

其他命令

目标

- 查找文件
 - `find`
- 软链接
 - `ln`
- 打包和压缩
 - `tar`
- 软件安装
 - `apt-get`

01. 查找文件

- `find` 命令功能非常强大，通常用来在 **特定的目录下搜索** 符合条件的文件

序号	命令	作用
01	<code>find [路径] -name "*.py"</code>	查找指定路径下扩展名是 <code>.py</code> 的文件，包括子目录

- 如果省略路径，表示在当前文件夹下查找
- 之前学习的通配符，在使用 `find` 命令时同时可用
- 有关 `find` 的高级使用，在就业班会讲

演练目标

- 1. 搜索桌面目录下，文件名包含 `1` 的文件

```
find -name "*1*"
```

- 2. 搜索桌面目录下，所有以 `.txt` 为扩展名的文件

```
find -name "*.txt"
```

- 3. 搜索桌面目录下，以数字 `1` 开头的文件

```
find -name "1*"
```

02. 软链接

序号	命令	作用
01	<code>ln -s 被链接的源文件 链接文件</code>	建立文件的软链接，用通俗的方式讲类似于 Windows 下的快捷方式

- 注意：
 - 1. 没有 `-s` 选项建立的是一个 **硬链接文件**
 - 两个文件占用相同大小的硬盘空间，工作中几乎不会建立文件的硬链接
 - 2. 源文件要使用绝对路径，不能使用相对路径，这样可以方便移动链接文件后，仍然能够正常使用

演练目标

- 1. 将桌面目录下的 `01.py` 移动到 `demo/b/c` 目录下
- 2. 在桌面目录下新建 `01.py` 的软链接 `FirstPython`
 - 分别使用 **相对路径** 和 **绝对路径** 建立 `FirstPython` 的软链接
- 3. 将 `FirstPython` 移动到 `demo` 目录下，对比使用 **相对路径** 和 **绝对路径** 的区别

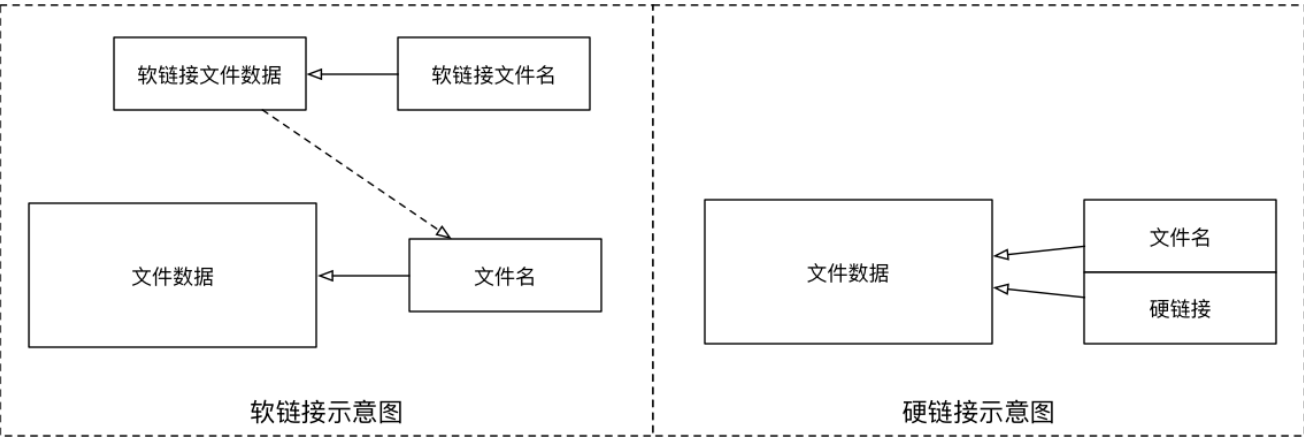
硬链接简介（知道）

- 在使用 `ln` 创建链接时，如果没有 `-s` 选项，会创建一个 **硬链接**，而不是软链接

硬链接演练

- 1. 在 `~/Desktop/demo` 目录下建立 `~/Desktop/demo/b/c/01.py` 的硬链接 `01_hard`
- 2. 使用 `ls -l` 查看文件的**硬链接数**（硬链接——有多少种方式可以访问文件或者目录）
- 3. 删除 `~/Desktop/demo/b/c/01.py`，并且使用 `tree` 来确认 `demo` 目录下的三个链接文件

文件软硬链接的示意图



在 Linux 中，**文件名** 和 **文件的数据** 是分开存储的

- 提示：
 - 在 Linux 中，只有文件的 `硬链接数 == 0` 才会被删除
 - 使用 `ls -l` 可以查看一个文件的硬链接的数量
 - 在日常工作中，几乎不会建立文件的硬链接，知道即可

03. 打包压缩

- **打包压缩** 是日常工作中备份文件的一种方式
- 在不同操作系统中，常用的打包压缩方式是不同的
 - Windows 常用 rar
 - Mac 常用 zip
 - Linux 常用 tar.gz

3.1 打包 / 解包

- `tar` 是 Linux 中最常用的 **备份工具**，此命令可以 **把一系列文件** 打包到 **一个大文件中**，也可以把一个 **打包的大文件恢复成一系列文件**
- `tar` 的命令格式如下：

```
# 打包文件
tar -cvf 打包文件.tar 被打包的文件 / 路径...

# 解包文件
tar -xvf 打包文件.tar
```

- `tar` 选项说明

选项	含义
c	生成档案文件，创建打包文件
x	解开档案文件
v	列出归档解档的详细过程，显示进度
f	指定档案文件名称，f 后面一定是 .tar 文件，所以必须放选项最后

注意：f 选项必须放在最后，其他选项顺序可以随意

打包解包演练

1. 删除桌面下的所有内容
2. 在桌面下新建三个空白文件 01.py 、 02.py 、 03.py
3. 将这三个文件打一个 py.tar 的包
4. 新建 tar 目录，并且将 py.tar 移动到 tar 目录下
5. 解包 py.tar

3.2 压缩 / 解压缩

1) gzip

- `tar` 与 `gzip` 命令结合可以使用实现文件 **打包和压缩**
 - `tar` 只负责打包文件，但不压缩
 - 用 `gzip` 压缩 `tar` 打包后的文件，其扩展名一般用 xxx.tar.gz

在 `Linux` 中，最常见的压缩文件格式就是 `xxx.tar.gz`

- 在 `tar` 命令中有一个选项 `-z` 可以调用 `gzip`，从而可以方便的实现压缩和解压缩的功能
- 命令格式如下：

```
# 压缩文件
tar -zcvf 打包文件.tar.gz 被压缩的文件 / 路径...

# 解压缩文件
tar -zxvf 打包文件.tar.gz

# 解压缩到指定路径
tar -zxvf 打包文件.tar.gz -C 目标路径
```

选项	含义
-C	解压缩到指定目录，注意：要解压缩的目录必须存在

2) `bzip2(two)`

- `tar` 与 `bzip2` 命令结合可以使用实现文件 **打包和压缩**（用法和 `gzip` 一样）
 - `tar` 只负责打包文件，但不压缩，
 - 用 `bzip2` 压缩 `tar` 打包后的文件，其扩展名一般用 `xxx.tar.bz2`
- 在 `tar` 命令中有一个选项 `-j` 可以调用 `bzip2`，从而可以方便的实现压缩和解压缩的功能
- 命令格式如下：

```
# 压缩文件
tar -jcvf 打包文件.tar.bz2 被压缩的文件 / 路径...

# 解压缩文件
tar -jxvf 打包文件.tar.bz2
```

04. 软件安装

4.1 通过 `apt` 安装 / 卸载软件

- `apt` 是 `Advanced Packaging Tool`，是 `Linux` 下的一款安装包管理工具
- 可以在终端中方便的 **安装 / 卸载 / 更新软件包**

```
# 1. 安装软件
$ sudo apt install 软件包

# 2. 卸载软件
$ sudo apt remove 软件名

# 3. 更新已安装的包
$ sudo apt upgrade
```

安装演练

```
# 一个小火车提示
$ sudo apt install sl

# 一个比较漂亮的查看当前进程排名的软件
$ sudo apt install htop
```

4.2 配置软件源

- 如果希望在 `ubuntu` 中安装软件，**更加快速**，可以通过设置**镜像源**，选择一个访问网速更快的服务器，来提供软件下载 / 安装服务
- 提示：更换服务器之后，需要一个相对比较长时间的更新过程，需要耐心等待。更新完成后，再安装软件都会从新设置的服务器下载软件了

所谓镜像源，就是**所有服务器的内容是相同的（镜像）**，但是根据所在位置不同，国内服务器通常速度会更快一些！

