

# 文件时间

任何一个操作系统都有时间的概念，时间的概念主要用于对文件和系统中发生的时间进行记录，在 Linux 中，可以使用 stat 查看 Linux 系统中文件的时间。

## stat

用于显示文件时间和 inode 内容，inode 相关的知识会在后面的磁盘管理章节详细讲解，这边主要来看文件的时间

```
stat [选项]... 文件...
```

## 案例演示

stat 查看文件时间，这边为了我们方便看得懂，建议改为英文系统环境

```
[root@localhost ~]# export LANG="en_US.UTF-8"
# 改回中文是LANG="zh_CN.UTF-8"
[root@localhost ~]# stat anaconda-ks.cfg
  File: 'anaconda-ks.cfg'
  Size: 1241          Blocks: 8          IO Block: 4096   regular file
Device: fd00h/64768d    Inode: 33574979    Links: 1
Access: (0600/-rw-----)  Uid: (  0/   root)   Gid: (  0/   root)
Context: system_u:object_r:admin_home_t:s0
Access: 2021-04-04 17:54:09.700844151 +0800
Modify: 2021-04-04 16:53:30.524854041 +0800
Change: 2021-04-04 16:53:30.524854041 +0800
 Birth: -
```

- **Access：访问时间，也叫 atime**
  - 当文件被访问的时候，这个时间就会发生改变
  - Linux 文件运行的时候查看文件又频繁数量又大，如果每次 atime 发生变化都记入硬盘，或造成很大的压力。必须满足其中一个条件：
    - 自上次 atime 修改后，已达到 86400 秒
    - 发生写操作时
- **Modify：修改时间，也叫 mtime**
  - 当文件内容发生变化的时候，这个时间就会发生改变
- **Change：改变时间，也叫 ctime**
  - 当文件状态被改变的时候，这个时间就会发生修改

# 文件类型

Linux 系统和 Windows 系统有很大的区别，Windows 系统查看文件的后缀名就可以知道这个是什么类型的文件，比如：`test.jpg` 这个是一个图片，如果你在 Windows 上双击打开，就会使用支持查看图片的软件打

开。

Linux 系统就根本不看文件的后缀名，你认为这个是什么文件，你就使用什么工具打开这个文件，如果打开错误，就会报错，看下面的案例

```
[root@localhost ~]# cat file
cat: file: Is a directory
```

当你以为 file 是个文件，使用 cat 命令查看的时候，cat 会提示你这个是个文件夹，你才反应过来，应该使用 cd 命令打开。所以在 Linux 中，我们需要使用和 windows 不一样的方法来判断这个文件的类型。

## 方式一：ls

通过 `ls -l` 查看第一个字母：

| 标识符 | 文件类型 || - | 普通文件(文本文档，二进制文件，压缩文件，电影，图片等等) || d | 目录文件 || b | 块设备文件(块设备)存储设备硬盘，U盘 `/dev/sda` 和 `/dev/sda1` || c | 字符设备文件(字符设备)打印机，终端 `/dev/tty1` 和 `/dev/zero` || s | 套接字文件 || p | 管道文件 || l | 链接文件 |

### 案例演示

```
[root@localhost ~]# type ll
[root@localhost ~]# ll -d /etc/hosts /bin/ls /home /dev/nvme0n1 /dev/tty1
/etc/grub2.cfg /dev/log
-rwxr-xr-x. 1 root root 140872 4月 21 2024 /bin/ls
lrwxrwxrwx. 1 root root      28 11月 8 21:42 /dev/log ->
/run/systemd/journal/dev-log
brw-rw----. 1 root disk 259, 0 11月 8 21:42 /dev/nvme0n1
crw--w----. 1 root tty    4, 1 11月 8 21:42 /dev/tty1
lrwxrwxrwx. 1 root root    22 9月 5 03:00 /etc/grub2.cfg ->
../boot/grub2/grub.cfg
-rw-r--r--. 1 root root    158 6月 23 2020 /etc/hosts
drwxr-xr-x. 10 root root    124 11月 8 09:57 /home
```

对于初学者而言，我们现在只要知道可以通过这样的方式查看文件的类型，并且能够知道 - 和 d 的意思即可。后面在学习的过程中，会慢慢的将所有文件类型都掌握的。

## 方法二：file

file 是专门用来查看文件的类型的命令，有时候也可以使用

### 案例演示

```
[root@localhost ~]# file /etc/hosts
/etc/hosts: ASCII text
[root@localhost ~]# file /bin/ls
/bin/ls: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically
linked (uses shared libs), for GNU/Linux 2.6.32,
```

```
BuildID[sha1]=ceaf496f3aec08afced234f4f36330d3d13a657b, stripped
[root@localhost ~]# file /dev/nvme0n1
/dev/nvme0n1: block special (259/0)
[root@localhost ~]# file /dev/tty1
/dev/tty1: character special
[root@localhost ~]# file /etc/grub2.cfg
/etc/grub2.cfg: symbolic link to `../boot/grub2/grub.cfg'
[root@localhost ~]# file /home
/home: directory
[root@localhost ~]# file /run/dmeventd-client
/run/dmeventd-client: fifo (named pipe)
```

## 方法三：stat

这个命令上面已经介绍过了，在输出结果中也是可以看到文件的类型

## 文件查找

在 windows 中可以在文件管理器中很方便的输入文件名查找文件，然而 Linux 的文件查找功能更加的方便，并且功能更加的强大。

find 命令实时查找工具，通过遍历指定路径下的文件系统完成文件查找

工作特点:

- 查找速度略慢
- 精确查找
- 实时查找
- 可以满足多种条件匹配

find [选项] [路径] [查找条件 + 处理动作]

查找路径：指定具体目录路径，默认是当前文件夹

查找条件：指定的查找标准（文件名/大小/类型/权限等），默认是找出所有文件

处理动作：对符合条件的文件做什么操作，默认输出屏幕

## 查找条件

根据文件名查找

```
[root@localhost ~]# find /etc -name "ens160.nmconnection"
[root@localhost ~]# find /etc -iname "ens160.nmconnection"      # 忽略大小写
[root@localhost ~]# find /etc -iname "ens*"
```

按文件大小

```
[root@localhost ~]# find /etc -size +5M      # 大于5M
[root@localhost ~]# find /etc -size 5M      # 等于5M
[root@localhost ~]# find /etc -size -5M     # 小于5M
[root@localhost ~]# find /etc -size +5M -ls  # 找到的处理动作-ls
```

### 指定查找的目录深度

```
[root@localhost ~]# find / -maxdepth 3 -a -name "ens160.nmconnection"  #
最大查找深度
# -a是同时满足, -o是或
[root@localhost ~]# find / -mindepth 3 -a -name "ens160.nmconnection"  #
最小查找深度
```

### 按时间找

```
[root@localhost ~]# find /etc -mtime +5      # 修改时间超过5天
[root@localhost ~]# find /etc -mtime 5      # 修改时间等于5天
[root@localhost ~]# find /etc -mtime -5     # 修改时间5天以内
```

按照文件属主、属组找，文件的属主和属组，会在下一篇详细讲解。

```
[root@localhost ~]# find /home -user xwz     # 属主是xwz的文件
[root@localhost ~]# find /home -group xwz
[root@localhost ~]# find /home -user xwz -group xwz
[root@localhost ~]# find /home -user xwz -a -group root
[root@localhost ~]# find /home -user xwz -o -group root
[root@localhost ~]# find /home -nouser      # 没有属主的文件
[root@localhost ~]# find /home -nogroup     # 没有属组的文件
```

### 按文件类型

```
[root@localhost ~]# find /dev -type d
```

按文件权限，文件权限会在下一篇详细讲解

```
[root@localhost ~]# find / -perm 644 -ls
[root@localhost ~]# find / -perm -644 -ls  # 权限小于644的
```

### 按正则表达式

```
[root@localhost ~]# find /etc -regex '.*ens[0-9][0-9][0-9].*'
# .*      任意多个字符
# [0-9]    任意一个数字
```

- 条件组合
  - -a: 多个条件and并列
  - -o: 多个条件or并列
  - -not: 条件取反

处理动作

动作	含义
-print	默认的处理动作，显示至屏幕
-ls	对查找到的文件执行 <code>ls -l</code> 命令
-delete	删除查找到的文件
-fls /path/to/filename	查找到的所有文件的长格式信息保存至指定文件中
{}	用于引用查找到的文件名称自身
-exec	允许对找到的每个文件执行一个命令

下面的相关案例大家学习完后续用户权限管理之后，就可以完全看的懂了

相关案例

- 查找到root目录下面以.log结尾的文件，并且复制到/home/dir1中

```
[root@localhost ~]# find /root -name "*.log" -exec mv {} /home/dir1 \;
```

- 查找/var目录下属主为root，且属组为mail的所有文件或目录

```
[root@localhost ~]# find /var -user root -group mail
```

- 查找/usr目录下不属于root，bin或ftp用户的所有文件或目录

```
[root@localhost ~]# find /usr -not -user root -a -not -user bin -a -not -user ftp
[root@localhost ~]# find /usr -not \( -user root -o -user bin -o -user ftp \)
```

- 查找/etc目录下最近一周内容曾被修改过的文件或目录

```
[root@localhost ~]# find /etc -mtime -7
```

- 查找当前系统上没有属主或属组，且最近一周内曾被访问过的文件或目录

```
[root@localhost ~]# find / \( -nouser -o -nogroup \) -a -atime -7
```

- 查找/etc目录下大于1M且类型为普通文件的所有文件或目录

```
[root@localhost ~]# find /etc -size +1M -type f
```

- 查找/etc目录下所有用户都没有写权限的文件

```
[root@localhost ~]# find /etc -not -perm /222
```

- 查找/etc目录下至少一类用户没有执行权限的文件

```
[root@localhost ~]# find /etc -not -perm -111
```

- 查找/etc/init.d目录下，所有用户都有执行权限，且其它用户写权限的文件

```
[root@localhost ~]# find /etc/init.d -perm -113
```