

Linux学习文档

第一章、Linux软件配置

- 1.1 虚拟机配置
- 1.2 使用WSL在windows配置ubuntu系统

第二章、Linux基础命令

- 2.1 linux命令格式
- 2.2 ls命令——显示路径文件信息
- 2.3 cd命令——工作目录切换
- 2.4 pwd命令——查看当前工作目录
- 2.5 mkdir命令——创建文件夹
- 2.6 touch命令——创建文件
- 2.7 cat命令——查看文件内容
- 2.8 more命令——查看文件内容
- 2.9 cp命令——复制文件、文件夹
- 2.10 mv命令——移动文件、文件夹
- 2.11 rm命令——删除文件、文件夹
- 2.12 which命令——查找命令的程序文件
- 2.13 find命令——按文件名查找文件
- 2.14 find命令——按文件大小查找文件
- 2.15 grep命令——关键字过滤文件行
- 2.16 wc命令——数量统计
- 2.17 | 管道符——将左边命令的结果，作为右边命令的输入
- 2.18 echo命令——在命令行内输出指定内容
- 2.19 > 重定向符——将结果输入到文件中
- 2.20 tail命令——查看文件尾部内容，跟踪文件的最新更改
- 2.21 vim编辑器
- 2.22 基础命令补充
 - 2.22.1 help
 - 2.22.2 man

第三章、用户和权限操作

- 3.1 su和exit命令——切换用户和退回用户
- 3.2 sudo命令——为普通的命令授权，临时以root身份执行
- 3.3 用户和用户组
 - 3.3.1 groupadd命令——创建用户组
 - 3.3.2 groupdel命令——删除用户组
 - 3.3.3 useradd命令——创建用户
 - 3.3.4 userdel命令——删除用户
 - 3.3.5 id命令——查看用户所属组
 - 3.3.6 usermod命令——修改用户所属组，将指定用户加入指定用户组
 - 3.3.7 getent命令——查看当前系统中有哪些用户
 - 3.3.8 getent命令——查看当前系统中有哪些用户组

3.4 查看权限控制

- 3.5 chmod命令——修改权限控制
- 3.6 chown命令——修改权限控制

第四章、linux实用操作

- 4.1 ctrl+c命令——强制停止程序
- 4.2 ctrl+d命令——退出或登出
- 4.3 history命令——查看历史命令
- 4.4 ! 命令前缀——自动执行上一次匹配前缀的命令
- 4.5 ctrl+r命令——输入内容去匹配历史命令
- 4.6 光标移动快捷键
- 4.7 ctrl+l或者clear命令——清屏
- 4.8 yum命令——centos系统软件安装

4.9 apt命令——ubuntu系统软件安装
4.10 systemctl命令——控制软件：启动、停止、开机自启
4.11 ln命令——创建软连接
4.12 date命令——查看系统的时间
4.13 使用ln和date修改linux时区
4.14 ntp程序——自动校准系统时间
4.15 ifconfig命令——查看ip
4.16 hostname命令——查看主机名
4.17 hostnamectl命令——修改主机名
4.18 finalshell通过主机名连接linux
4.19 固定ip地址
4.20 ping命令——检查指定的网络服务器是否是可联通状态
4.21 wget命令——命令行内下载网络文件
4.22 curl命令——发送http网络请求，可用于：下载文件、获取信息等
4.23 nmap命令——查看端口占用
4.24 netstat命令——查看指定端口的占用情况
4.25 ps命令——查看Linux系统中的进程信息
4.26 kill命令——关闭进程
4.27 top命令——查看CPU、内存使用情况
4.28 df命令——磁盘信息监控
4.29 iostat命令——查看CPU、磁盘的相关信息
4.30 sar命令——网络状态监控
4.31 环境变量介绍
4.32 \$符号——环境变量操作
4.33 自定义环境变量PATH
4.34 rz、sz命令——finalshell快速下载、上传
 4.34.1 rz、sz命令安装
 4.34.2 rz命令——上传
 4.34.3 sz命令——下载
4.35 tar、zip、unzip命令——文件压缩与解压
 4.35.1 tar命令——压缩与解压
 4.35.2 zip命令——压缩成zip文件
 4.35.3 unzip命令——zip文件解压缩

Linux学习文档

第一章、Linux软件配置

1.1 虚拟机配置

远程连接虚拟机（通过finalshell操作虚拟机）：

-
- 查看虚拟机ip：ifconfig
- 打开finalshell进行配置

1.2 使用WSL在windows配置ubuntu系统

- 开启wsl：右键windows-->应用和功能-->程序和功能-->启用或关闭windows功能-->勾上适用于linux的windows子系统
- 安装ubuntu系统：打开应用商店-->下载ubuntu-->设置账号、密码
- 安装terminal：打开应用商店-->下载windows terminal

第二章、Linux基础命令

2.1 linux命令格式

无论是什么命令，用于什么用途，在Linux中，命令有其通用的格式：

```
command [-options] [parameter]
```

- command：命令本身
- -options：[可选，非必填]命令的一些选项，可以通过选项控制命令的行为细节
- parameter：[可选，非必填]命令的参数，多数用于命令的指向目标等

2.2 ls命令——显示路径文件信息

```
ls [-a -l -h] [linux路径]
```

- **-a** 显示隐藏文件（夹） **-l** 列表形式展示 **-h** 显示文件（夹）大小
- Linux路径是此命令可选的参数

当不使用选项和参数，直接使用ls命令本体，表示：以平铺形式，列出当前工作目录下的内容

选项可组合使用： (h必须和l一起使用才能显示文件大小)

```
ls -a  
ls -h  
ls -ah
```

2.3 cd命令——工作目录切换

cd命令来自英文：Change Directory

```
cd [linux路径]
```

- cd命令无需选项，只有参数，表示要切换到哪个目录下
- cd命令直接执行，不写参数，表示回到用户的HOME目录

2.4 pwd命令——查看当前工作目录

pwd命令来自：Print Work Directory

```
pwd
```

- pwd命令，无选项，无参数，直接输入pwd即可

2.5 mkdir命令——创建文件夹

mkdir来自英文：Make Directory

```
mkdir [-p] linux路径
```

- 参数必填，表示Linux路径，即要创建的文件夹的路径，相对路径或绝对路径均可
- -p选项可选，表示自动创建不存在的父目录，适用于创建连续多层级的目录

2.6 touch命令——创建文件

```
touch linux路径
```

- touch命令无选项，参数必填，表示要创建的文件路径，相对、绝对、特殊路径符均可以使用

2.7 cat命令——查看文件内容

```
cat linux路径
```

- cat同样没有选项，只有必填参数，参数表示：被查看的文件路径，相对、绝对、特殊路径符都可以使用

2.8 more命令——查看文件内容

more命令同样可以查看文件内容，同cat不同的是：

- cat是直接将内容全部显示出来
- more支持翻页，如果文件内容过多，可以一页页的展示

```
more linux路径
```

- 同样没有选项，只有必填参数，参数表示：被查看的文件路径，相对、绝对、特殊路径符都可以使用
- 在查看的过程中，通过空格翻页
- 通过q退出查看

2.9 cp命令——复制文件、文件夹

cp命令可以用于复制文件\文件夹，cp命令来自英文单词：copy

```
cp [-r] 参数1 参数2
```

- -r选项，可选，用于复制文件夹使用，表示递归

- 参数1, Linux路径, 表示被复制的文件或文件夹
- 参数2, Linux路径, 表示要复制去的地方

效果类似于windows系统中的**复制+粘贴+重命名**

2.10 mv命令——移动文件、文件夹

mv命令可以用于移动文件\文件夹, mv命令来自英文单词: move

```
mv 参数1 参数2
```

- 参数1, Linux路径, 表示被移动的文件或文件夹
- 参数2, Linux路径, 表示要移动去的地方, 如果目标不存在, 则进行改名, 确保目标存在

效果类似于windows系统中的**剪切+粘贴+重命名**

2.11 rm命令——删除文件、文件夹

rm命令来自英文单词: remove

```
rm [-r -f] 参数1 参数2 ..... 参数n
```

- 同cp命令一样, -r选项用于删除文件夹
- -f表示force, 强制删除 (不会弹出提示确认信息)

普通用户删除内容不会弹出提示, 只有root管理员用户删除内容会有提示

所以一般普通用户用不到-f选项

- 参数1、参数2、.....、参数n 表示要删除的文件或文件夹路径, 按照空格隔开

2.12 which命令——查找命令的程序文件

通过which命令, 查看所使用的一系列命令的程序文件存放在哪里

```
which 要查找的命令
```

2.13 find命令——按文件名查找文件

```
find 起始路径 -name "搜索的文件名"
```

被查找文件名, 支持使用通配符 * 来做模糊查询。

- 符号* 表示通配符, 即匹配任意内容 (包含空), 示例:
- test*, 表示匹配任何以test开头的内容
- *test, 表示匹配任何以test结尾的内容
- *test*, 表示匹配任何包含test的内容

基于通配符的含义, 可以结合find命令做文件的模糊查询。

2.14 find命令——按文件大小查找文件

```
find 起始路径 -size +|-n[kMG]
```

- +、- 表示大于和小于
- n表示大小数字
- kMG表示大小单位，k(小写字母)表示kb，M表示MB，G表示GB

示例：

- 查找小于10KB的文件：find / -size -10k
- 查找大于100MB的文件：find / -size +100M
- 查找大于1GB的文件：find / -size +1G

2.15 grep命令——关键字过滤文件行

```
grep [-n] 关键字 文件路径
```

- 选项-n，可选，表示在结果中显示匹配的行的行号。
- 参数，关键字，必填，表示过滤的关键字，带有空格或其它特殊符号，建议使用""将关键字包围起来
- 参数，文件路径，必填，表示要过滤内容的文件路径，**可作为内容输入端口**

示例：过滤itcast关键字

```
[itheima@bogon ~]$ grep "itheima" itheima.txt
itheima is a brand of itcast.
```

2.16 wc命令——数量统计

```
wc [-c -m -l -w] 文件路径
```

- 选项，-c，统计bytes数量
- 选项，-m，统计字符数量
- 选项，-l，统计行数
- 选项，-w，统计单词数量
- 参数，文件路径，被统计的文件，可作为内容输入端口

不带选项：

```
[itheima@bogon ~]$ wc itheima.txt
2 11 59 itheima.txt
[itheima@bogon ~]$
行 单 字
数 词 节
数 数
```

2.17 | 管道符——将左边命令的结果，作为右边命令的输入

一般和grep命令或其他命令配合使用

```
[itheima@bogon ~]$ cat itheima.txt | grep itheima  
itheima is a brand of itcast.  
[itheima@bogon ~]$
```

被过滤文件（输入）

- cat itheima.txt的输出结果（文件内容）
- 作为右边grep命令的输入（被过滤文件）
- 可嵌套使用

2.18 echo命令——在命令行内输出指定内容

echo 输出的内容

- 无需选项，只有一个参数，表示要输出的内容，复杂内容可以用""包围
- 带有空格或\等特殊符号，建议使用双引号包围
- 通过将命令用反引号（通常也称之为飘号）`将其包围，被`包围的内容，会被作为命令执行，而非普通字符

```
[itheima@bogon ~]$ echo `pwd`  
/home/itheima
```

2.19 > 重定向符——将结果输入到文件中

我们再来学习两个特殊符号，重定向符：>和>>

- >，将左侧命令的结果，**覆盖**写入到符号右侧指定的文件中
- >>，将左侧命令的结果，**追加**写入到符号右侧指定的文件中

```
[itheima@bogon ~]$ echo "Hello Linux" > itheima.txt  
[itheima@bogon ~]$ cat itheima.txt  
Hello Linux
```

```
[itheima@bogon ~]$ echo "Hello itcast" >> itheima.txt  
[itheima@bogon ~]$ cat itheima.txt  
Hello itcast  
Hello itcast
```

2.20 tail命令——查看文件尾部内容，跟踪文件的最新更改

tail [-f -num] linux路径

- 参数，Linux路径，表示被跟踪的文件路径
- 选项，-f，表示持续跟踪
- 选项，-num，表示，查看尾部多少行，不填默认10行

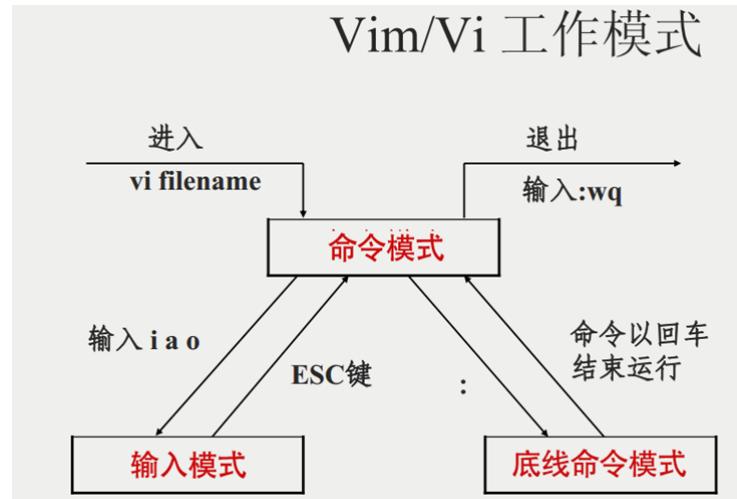
2.21 vim编辑器

vi\vim是visual interface的简称, 是Linux中最经典的文本编辑器

同图形化界面中的 文本编辑器一样, vi是命令行下对文本文件进行编辑的绝佳选择。

vim 是 vi 的加强版本, 兼容 vi 的所有指令, 不仅能编辑文本, 而且还具有 shell 程序编辑的功能, 可以不同颜色的字体来辨别语法的正确性, 极大方便了程序的设计和编辑性。

vi\vim编辑器的三种工作模式:



- 命令模式 (Command mode)

命令模式下, 所敲的按键编辑器都理解为命令, 以命令驱动执行不同的功能。此模型下, 不能自由进行文本编辑。

- 输入模式 (Insert mode)

也就是所谓的编辑模式、插入模式。此模式下, 可以对文件内容进行自由编辑。

- 底线命令模式 (Last line mode)

以: 开始, 通常用于文件的保存、退出。

快捷键:

- 命令模式

模式	命令	描述
命令模式	i	在当前光标位置进入 输入模式
命令模式	a	在当前光标位置 之后 进入 输入模式
命令模式	I	在当前行的开头，进入 输入模式
命令模式	A	在当前行的结尾，进入 输入模式
命令模式	o	在当前光标下一行进入 输入模式
命令模式	O	在当前光标上一行进入 输入模式
输入模式	esc	任何情况下输入 esc 都能回到命令模式
命令模式	键盘上、键盘k	向上移动光标
命令模式	键盘下、键盘j	向下移动光标
命令模式	键盘左、键盘h	向左移动光标
命令模式	键盘右、键盘l	向后移动光标
命令模式	0	移动光标到当前行的开头
命令模式	\$	移动光标到当前行的结尾
命令模式	pageup (PgUp)	向上翻页
命令模式	pangdown (PgDn)	向下翻页
命令模式	/	进入搜索模式
命令模式	n	向下继续搜索
命令模式	N	向上继续搜索
命令模式	dd	删除光标所在行的内容
命令模式	ndd	n是数字，表示删除当前光标向下n行
命令模式	yy	复制当前行
命令模式	nyy	n是数字，复制当前行和下面的n行
命令模式	p	粘贴复制的内容
命令模式	u	撤销修改
命令模式	ctrl + r	反向撤销修改
命令模式	gg	跳到首行
命令模式	G	跳到行尾
命令模式	dG	从当前行开始，向下全部删除
命令模式	dgg	从当前行开始，向上全部删除
命令模式	d\$	从当前光标开始，删除到本行的结尾
命令模式	d0	从当前光标开始，删除到本行的开头

- 底线模式

底线命令模式	:wq	保存并退出
底线命令模式	:q	仅退出
底线命令模式	:q!	强制退出
底线命令模式	:w	仅保存
底线命令模式	:set nu	显示行号
底线命令模式	:set paste	设置粘贴模式

2.22 基础命令补充

2.22.1 help

任何命令都支持：--help 选项，可以通过这个选项，查看命令的帮助。

如：ls --help，会列出ls命令的帮助文档

2.22.2 man

如果想要查看命令的详细手册，可以通过man (manual, 手册) 命令查看

比如：

- man ls，就是查看ls命令的详细手册
- man cd，就是查看cd命令的详细手册

大多数手册都是全英文的，如果阅读吃力，可以通过重定向符：man ls > ls-man.txt，输出手册到文件然后通过翻译软件翻译内容查看哦

第三章、用户和权限操作

3.1 su和exit命令——切换用户和退回用户

su命令就是用于账户切换的系统命令，其来源英文单词：Switch User

su [-] 用户名

- -符号是可选的，表示是否在切换用户后加载环境变量（后续讲解），建议带上
- 参数：用户名，表示要切换的用户，用户名也可以省略，省略表示切换到root
- 切换用户后，可以通过exit命令退回上一个用户，也可以使用快捷键：ctrl + d
- 使用普通用户，切换到其它用户需要输入密码，如切换到root用户
- 使用root用户切换到其它用户，无需密码，可以直接切换

3.2 sudo命令——为普通的命令授权，临时以root身份执行

`sudo 其他命令`

- 在其它命令之前，带上sudo，即可为这一条命令临时赋予root授权
- 但是并不是所有的用户，都有权利使用sudo，我们需要为普通用户配置sudo认证

普通用户配置sudo认证：

1. 切换到root用户，执行visudo命令，会自动通过vi编辑器打开：/etc/sudoers
2. 在文件的最后添加：

```
itheima ALL=(ALL) NOPASSWD: ALL
```

3. 其中最后的NOPASSWD:ALL 表示使用sudo命令，无需输入密码
4. 最后通过 wq 保存
5. 切换回普通用户

```
[itheima@localhost ~]$ sudo ls /root  
anaconda-ks.cfg original-ks.cfg  
[itheima@localhost ~]$ sudo mkdir /test  
[itheima@localhost ~]$
```

6. 执行的命令，均以root运行

3.3 用户和用户组

Linux系统中可以：

- 配置多个用户
- 配置多个用户组
- 用户可以加入多个用户组中

Linux中关于权限的管控级别有2个级别，分别是：

- 针对用户的权限控制
- 针对用户组的权限控制

比如，针对某文件，可以控制用户的权限，也可以控制用户组的权限。

用户组管理：

3.3.1 groupadd命令——创建用户组

`groupadd 用户组名`

3.3.2 groupdel命令——删除用户组

`groupdel 用户组名`

用户管理：

3.3.3 useradd命令——创建用户

```
useradd [-g -d] 用户名
```

- 选项：-g指定用户的组，不指定-g，会创建同名组并自动加入，指定-g需要组已经存在，如已存在同名组，必须使用-g
- 选项：-d指定用户HOME路径，不指定，HOME目录默认在：/home/用户名

3.3.4 userdel命令——删除用户

```
userdel [-r] 用户名
```

- 选项：-r，删除用户的HOME目录，不使用-r，删除用户时，HOME目录保留

3.3.5 id命令——查看用户所属组

```
id [用户名]
```

- 参数：用户名，被查看的用户，如果不提供则查看自身

3.3.6 usermod命令——修改用户所属组，将指定用户加入指定用户组

```
usermod -aG 用户组名 用户名
```

3.3.7 getent命令——查看当前系统中有哪些用户

```
getent passwd
```

```
[itheima@localhost ~]$ getent passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin-sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
itheima:x:1000:1000:itheima:/home/itheima:/bin/bash
```

共有7份信息，分别是：

用户名:密码(x):用户ID:组ID:描述信息(无用):HOME目录:执行终端(默认bash)

3.3.8 getent命令——查看当前系统中有哪些用户组

```
getent group
```

```
[itheima@localhost ~]$ getent group
root:x:0:
bin:x:1:
daemon:x:2:
sys:x:3:
adm:x:4:
tty:x:5:
disk:x:6:

itheima:x:1000:
```

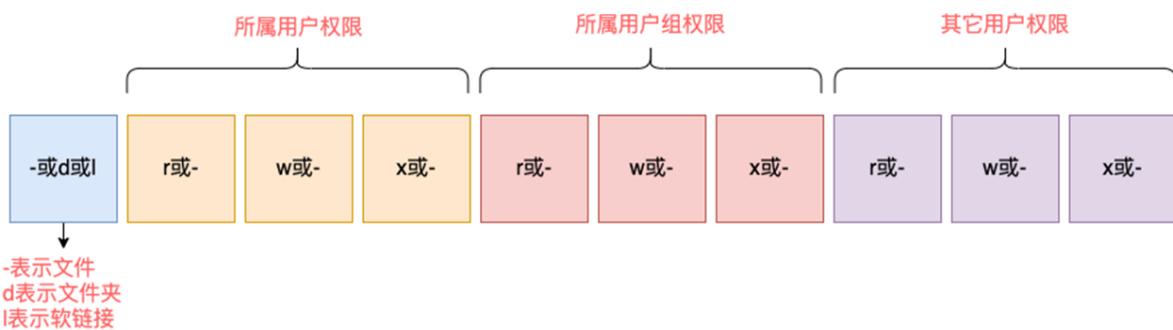
包含3份信息，组名称:组认证(显示为x):组ID

3.4 查看权限控制

通过ls -l 可以以列表形式查看内容，并显示权限细节

```
[itheima@localhost ~]$ ls -l
总用 量 0          2          3
drwxr-xr-x. 3 itheima itheima 37 9月 23 03:17 Desktop
drwxr-xr-x. 2 itheima itheima 6 9月 22 23:57 Documents
drwxr-xr-x. 2 itheima itheima 6 9月 22 23:57 Downloads
-rw-rw-r--. 1 itheima itheima 0 9月 26 00:16 hello.txt
drwxr-xr-x. 2 itheima itheima 6 9月 22 23:57 Music
drwxr-xr-x. 2 itheima itheima 6 9月 22 23:57 Pictures
drwxr-xr-x. 2 itheima itheima 6 9月 22 23:57 Public
drwxr-xr-x. 2 itheima itheima 6 9月 22 23:57 Templates
drwxr-xr-x. 2 itheima itheima 6 9月 22 23:57 Videos
```

- 序号1，表示文件、文件夹的权限控制信息
- 序号2，表示文件、文件夹所属用户
- 序号3，表示文件、文件夹所属用户组



举例：drwxr-xr-x，表示：

- 这是一个文件夹，首字母d表示
- 所属用户(右上角图序号2)的权限是：有r有w有x， rwx
- 所属用户组(右上角图序号3)的权限是：有r无w有x， r-x (-表示无此权限)
- 其它用户的权限是：有r无w有x， r-x

那么，rwx到底代表什么呢？

- r表示读权限
- w表示写权限
- x表示执行权限

针对文件、文件夹的不同，rwx的含义有细微差别

- r，针对文件可以查看文件内容
- 针对文件夹，可以查看文件夹内容，如ls命令

- w, 针对文件表示可以修改此文件
- 针对文件夹, 可以在文件夹内: 创建、删除、改名等操作
- x, 针对文件表示可以将文件作为程序执行
- 针对文件夹, 表示可以更改工作目录到此文件夹, 即cd进入

3.5 chmod命令——修改权限控制

使用chmod命令, 可以修改文件、文件夹的权限信息。注意, 只有文件、文件夹的所属用户或root用户可以修改。

```
chmod [-R] 权限 文件或文件夹
```

- 选项: -R, 对文件夹内的全部内容应用同样的操作

示例:

- chmod u=rwx,g=rx,o=x hello.txt , 将文件权限修改为: rwxr-x--x
其中: u表示user所属用户权限, g表示group组权限, o表示other其它用户权限
- chmod -R u=rwx,g=rx,o=x test, 将文件夹test以及文件夹内全部内容权限设置为: rwxr-x--x

除此之外, 还有快捷写法: chmod 751 hello.txt (r记为4, w记为2, x记为1)

3.6 chown命令——修改权限控制

使用chown命令, 可以修改文件、文件夹的所属用户和用户组, 普通用户无法修改所属为其它用户或组, 所以此命令只适用于root用户执行

```
chmod [-R] [用户] [:] [用户组] 文件或文件夹
```

- 选项, -R, 同chmod, 对文件夹内全部内容应用相同规则
- 选项, 用户, 修改所属用户
- 选项, 用户组, 修改所属用户组
- :用于分隔用户和用户组

示例:

- chown root hello.txt, 将hello.txt所属用户修改为root
- chown :root hello.txt, 将hello.txt所属用户组修改为root
- chown root:itheima hello.txt, 将hello.txt所属用户修改为root, 用户组修改为itheima
- chown -R root test, 将文件夹test的所属用户修改为root并对文件夹内全部内容应用同样规则

第四章、linux实用操作

4.1 ctrl+c命令——强制停止程序

- Linux某些程序的运行，如果想要强制停止它，可以使用快捷键ctrl + c

```
[itheima@localhost ~]$ tail  
^C  
[itheima@localhost ~]$
```

- 命令输入错误，也可以通过快捷键ctrl + c，退出当前输入，重新输入

```
[itheima@localhost ~]$ cd /a/b/c/d/e^C  
[itheima@localhost ~]$
```

4.2 ctrl+d命令——退出或登出

- 可以通过快捷键：ctrl + d，退出账户的登录
- 或者退出某些特定程序的专属页面

4.3 history命令——查看历史命令

```
history
```

- 可以通过history命令，查看历史输入过的命令

4.4 ! 命令前缀——自动执行上一次匹配前缀的命令

- 可以通过：!命令前缀，自动执行上一次匹配前缀的命令

```
[itheima@localhost ~]$ history  
1 ls /  
2 ifconfig  
3 ls  
4 ifconfig  
5 set +o history;  
6 wc  
7 tail  
8 set +o history;  
9 echo "hello linux" >> test.txt  
10 echo "im linux" >> test.txt  
11 echo "123" >> test.txt  
12 tail  
13 su - root  
14 python  
15 history  
[itheima@localhost ~]$ !py  
python  
Python 2.7.5 (default, Oct 30 2018, 23:45:53)  
[GCC 4.8.5 20150623 (Red Hat 4.8.5-36)] on linux2  
Type "help", "copyright", "credits" or "license" for more information.  
>>>
```

4.5 ctrl+r命令——输入内容去匹配历史命令

如果搜索到的内容是你需要的，那么：

- 回车键可以直接执行
- 键盘左右键，可以得到此命令（不执行）

4.6 光标移动快捷键

- `ctrl + a`, 跳到命令开头
- `ctrl + e`, 跳到命令结尾
- `ctrl + 键盘左键`, 向左跳一个单词
- `ctrl + 键盘右键`, 向右跳一个单词

4.7 `ctrl+l`或者`clear`命令——清屏

- 通过快捷键`ctrl + l`, 可以清空终端内容
- 或通过命令`clear`得到同样效果

4.8 `yum`命令——centos系统软件安装

RPM包软件管理器, 用于自动化安装配置Linux软件, 并可以自动解决依赖问题。

```
yum [-y] [install | remove | search] 软件名称
```

- 选项: `-y`, 自动确认, 无需手动确认安装或卸载过程
- `install`: 安装
- `remove`: 卸载
- `search`: 搜索

`yum`命令需要root权限哦, 可以`su`切换到root, 或使用`sudo`提权。

`yum`命令需要联网

4.9 `apt`命令——ubuntu系统软件安装

```
apt [-y] [install | remove | search] 软件名称
```

用法和`yum`一致, 同样需要root权限

- `apt install wget`, 安装`wget`
- `apt remove wget`, 移除`wget`
- `apt search wget`, 搜索`wget`

4.10 `systemctl`命令——控制软件: 启动、停止、开机自启

Linux系统很多软件(内置或第三方)均支持使用`systemctl`命令控制: 启动、停止、开机自启

能够被`systemctl`管理的软件, 一般也称之为: 服务

```
systemctl start | stop | status | enable | disable 服务名
```

- `start` 启动
- `stop` 关闭
- `status` 查看状态
- `enable` 开启开机自启
- `disable` 关闭开机自启

系统内置的服务比较多, 比如:

- `NetworkManager`, 主网络服务

- network, 副网络服务
- firewalld, 防火墙服务
- sshd, ssh服务 (FinalShell远程登录Linux使用的就是这个服务)

除了内置的服务以外，部分第三方软件安装后也可以以systemctl进行控制。

部分软件安装后没有自动集成到systemctl中，我们可以手动添加。

4.11 ln命令——创建软连接

在系统中创建软链接，可以将文件、文件夹链接到其它位置。

类似Windows系统中的《快捷方式》

```
ln -s 参数1 参数2
```

- -s选项，创建软连接
- 参数1：被链接的文件或文件夹
- 参数2：要链接去的目的地

实例：

- ln -s /etc/yum.conf ~/yum.conf
- ln -s /etc/yum ~/yum

4.12 date命令——查看系统的时间

```
date [-d] [+格式化字符串]
```

- -d 按照给定的字符串显示日期，一般用于日期计算

date -d "+1 day" +%Y%m%d	# 显示后一天的日期
date -d "-1 day" +%Y%m%d	# 显示前一天的日期
date -d "-1 month" +%Y%m%d	# 显示上一月的日期
date -d "+1 month" +%Y%m%d	# 显示下一月的日期
date -d "-1 year" +%Y%m%d	# 显示前一年的日期
date -d "+1 year" +%Y%m%d	# 显示下一年的日期

其中支持的时间标记为：

year 年

month 月

day 天

hour 小时

minute 分钟

second 秒

-d选项可以和 格式化字符串配合一起使用哦

- 格式化字符串：通过特定的字符串标记，来控制显示的日期格式

%Y 年

```
%y 年份后两位数字 (00..99)  
%m 月份 (01..12)  
%d 日 (01..31)  
%H 小时 (00..23)  
%M 分钟 (00..59)  
%S 秒 (00..60)  
%s 自 1970-01-01 00:00:00 UTC 到现在的秒数
```

1. 使用date命令本体，无选项，直接查看时间

```
[itheima@bogon ~]$ date  
2022年 10月 08日 星期六 00:45:45 PDT
```

可以看到这个格式非常的不习惯。我们可以通过格式化字符串自定义显示格式

2. 按照2022-01-01的格式显示日期

```
[itheima@bogon ~]$ date +%Y-%m-%d  
2022-10-08
```

3. 按照2022-01-01 10:00:00的格式显示日期

```
[itheima@bogon ~]$ date +%Y-%m-%d %H:%M:%S  
date: 额外的操作数 "%H:%M:%S"  
Try 'date --help' for more information.  
[itheima@bogon ~]$ date "+%Y-%m-%d %H:%M:%S"  
2022-10-08 00:48:12
```

如上，由于中间带有空格，所以使用双引号包围格式化字符串，作为整体。

4.13 使用ln和date修改linux时区

细心的同学可能会发现，通过date查看的日期时间是不准确的，这是因为：系统默认时区非中国的东八区。

使用root权限，执行如下命令，修改时区为东八区时区

```
rm -f /etc/localtime  
sudo ln -s /usr/share/zoneinfo/Asia/Shanghai /etc/localtime
```

将系统自带的localtime文件删除，并将/usr/share/zoneinfo/Asia/Shanghai文件链接为localtime文件即可

4.14 ntp程序——自动校准系统时间

我们可以通过ntp程序自动校准系统时间

安装ntp： yum -y install ntp

启动并设置开机自启：

- systemctl start ntpd
- systemctl enable ntpd

当ntpd启动后会定期的帮助我们联网校准系统的时间

也可以手动校准（需root权限）：ntpdate -u ntp.aliyun.com

通过阿里云提供的服务网址配合ntpdate（安装ntp后会附带这个命令）命令自动校准

```
[root@bogon ~]# ntpdate -u ntp.aliyun.com
8 Oct 15:59:21 ntpdate[18492]: adjust time server 203.107.6.88 offset -0.009122 sec
```

4.15 ifconfig命令——查看ip

每一台联网的电脑都会有一个地址，用于和其它计算机进行通讯

IP地址主要有2个版本，V4版本和V6版本（V6很少用，课程暂不涉及）

IPv4版本的地址格式是：a.b.c.d，其中abcd表示0~255的数字，如192.168.88.101就是一个标准的IP地址

可以通过命令：ifconfig，查看本机的ip地址，如无法使用ifconfig命令，可以安装：yum -y install net-tools

特殊ip地址：

除了标准的IP地址以外，还有几个特殊的IP地址需要我们了解：

- 127.0.0.1，这个IP地址用于指代本机
- 0.0.0.0，特殊IP地址

可以用于指代本机

可以在端口绑定中用来确定绑定关系（后续讲解）

在一些IP地址限制中，表示所有IP的意思，如放行规则设置为0.0.0.0，表示允许任意IP访问

4.16 hostname命令——查看主机名

每一台电脑除了对外联络地址（IP地址）以外，也可以有一个名字，称之为“主机名”

无论是Windows或Linux系统，都可以给系统设置主机名

```
[root@bogon ~]# hostname
bogon
```

4.17 hostnamectl命令——修改主机名

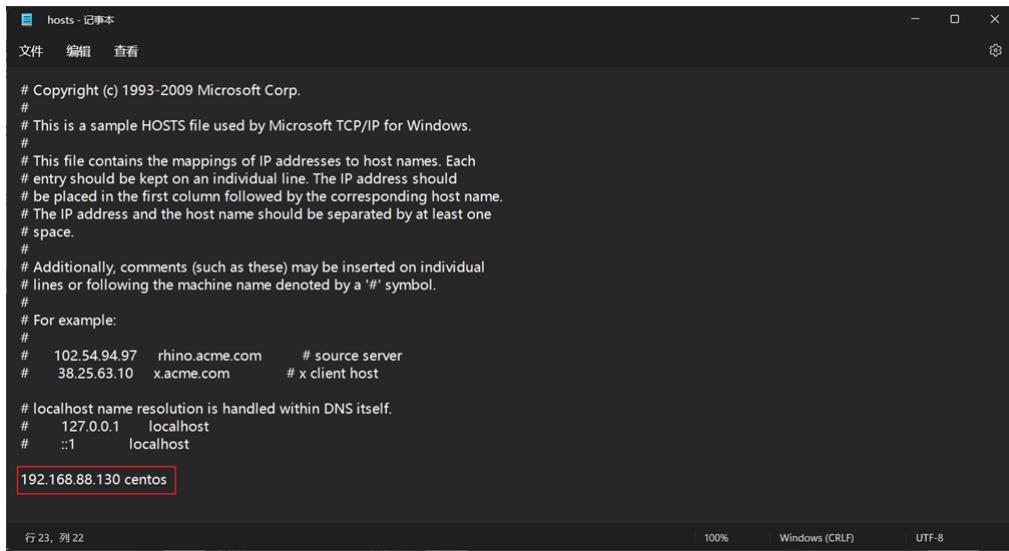
hostnamectl set-hostname 主机名

```
[root@bogon ~]# hostnamectl set-hostname centos
[root@bogon ~]# hostname
centos
```

4.18 finalshell通过主机名连接linux

比如，我们FinalShell是通过IP地址连接到的Linux服务器，那有没有可能通过域名（主机名）连接呢？

我们只需要在Windows系统的：C:\Windows\System32\drivers\etc\hosts文件中配置记录即可



```
# Copyright (c) 1993-2009 Microsoft Corp.  
#  
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.  
#  
# This file contains the mappings of IP addresses to host names. Each  
# entry should be kept on an individual line. The IP address should  
# be placed in the first column followed by the corresponding host name.  
# The IP address and the host name should be separated by at least one  
# space.  
#  
# Additionally, comments (such as these) may be inserted on individual  
# lines or following the machine name denoted by a '#' symbol.  
#  
# For example:  
#  
#   102.54.94.97  rhino.acme.com      # source server  
#   38.25.63.10   x.acme.com          # x client host  
  
# localhost name resolution is handled within DNS itself.  
#  
#   127.0.0.1    localhost  
#   ::1          localhost  
  
192.168.88.130 centos
```

4.19 固定ip地址

为什么需要固定 IP

当前我们虚拟机的Linux操作系统，其IP地址是通过DHCP服务获取的。

DHCP：动态获取IP地址，即每次重启设备后都会获取一次，可能导致IP地址频繁变更

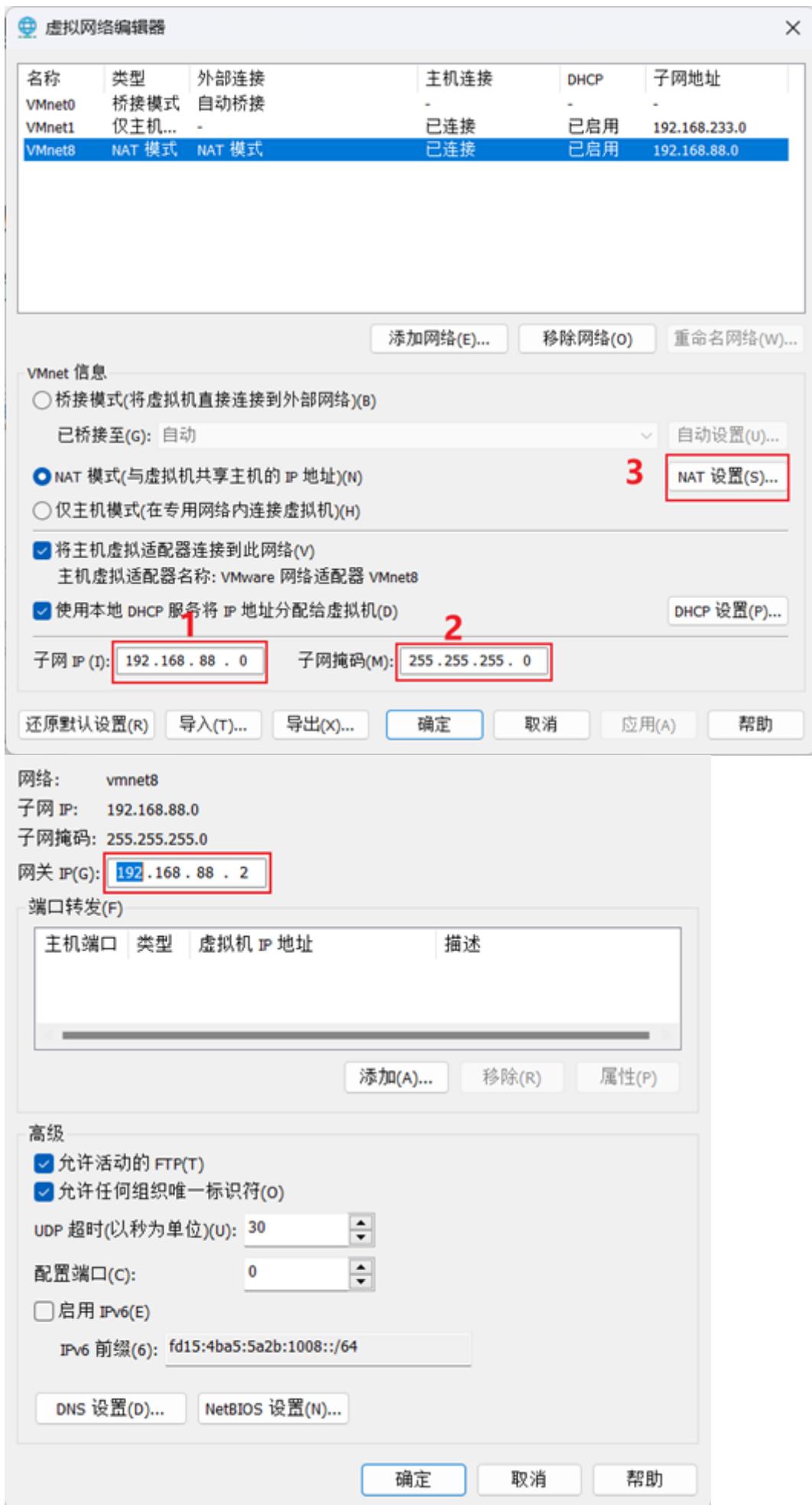
原因1：办公电脑IP地址变化无所谓，但是我们要远程连接到Linux系统，如果IP地址经常变化我们就要频繁修改适配很麻烦

原因2：在刚刚我们配置了虚拟机IP地址和主机名的映射，如果IP频繁更改，我们也需要频繁更新映射关系

综上所述，我们需要IP地址固定下来，不要变化了。

配置固定IP需要2个大步骤：

1. 在VMware Workstation（或Fusion）中配置IP地址网关和网段（IP地址的范围）



2. 在Linux系统中手动修改配置文件，固定IP

使用vim编辑/etc/sysconfig/network-scripts/ifcfg-ens33文件，填入如下内容

```

TYPE="Ethernet"
PROXY_METHOD="none"
BROWSER_ONLY="no"
BOOTPROTO="static" 由dhcp改为static
DEFROUTE="yes"
IPV4_FAILURE_FATAL="no"
IPV6INIT="yes"
IPV6_AUTOCONF="yes"
IPV6_DEFROUTE="yes"
IPV6_FAILURE_FATAL="no"
IPV6_ADDR_GEN_MODE="stable-privacy"
NAME="ens33"
UUID="f1ccb4a3-eecb-4f24-9cf0-97e6446f934b"
DEVICE="ens33"
ONBOOT="yes" 新增如下内容:
IPADDR="192.168.88.130" IP地址
NETMASK="255.255.255.0" 子网掩码固定: 255.255.255.0
GATEWAY="192.168.88.2" 网关和VMware虚拟网络编辑器中设置的一致
DNS1="192.168.88.2" DNS1设置为网关即可

```

执行: systemctl restart network 重启网卡, 执行ifconfig即可看到ip地址固定为192.168.88.130了

4.20 ping命令——检查指定的网络服务器是否是可联通状态

```
ping [-c num] ip或主机名
```

- 选项: -c, 检查的次数, 不使用-c选项, 将无限次数持续检查
- 参数: ip或主机名, 被检查的服务器的ip地址或主机名地址

示例:

- 检查到baidu.com是否联通

```
[itheima@bogon ~]$ ping baidu.com
PING baidu.com (39.156.66.10) 56(84) bytes of data.
64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=1 ttl=128 time=8.64 ms
64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=2 ttl=128 time=7.65 ms
64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=3 ttl=128 time=7.70 ms
```

结果表示联通, 延迟8ms左右

- 检查到39.156.66.10是否联通, 并检查3次

```
[itheima@bogon ~]$ ping -c 3 39.156.66.10
PING 39.156.66.10 (39.156.66.10) 56(84) bytes of data.
64 bytes from 39.156.66.10: icmp_seq=1 ttl=128 time=8.20 ms
64 bytes from 39.156.66.10: icmp_seq=2 ttl=128 time=6.59 ms
64 bytes from 39.156.66.10: icmp_seq=3 ttl=128 time=8.18 ms

--- 39.156.66.10 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 6.598/7.663/8.208/0.756 ms
```

4.21 wget命令——命令行内下载网络文件

wget是非交互式的文件下载器，可以在命令行内下载网络文件

```
wget [-b] url
```

- 选项：-b，可选，后台下载，会将日志写入到当前工作目录的wget-log文件
- 参数：url，下载链接

示例：

- 下载apache-hadoop 3.3.0版本：wget <http://archive.apache.org/dist/hadoop/common/hadoop-3.3.0/hadoop-3.3.0.tar.gz>

```
[itheima@bogon ~]$ wget http://archive.apache.org/dist/hadoop/common/hadoop-3.3.0/hadoop-3.3.0.tar.gz
--2022-10-08 17:50:01--  http://archive.apache.org/dist/hadoop/common/hadoop-3.3.0/hadoop-3.3.0.tar.gz
正在解析主机 archive.apache.org (archive.apache.org)... 138.201.131.134, 2a01:4f8:172:2ec5::2
正在连接 archive.apache.org (archive.apache.org)|138.201.131.134|:80... 已连接。
已发出 HTTP 请求，正在等待回应... 200 OK
长度: 500749234 (478M) [application/x-gzip]
正在保存至: "hadoop-3.3.0.tar.gz.1"

0% [                                         ] 221,865          230KB/s
```

- 在后台下载：wget -b <http://archive.apache.org/dist/hadoop/common/hadoop-3.3.0/hadoop-3.3.0.tar.gz>
- 通过tail命令可以监控后台下载进度：tail -f wget-log

注意：无论下载是否完成，都会生成要下载的文件，如果下载未完成，请及时清理未完成的不可用文件。

4.22 curl命令——发送http网络请求，可用于：下载文件、获取信息等

```
curl [-O] url
```

- 选项：-O，用于下载文件，当url是下载链接时，可以使用此选项保存文件
- 参数：url，要发起请求的网络地址

示例：

- 向cip.cc发起网络请求：curl cip.cc

```
[root@bogon vmware-tools-distrib]# curl cip.cc
IP      : 114.242.26.45
地址    : 中国 北京
运营商  : 联通

数据二  : 北京市 | 联通

数据三  : 中国北京北京市 | 联通

URL     : http://www.cip.cc/114.242.26.45
```

- 向python.itheima.com发起网络请求：curl python.itheima.com
- 通过curl下载hadoop-3.3.0安装包：curl -O <http://archive.apache.org/dist/hadoop/common/hadoop-3.3.0/hadoop-3.3.0.tar.gz>

4.23 nmap命令——查看端口占用

可以通过Linux命令去查看端口的占用情况

- 使用nmap命令，安装nmap： yum -y install nmap

```
nmap 被查看的ip
```

```
[root@bogon ~]# nmap 127.0.0.1

Starting Nmap 6.40 ( http://nmap.org ) at 2022-10-10 15:36 CST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0000030s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
111/tcp   open  rpcbind
631/tcp   open  ipp
6000/tcp  open  X11

Nmap done: 1 IP address (1 host up) scanned in 0.03 seconds
```

可以看到，本机（127.0.0.1）上有5个端口现在被程序占用了。

- 其中22端口，一般是SSH服务使用，即FinalShell远程连接Linux所使用的端口

4.24 netstat命令——查看指定端口的占用情况

安装netstat： yum -y install net-tools

```
netstat -anp | grep 端口号
```

```
[root@bogon ~]# netstat -anp | grep 6000
tcp        0      0 0.0.0.0:6000          0.0.0.0:*              LISTEN      7174/X
tcp6       0      0 ::1:6000             ::*:*                  LISTEN      7174/X
```

如图，可以看到当前系统6000端口被程序（进程号7174）占用了

其中，0.0.0.0:6000，表示端口绑定在0.0.0.0这个IP地址上，表示允许外部访问

```
[root@bogon ~]# netstat -anp | grep 12345
[root@bogon ~]#
```

可以看到，当前系统12345端口，无人使用哦。

4.25 ps命令——查看Linux系统中的进程信息

```
ps [-e -f]
```

- 选项：-e，显示出全部的进程
- 选项：-f，以完全格式化的形式展示信息（展示全部信息）

一般来说，固定用法就是： ps -ef 列出全部进程的全部信息

```
[itheima@centos ~]$ ps -ef
UID        PID  PPID  C STIME TTY          TIME CMD
root      1      0  0 11:56 ?        00:00:01 /usr/lib/systemd
root      2      0  0 11:56 ?        00:00:00 [kthreadd]
root      3      2  0 11:56 ?        00:00:03 [ksoftirqd/0]
root      5      2  0 11:56 ?        00:00:00 [kworker/0:0H]
root      7      2  0 11:56 ?        00:00:00 [migration/0]
root      8      2  0 11:56 ?        00:00:00 [rcu_bh]
root      9      2  0 11:56 ?        00:00:01 [rcu_sched]
root     10      2  0 11:56 ?        00:00:00 [lru-add-drain]
root     11      2  0 11:56 ?        00:00:00 [watchdog/0]
root     13      2  0 11:56 ?        00:00:00 [kdevtmpfs]
root     14      2  0 11:56 ?        00:00:00 [netns]
root     15      2  0 11:56 ?        00:00:00 [khungtaskd]
root     16      2  0 11:56 ?        00:00:00 [writeback]
root     17      2  0 11:56 ?        00:00:00 [kintegrityd]
```

从左到右分别是：

- **UID**: 进程所属的用户ID
- **PID**: 进程的进程号ID
- **PPID**: 进程的父ID (启动此进程的其它进程)
- **C**: 此进程的CPU占用率 (百分比)
- **STIME**: 进程的启动时间
- **TTY**: 启动此进程的终端序号, 如显示? 表示非终端启动
- **TIME**: 进程占用CPU的时间
- **CMD**: 进程对应的名称或启动路径或启动命令

查看指定进程：

- 在FinalShell中, 执行命令: tail, 可以看到, 此命令一直阻塞在那里
- 在FinalShell中, 复制一个标签页, 执行: ps -ef 找出tail这个程序的进程信息
- 问题: 是否会发现, 列出的信息太多, 无法准确的找到或很麻烦怎么办?

我们可以使用管道符配合grep来进行过滤, 如:

```
ps -ef | grep tail
```

即可准确的找到tail命令的信息

```
[itheima@centos ~]$ ps -ef|grep tail
itheima  46712  8021  0 17:14 pts/0    00:00:00 tail
itheima  47217  46758  0 17:14 pts/2    00:00:00 grep --color=auto tail
```

- 过滤不仅仅过滤名称, 进程号, 用户ID等等, 都可以被grep过滤哦
- 如: ps -ef | grep 30001, 过滤带有30001关键字的进程信息 (一般指代过滤30001进程号)

4.26 kill命令——关闭进程

在Windows系统中, 可以通过任务管理器选择进程后, 点击结束进程从而关闭它。

同样, 在Linux中, 可以通过kill命令关闭进程。

```
kill [-9] 进程ID
```

- 选项: -9, 表示强制关闭进程。不使用此选项会向进程发送信号要求其关闭, 但是否关闭看进程自身的处理机制。

4.27 top命令——查看CPU、内存使用情况

可以通过top命令查看CPU、内存使用情况, 类似Windows的任务管理器

默认每5秒刷新一次, 语法: 直接输入top即可, 按q或ctrl + c退出

选项	功能
-p	只显示某个进程的信息
-d	设置刷新时间，默认是5s
-c	显示产生进程的完整命令，默认是进程名
-n	指定刷新次数，比如 top -n 3，刷新输出3次后退出
-b	以非交互非全屏模式运行，以批次的方式执行top，一般配合-n指定输出几次统计信息，将输出重定向到指定文件，比如 top -b -n 3 > /tmp/top.tmp
-i	不显示任何闲置 (idle) 或无用 (zombie) 的进程
-u	查找特定用户启动的进程

按键	功能
h键	按下h键，会显示帮助画面
c键	按下c键，会显示产生进程的完整命令，等同于-c参数，再次按下c键，变为默认显示
f键	按下f键，可以选择需要展示的项目
M键	按下M键，根据驻留内存大小 (RES) 排序
P键	按下P键，根据CPU使用百分比大小进行排序
T键	按下T键，根据时间/累计时间进行排序
E键	按下E键，切换顶部内存显示单位
e键	按下e键，切换进程内存显示单位
l键	按下l键，切换显示平均负载和启动时间信息。
i键	按下i键，不显示闲置或无用的进程，等同于-i参数，再次按下，变为默认显示
t键	按下t键，切换显示CPU状态信息
m键	按下m键，切换显示内存信息

top - 14:36:57 up 3 min, 2 users, load average: 0.09, 0.23, 0.12 1分钟, 5分钟, 15分钟负载										
Tasks: 179 total, 2 running, 177 sleeping, 0 stopped, 0 zombie										
%Cpu(s): 0.7 us, 2.8 sy, 0.0 ni, 96.5 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st CPU										
KiB Mem : 995892 total, 196192 free, 387648 used, 412052 buff/cache 内存										
KiB Swap: 2098172 total, 2098172 free, 0 used. 399096 avail Mem										
PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+ COMMAND
6456	root	20	0	320028	6740	5268	S	0.7	0.7	0:00.21 vmtoolsd
8091	itheima	20	0	161164	2900	1212	S	0.7	0.3	0:00.07 sshd
8329	itheima	20	0	162012	2308	1596	R	0.3	0.2	0:00.01 top
1	root	20	0	193668	6844	4176	S	0.0	0.7	0:01.23 systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00 kthreadd
3	root	20	0	0	0	0	S	0.0	0.0	0:00.07 ksoftirqd/0
4	root	20	0	0	0	0	S	0.0	0.0	0:00.13 kworker/0:0
5	root	0	-20	0	0	0	S	0.0	0.0	0:00.00 kworker/0:0H
6	root	20	0	0	0	0	S	0.0	0.0	0:00.02 kworker/u256:0
7	root	rt	0	0	0	0	S	0.0	0.0	0:00.00 migration/0
8	root	20	0	0	0	0	S	0.0	0.0	0:00.00 rcu_bh
9	root	20	0	0	0	0	S	0.0	0.0	0:00.25 rcu_sched
10	root	0	-20	0	0	0	S	0.0	0.0	0:00.00 lru-add-drain

- 第一行：

```
top - 14:39:58 up 6 min, 2 users, load average: 0.06, 0.17, 0.13
```

top：命令名称，14:39:58：当前系统时间，up 6 min：启动了6分钟，2 users：2个用户登录，load：1、5、15分钟负载

- 第二行：

```
Tasks: 175 total, 1 running, 174 sleeping, 0 stopped, 0 zombie
```

Tasks: 175个进程, 1 running: 1个进程子在运行, 174 sleeping: 174个进程睡眠, 0个停止进程, 0个僵尸进程

- 第三行:

```
%Cpu(s): 0.3 us, 1.4 sy, 0.0 ni, 98.3 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
```

%Cpu(s): CPU使用率, us: 用户CPU使用率, sy: 系统CPU使用率, ni: 高优先级进程占用CPU时间百分比, id: 空闲CPU率, wa: IO等待CPU占用率, hi: CPU硬件中断率, si: CPU软件中断率, st: 强制等待占用CPU率

- 第四、五行:

```
KiB Mem : 995892 total, 187672 free, 394912 used, 413308 buff/cache  
KiB Swap: 2098172 total, 2098172 free, 0 used. 391852 avail Mem
```

KiB Mem: 物理内存, total: 总量, free: 空闲, used: 使用, buff/cache: buff和cache占用

KiBSwap: 虚拟内存 (交换空间), total: 总量, free: 空闲, used: 使用, buff/cache: buff和cache占用

- 其余内容

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
3	root	20	0	0	0	0	S	0.3	0.0	0:00.20	ksoftirqd/0
2891	root	20	0	0	0	0	S	0.3	0.0	0:00.30	xfsaield/sda3
6409	root	12	-8	84552	824	784	S	0.3	0.1	0:00.12	audispd
8091	itheima	20	0	161164	2940	1212	S	0.3	0.3	0:02.69	sshd

- PID: 进程id
- USER: 进程所属用户
- PR: 进程优先级, 越小越高
- NI: 负值表示高优先级, 正表示低优先级
- VIRT: 进程使用虚拟内存, 单位KB
- RES: 进程使用物理内存, 单位KB
- SHR: 进程使用共享内存, 单位KB
- S: 进程状态 (S休眠, R运行, Z僵死状态, N负数优先级, I空闲状态)
- %CPU: 进程占用CPU率
- %MEM: 进程占用内存率
- TIME+: 进程使用CPU时间总计, 单位10毫秒
- COMMAND: 进程的命令或名称或程序文件路径

4.28 df命令——磁盘信息监控

```
df [-h]
```

- 选项: -h, 以更加人性化的单位显示

文件系统	1K-块	已用	可用	已用%	挂载点
/dev/sda3	39517336	5720616	33796720	15%	/
devtmpfs	482152	0	482152	0%	/dev
tmpfs	497944	0	497944	0%	/dev/shm
tmpfs	497944	8512	489432	2%	/run
tmpfs	497944	0	497944	0%	/sys/fs/cgroup
/dev/sda1	303780	149528	154252	50%	/boot
tmpfs	99592	12	99580	1%	/run/user/42
tmpfs	99592	0	99592	0%	/run/user/1000

文件系统	容量	已用	可用	已用%	挂载点
/dev/sda3	38G	5.5G	33G	15%	/
devtmpfs	471M	0	471M	0%	/dev
tmpfs	487M	0	487M	0%	/dev/shm
tmpfs	487M	8.4M	478M	2%	/run
tmpfs	487M	0	487M	0%	/sys/fs/cgroup
/dev/sda1	297M	147M	151M	50%	/boot
tmpfs	98M	12K	98M	1%	/run/user/42
tmpfs	98M	0	98M	0%	/run/user/1000

4.29 iostat命令——查看CPU、磁盘的相关信息

```
iostat [-x] [num1] [num2]
```

- 选项: -x, 显示更多信息
- num1: 数字, 刷新间隔, num2: 数字, 刷新几次

[itheima@centos ~]\$ iostat						
Linux 3.10.0-957.el7.x86_64 (centos) 2022年10月12日 _x86_64_ (1 CPU)						
avg-cpu:	%user	%nice	%system	%iowait	%steal	%idle
	0.42	0.00	2.05	0.02	0.00	97.52

Device:	tps	kB_read/s	kB_wrtn/s	kB_read	kB_wrtn
sda	5.89	159.98	9.37	349348	20455

tps: 该设备每秒的传输次数 (Indicate the number of transfers per second that were issued to the device.) 。 "一次传输"意思是"一次I/O请求"。多个逻辑请求可能会被合并为"一次I/O请求"。 "一次传输"请求的大小是未知的。

使用iostat的-x选项, 可以显示更多信息

[itheima@centos ~]\$ iostat -x												
Linux 3.10.0-957.el7.x86_64 (centos) 2022年10月12日 _x86_64_ (1 CPU)												
avg-cpu:	%user	%nice	%system	%iowait	%steal	%idle	rrqm/s	wrqm/s	r/s	w/s	rkB/s	wkB/s
	0.40	0.00	1.98	0.01	0.00	97.60	0.00	0.07	3.50	1.30	128.29	8.05

```

rrqm/s: 每秒这个设备相关的读取请求有多少被Merge了（当系统调用需要读取数据的时候，VFS将请求发到各个FS，如果FS发现不同的读取请求读取的是相同Block的数据，FS会将这个请求合并Merge，提高IO利用率，避免重复调用）；
wrqm/s: 每秒这个设备相关的写入请求有多少被Merge了。
rsec/s: 每秒读取的扇区数：sectors
wsec/s: 每秒写入的扇区数。
rKB/s: 每秒发送到设备的读取请求数
wKB/s: 每秒发送到设备的写入请求数
avgrq-sz 平均请求扇区的小大
avgqu-sz 平均请求队列的长度。毫无疑问，队列长度越短越好。
await: 每一个IO请求的处理的平均时间（单位是微秒毫秒）。
svctm 表示平均每次设备I/O操作的服务时间（以毫秒为单位）
%util: 磁盘利用率

```

4.30 sar命令——网络状态监控

可以使用sar命令查看网络的相关统计（sar命令非常复杂，这里仅简单用于统计网络）

```
sar -n DEV num1 num2
```

- 选项：-n，查看网络，DEV表示查看网络接口
- num1：刷新间隔（不填就查看一次结束），num2：查看次数（不填无限次数）

```
[itheima@centos ~]$ sar -n DEV 3 2
Linux 3.10.0-957.el7.x86_64 (centos)      2022年10月12日 _x86_64_          (1 CPU)

15时 16分 04秒      IFACE    rxpck/s    txpck/s    rxkB/s    txkB/s    rxcmp/s    txcmp/s    rxmcst/s
15时 16分 07秒        lo      0.00      0.00      0.00      0.00      0.00      0.00      0.00
15时 16分 07秒  virbr0-nic    0.00      0.00      0.00      0.00      0.00      0.00      0.00
15时 16分 07秒  virbr0      0.00      0.00      0.00      0.00      0.00      0.00      0.00
15时 16分 07秒     ens33    13.85    29.05     1.18      6.58      0.00      0.00      0.00

15时 16分 07秒      IFACE    rxpck/s    txpck/s    rxkB/s    txkB/s    rxcmp/s    txcmp/s    rxmcst/s
15时 16分 10秒        lo      0.00      0.00      0.00      0.00      0.00      0.00      0.00
15时 16分 10秒  virbr0-nic    0.00      0.00      0.00      0.00      0.00      0.00      0.00
15时 16分 10秒  virbr0      0.00      0.00      0.00      0.00      0.00      0.00      0.00
15时 16分 10秒     ens33    11.11    21.89     0.96      6.22      0.00      0.00      0.00

平均时间：      IFACE    rxpck/s    txpck/s    rxkB/s    txkB/s    rxcmp/s    txcmp/s    rxmcst/s
平均时间：        lo      0.00      0.00      0.00      0.00      0.00      0.00      0.00
平均时间：  virbr0-nic    0.00      0.00      0.00      0.00      0.00      0.00      0.00
平均时间：  virbr0      0.00      0.00      0.00      0.00      0.00      0.00      0.00
平均时间：     ens33    12.48    25.46     1.07      6.40      0.00      0.00      0.00
```

信息解读：

- IFACE 本地网卡接口的名称**
- rxpck/s 每秒钟接受的数据包**
- txpck/s 每秒钟发送的数据包**
- rxKB/S 每秒钟接受的数据包大小，单位为KB**
- txKB/S 每秒钟发送的数据包大小，单位为KB**
- rxcmp/s 每秒钟接受的压缩数据包**
- txcmp/s 每秒钟发送的压缩包**
- rxmcst/s 每秒钟接收的多播数据包**

如图，查看2次，隔3秒刷新一次，并最终汇总平均记录

4.31 环境变量介绍

在讲解which命令的时候，我们知道使用的一系列命令其实本质上就是一个个的可执行程序。

比如，cd命令的本体就是：/usr/bin/cd 这个程序文件。

我们是否会有疑问，为何无论当前工作目录在哪里，都能执行：/usr/bin/cd这个程序呢？

这就是环境变量的作用啦。

环境变量是操作系统（Windows、Linux、Mac）在运行的时候，记录的一些关键性信息，用以辅助系统运行。

在Linux系统中执行：env命令即可查看当前系统中记录的环境变量

环境变量是一种KeyValue型结构，即名称和值，如下图：

```
[itheima@centos ~]$ env
XDG_SESSION_ID=1
HOSTNAME=centos
SELINUX_ROLE_REQUESTED=
TERM=xterm
SHELL=/bin/bash
HISTSIZE=1000
SSH_CLIENT=192.168.88.1 4779 22
SELINUX_USE_CURRENT_RANGE=
SSH_TTY=/dev/pts/0
USER=itheima
LS_COLORS=rs=0:di=0;34:ln=0;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33:01:cd=40;33:01:or=40;31:01:mi=01;05:37;41:su=3
01;31:*.tgz=01;31:*.arc=01;31:*.arj=01;31:*.taz=01;31:*.lha=01;31:*.lz4=01;31:*.lzh=01;31:*.lzma=01;31:*.tlz=01;31:*.txz=0
1;31:*.gz=01;31:*.lrz=01;31:*.lz=01;31:*.lzo=01;31:*.xz=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.tz=01;
sar=01;31:*.rar=01;31:*.alz=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.cab=01;31:*.jpg=01;35:*.jp
;35:*.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.svgz=01;35:*.mng=01;35:*.pcx=0
*:*.webm=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:*.mp4v=01;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.ASF=01;35
v=01;35:*.gl=01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf=01;35:*.axv=01;35:*.anx=01;35:*.ogv=0
*:*.midi=01;36:*.mka=01;36:*.mp3=01;36:*.mpc=01;36:*.ogg=01;36:*.ra=01;36:*.wav=01;36:*.axa=01;36:*.oga=01;36:*.spx=01;36:*
MAIL=/var/spool/mail/itheima
PATH=/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/itheima/.local/bin:/home/itheima/bin
PWD=/home/itheima
LANG=zh_CN.UTF-8
SELINUX_LEVEL_REQUESTED=
HISTCONTROL=ignoredups
SHLVL=1
HOME=/home/itheima
LOGNAME=itheima
XDG_DATA_DIRS=/home/itheima/.local/share/flatpak/exports/share:/var/lib/flatpak/exports/share:/usr/local/share:/usr/share
SSH_CONNECTION=192.168.88.1 4779 192.168.88.130 22
LESSOPEN=||/usr/bin/lesspipe.sh %
XDG_RUNTIME_DIR=/run/user/1000
_= /usr/bin/env
```

如左图，图中记录了：

- **HOME: /home/itheima, 用户的HOME路径**
- **USER: itheima, 当前的操作用户**
- **PWD: 当前工作路径**
-

等等一系列信息，用于辅助系统在运行的时候
从环境变量中获取关键信息

在前面提出的问题中，我们说无论当前工作目录是什么，都能执行/usr/bin/cd这个程序，这个就是借助环境变量中：PATH这个项目的值来做到的。

```
[itheima@centos ~]$ env | grep PATH
PATH=/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/itheima/.local/bin:/home/itheima/bin
```

PATH记录了系统执行任何命令的搜索路径，如上图记录了（路径之间以:隔开）：

- /usr/local/bin
- /usr/bin
- /usr/local/sbin
- /usr/sbin
- /home/itheima/.local/bin
- /home/itheima/bin

当执行任何命令，都会按照顺序，从上述路径中搜索要执行的程序的本体

比如执行cd命令，就从第二个目录/usr/bin中搜索到了cd命令，并执行

4.32 \$符号——环境变量操作

在Linux系统中，\$符号被用于取“变量”的值。

环境变量记录的信息，除了给操作系统自己使用外，如果我们想要取用，也可以使用。

取得环境变量的值就可以通过语法：\$环境变量名 来取得

比如：

```
echo $PATH
```

就可以取得PATH这个环境变量的值，并通过echo语句输出出来。

```
[itheima@centos ~]$ echo $PATH  
/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/itheima/.local/bin:/home/itheima/bin
```

又或者：

```
echo ${PATH}ABC
```

```
[itheima@centos ~]$ echo ${PATH}ABC  
/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/itheima/.local/bin:/home/itheima/binABC
```

当和其它内容混合在一起的时候，可以通过{}来标注取的变量是谁

Linux环境变量可以用户自行设置，其中分为：

- 临时设置，语法：export 变量名=变量值

```
[itheima@centos ~]$ export ITHEIMA=666  
[itheima@centos ~]$ echo $ITHEIMA  
666
```

临时环境变量

- 永久生效

针对当前用户生效，配置在当前用户的：~/.bashrc文件中

针对所有用户生效，配置在系统的：/etc/profile文件中

并通过语法：source 配置文件，进行立刻生效，或重新登录FinalShell生效

4.33 自定义环境变量PATH

环境变量PATH这个项目里面记录了系统执行命令的搜索路径。

这些搜索路径我们也可以自行添加到PATH中去。

修改PATH的值：

- 临时修改PATH:

```
export PATH=$PATH:/home/itheima/myenv
```

无论在哪里都能执行了

- 永久修改PATH:

将export PATH=\$PATH:/home/itheima/myenv, 填入用户环境变量文件或系统环境变量文件中去

4.34 rz、sz命令——finalshell快速下载、上传

我们可以通过FinalShell工具，方便的和虚拟机进行数据交换。

在FinalShell软件的下方窗体中，提供了Linux的文件系统视图，可以方便的：

- 浏览文件系统，找到合适的文件，右键点击下载，即可传输到本地电脑
- 浏览文件系统，找到合适的目录，将本地电脑的文件拖拽进入，即可方便的上传数据到Linux中

通过拖拽方式下载更快

当然，除了通过FinalShell的下方窗体进行文件的传输以外，也可以通过rz、sz命令进行文件传输。

注意，rz、sz命令需要终端软件支持才可正常运行

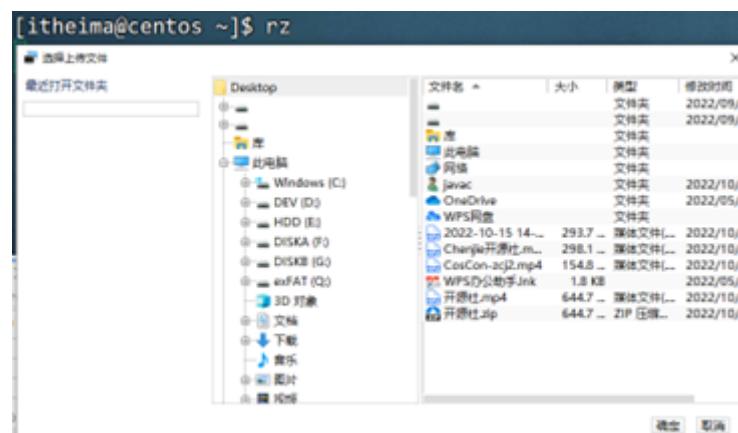
FinalShell、SecureCRT、XShell等常用终端软件均支持此操作

4.34.1 rz、sz命令安装

```
yum -y install lrzs
```

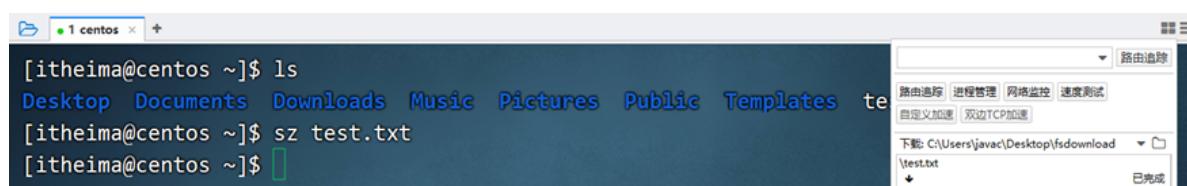
4.34.2 rz命令——上传

- 直接输入rz即可



4.34.3 sz命令——下载

- sz命令进行下载，语法：sz 要下载的文件



文件会自动下载到桌面的：fsdownload文件夹中。

4.35 tar、zip、unzip命令——文件压缩与解压

市面上有非常多的压缩格式

- zip格式：Linux、Windows、MacOS，常用
- 7zip：Windows系统常用
- rar：Windows系统常用
- tar：Linux、MacOS常用
- gzip：Linux、MacOS常用

在Windows系统中常用的软件如：winrar、bandizip等软件，都支持各类常见的压缩格式，这里不多做讨论。

我们现在要学习，如何在Linux系统中操作：tar、gzip、zip这三种压缩格式

完成文件的压缩、解压操作。

4.35.1 tar命令——压缩与解压

- .tar，称之为tarball，归档文件，即简单的将文件组装到一个.tar的文件内，并没有太多文件体积的减少，仅仅是简单的封装
- .gz，也常见为.tar.gz，gzip格式压缩文件，即使用gzip压缩算法将文件压缩到一个文件内，可以极大的减少压缩后的体积

```
tar [-c -v -x -f -z -C] 参数1 参数2 ..... 参数n
```

- -c，创建压缩文件，用于压缩模式
- -v，显示压缩、解压过程，用于查看进度
- -x，解压模式
- -f，要创建的文件，或要解压的文件，-f选项必须在所有选项中位置处于最后一个
- -z，gzip模式，不使用-z就是普通的tarball格式
- -C，选择解压的目的地，用于解压模式

注意：

- -z选项如果使用的话，一般处于选项位第一个
- -f选项，必须在选项位最后一个
- -C选项单独使用，和解压所需的其它参数分开

压缩常用命令：

```
tar -cvf test.tar 1.txt 2.txt 3.txt      #将1.txt 2.txt 3.txt 压缩到test.tar文件  
内
```

```
tar -zcvf test.tar.gz 1.txt 2.txt 3.txt    #将1.txt 2.txt 3.txt 压缩到test.tar.gz  
文件内，使用gzip模式
```

解压常用命令：

```
tar -xvf test.tar          #解压test.tar, 将文件解压至当前目录  
tar -xvf test.tar -C /home/itheima      #解压test.tar, 将文件解压至指定目录  
(/home/itheima)  
  
tar -zxvf test.tar.gz -C /home/itheima  #以Gzip模式解压test.tar.gz, 将文件解压至指定目  
录
```

4.35.2 zip命令——压缩成zip文件

可以使用zip命令, 压缩文件为zip压缩包

```
zip [-r] 参数1 参数2 ..... 参数n
```

- -r, 被压缩的包含文件夹的时候, 需要使用-r选项, 和rm、cp等命令的-r效果一致, 文件夹递归压
缩

常用命令:

```
zip test.zip a.txt b.txt c.txt      #将a.txt b.txt c.txt 压缩到test.zip文件内  
  
zip -r test.zip test itheima a.txt  #将test、itheima两个文件夹和a.txt文件, 压缩到  
test.zip文件内
```

4.35.3 unzip命令——zip文件解压缩

使用unzip命令, 可以方便的解压zip压缩包

```
unzip [-d] 参数
```

- -d, 指定要解压去的位置, 同tar的-C选项
- 参数, 被解压的zip压缩包文件

常用命令:

```
unzip test.zip          #将test.zip解压到当前目录  
unzip test.zip -d /home/itheima    #将test.zip解压到指定文件夹内 (/home/itheima)
```