，不显示文件名称
`-f filelist` 列出文件filelist中文件名的文件类型

- F 使用指定分隔符号替换输出文件名后默认的":"分隔符
- L 查看对应软链接对应文件的文件类型
- help 显示命令在线帮助