Linux 运维必须掌握150个命令讲解

https://wangchujiang.com/linux-command/

apt-get linux 命令 在线中文手册 (51yip.com)

explainshell.com - find(1) - search for files in a directory hierarchy

1.线上查询及帮助命令（ 3 个）

man

```
man ls
```

help

Info

2.文件和目录操作命令（ 19 个）

ls

> 实例

```
$ ls        # 仅列出当前目录可见文件
$ ls -l     # 列出当前目录可见文件详细信息
$ ls -hl    # 列出详细信息并以可读大小显示文件大小
$ ls -al    # 列出所有文件（包括隐藏）的详细信息
$ ls --human-readable --size -1 -S --classify # 按文件大小排序
$ du -sh * | sort -h # 按文件大小排序(同上)
```

cd

> 实例

```
cd    # 进入用户主目录;
cd /  # 进入根目录
cd ~  # 进入用户主目录;
cd ..  # 返回上级目录（若当前目录为“/“，则执行完后还在“/"; ".."为上级目录的意思）;
cd ../..  # 返回上两级目录;
cd !$  # 把上个命令的参数作为cd参数使用。
cd -  # 切换到上一个工作目录的说明
cd ${OLDPWD}
# 命令会直接切换到上一个工作目录。
```

cp

```
cp -r dir destdir
```

find

> 实例

```
#语法
find(选项)(参数)
#选项
-amin<分钟>：查找在指定时间曾被存取过的文件或目录，单位以分钟计算；
-atime<24小时数>：查找在指定时间曾被存取过的文件或目录，单位以24小时计算；
-cmin<分钟>：查找在指定时间之时被更改过的文件或目录；
-ctime<24小时数>：查找在指定时间之时被更改的文件或目录，单位以24小时计算；
-depth：从指定目录下最深层的子目录开始查找；
-empty：寻找文件大小为0 Byte的文件，或目录下没有任何子目录或文件的空目录；
-exec<执行指令>：假设find指令的回传值为True，就执行该指令；
-maxdepth<目录层级>：设置最大目录层级；
-mmin<分钟>：查找在指定时间曾被更改过的文件或目录，单位以分钟计算；

-mtime<24小时数>：查找在指定时间曾被更改过的文件或目录，单位以24小时计算；

-perm<权限数值>：查找符合指定的权限数值的文件或目录；
-size<文件大小>：查找符合指定的文件大小的文件；
-type<文件类型>：只寻找符合指定的文件类型的文件；
```

```
# 当前目录搜索所有文件，文件内容 包含 “140.206.111.111” 的内容
find . -type f -name "*" | xargs grep "140.206.111.111"
find ./ |xargs egrep   "123"

#在/home目录下查找以.txt结尾的文件名
find /home -name "*.txt"
find . -maxdepth 3 -type f

#查看权限是644文件
find ./ -type f  -perm 644

#删除指定inode编号的文件(改名字等)
[root@180-143 test]# touch 'test test$@#$$'
[root@180-143 test]# ls -li
总用量 0
33580451 -rw-r--r-- 1 root root 0 2月  15 14:14 test test$@#$$
[root@180-143 test]# find ./ -inum 33580451
```

```
./test test$@#$$
[root@180-143 test]# find ./ -inum 33580451 | xargs rm -rf
```

```
UNIX/Linux文件系统每个文件都有三种时间戳：

访问时间 （-atime/天，-amin/分钟）：用户最近一次访问时间。
修改时间 （-mtime/天，-mmin/分钟）：文件最后一次修改时间。
变化时间 （-ctime/天，-cmin/分钟）：文件数据元（例如权限等）最后一次修改时间。
#查看7天以前的日志以log结尾
find . -type f -mtime -7 -name "*.log"
```

```
#统计代码行数
find . -name "*.java"|xargs cat|grep -v ^$|wc -l # 代码行数统计, 排除空行
```

mkdir

```
mkdir -p dir
```

mv

```
mv old new
```

rm

```
rm file
rm -rf file
```

touch

```
touch file
touch test_{1..100}
```

file

```
[root@localhost ~]# file install.log
install.log: UTF-8 Unicode text

[root@localhost ~]# file -b install.log      <== 不显示文件名称
UTF-8 Unicode text

[root@localhost ~]# file -i install.log      <== 显示MIME类别。
install.log: text/plain; charset=utf-8

[root@localhost ~]# file -b -i install.log
text/plain; charset=utf-8
```

tree

```
#列出目录/private/ 第一级文件名
tree  /private/ -L 1
/private/
├── etc
├── tftpboot
├── tmp
└── var
```

了解

basename dirname

chattr

> 用来改变文件属性

```
#用chattr命令防止系统中某个关键文件被修改：
chattr +i /etc/fstab
chattr -i /etc/fstab
#让某个文件只能往里面追加内容，不能删除，一些日志文件适用于这种操作：
chattr +a /data1/user_act.log
```

lsattr

> 查看文件的第二扩展文件系统属性

```
[root@180-143 test]# touch test
[root@180-143 test]# chattr +i test
[root@180-143 test]# lsattr test
----i----------- test
```

md5sum

```
#生成一个文件insert.sql的md5值:
[root@180-143 test]# md5sum insert.sql
d41d8cd98f00b204e9800998ecf8427e  insert.sql
```

3.查看文件及内容处理命令（ 19 个）

`vi` `vim`

vi命令 是UNIX操作系统和类UNIX操作系统中最通用的全屏幕纯文本编辑器。Linux中的vi编辑器叫vim，它是vi的增强版（vi Improved），与vi编辑器完全兼容，而且实现了很多增强功能。

vi编辑器支持编辑模式和命令模式，编辑模式下可以完成文本的编辑功能，命令模式下可以完成对文件的操作命令，要正确使用vi编辑器就必须熟练掌握着两种模式的切换。默认情况下，打开vi编辑器后自动进入命令模式。从编辑模式切换到命令模式使用“esc”键，从命令模式切换到编辑模式使用“A”、“a”、“O”、“o”、“I”、“i”键。

vi编辑器提供了丰富的内置命令，有些内置命令使用键盘组合键即可完成，有些内置命令则需要以冒号“：”开头输入。常用内置命令如下：

```
Esc：从编辑模式切换到命令模式；
ZZ：命令模式下保存当前文件所做的修改后退出vi；
:行号：光标跳转到指定行的行首；
:$：光标跳转到最后一行的行首；
x或X：删除一个字符，x删除光标后的，而X删除光标前的；
D：删除从当前光标到光标所在行尾的全部字符；
dd：删除光标行正行内容；
ndd：删除当前行及其后n-1行；
nyy：将当前行及其下n行的内容保存到寄存器？中，其中？为一个字母，n为一个数字；
p：粘贴文本操作，用于将缓存区的内容粘贴到当前光标所在位置的下方；
P：粘贴文本操作，用于将缓存区的内容粘贴到当前光标所在位置的上方；
```

/字符串：文本查找操作，用于从当前光标所在位置开始向文件尾部查找指定字符串的内容，查找的字符串会被加亮显示；
? 字符串：文本查找操作，用于从当前光标所在位置开始向文件头部查找指定字符串的内容，查找的字符串会被加亮显示；
a，bs/F/T：替换文本操作，用于在第a行到第b行之间，将F字符串换成T字符串。其中，“s/”表示进行替换操作；
a：在当前字符后添加文本；
A：在行末添加文本；
i：在当前字符前插入文本；
I：在行首插入文本；
o：在当前行后面插入一空行；
O：在当前行前面插入一空行；
gg：跳转到第一行
Shift+G： 跳转到最后一行
:wq：在命令模式下，执行存盘退出操作；
:w：在命令模式下，执行存盘操作；
:w! ：在命令模式下，执行强制存盘操作；
:q：在命令模式下，执行退出vi操作；
:q! ：在命令模式下，执行强制退出vi操作；
:e文件名：在命令模式下，打开并编辑指定名称的文件；
:n：在命令模式下，如果同时打开多个文件，则继续编辑下一个文件；
:f：在命令模式下，用于显示当前的文件名、光标所在行的行号以及显示比例；
:set number：在命令模式下，用于在最左端显示行号；
:set nonumber：在命令模式下，用于在最左端不显示行号；

cat

```
[root@180-143 test]# cat 1.txt
1
2
3

#多行文本追加
cat > 1.txt <<EOF
1
2
3
EOF

cat > xx.conf <<EOF
name=zz
age=18

EOF
```

more

head

显示文件的开头部分

```
# 查看历史文件的前6行：
[user2@pc ~]$ head -n 6 ~/.bash_history
#1575425555
cd ~
#1575425558
ls -lh
#1575425562
vi ~/Desktop/ZhuangZhu-74.txt
```

tail

在屏幕上显示指定文件的末尾若干行

```
tail file #（显示文件file的最后10行）
tail -n +20 file #（显示文件file的内容，从第20行至文件末尾）
tail -c 10 file #（显示文件file的最后10个字节）

tail -25 mail.log # 显示 mail.log 最后的 25 行
tail -f mail.log # 等同于--follow=descriptor，根据文件描述符进行追踪，当文件改名或被删除，追踪停止
tail -F mail.log # 等同于--follow=name --retry，根据文件名进行追踪，并保持重试，即该文件被删除或改名后，如果再次创建相同的文件名，会继续追踪
```

cut

> 实例

```
#语法
cut（选项）（参数）

#选项
-b：仅显示行中指定直接范围的内容；
-c：仅显示行中指定范围的字符；
-d：指定字段的分隔符，默认的字段分隔符为“TAB”；
-f：显示指定字段的内容；
-n：与“-b”选项连用，不分割多字节字符；
```

```
--complement：补足被选择的字节、字符或字段；
--out-delimiter= 字段分隔符：指定输出内容是的字段分割符；
--help：显示指令的帮助信息；
--version：显示指令的版本信息。
```

例如有一个学生报表信息，包含 No、Name、Mark、Percent：

```
[root@localhost text]# cat test.txt
No Name Mark Percent
01 tom 69 91
02 jack 71 87
03 alex 68 98
```

```
[root@localhost text]# cut -f 1 test.txt
No
01
02
03
[root@localhost text]# cut -f2,3 test.txt
Name Mark
tom 69
jack 71
alex 68
```

sort

```
[root@180-143 test]# netstat -ant | awk '{print $NF}' | sort  | uniq -c| sort -nrk 1
     74 ESTABLISHED
     25 LISTEN
      1 TIME_WAIT
      1 State
      1 established)
      1 CLOSE_WAIT

[root@180-143 test]# seq 100 | sort -nr | head -n 10
100
99
98
97
96
95
94
93
```

```
92
91
```

uniq

显示或忽略重复的行。

```
语法：
uniq [OPTION]... [INPUT [OUTPUT]]
-c, --count                 在每行开头增加重复次数。

[root@180-143 test]# cat test
1 2 3
1 2 3
[root@180-143 test]# uniq test
1 2 3
```

wc

统计文件的字节数、字数、行数

> 语法

```
wc(选项)(参数)
wc [选项]... [文件]...
wc [选项]... --files0-from=F
```

```
[root@180-143 test]# cat test
1 2 3
1 2 3
[root@180-143 test]# cat -n test
     1  1 2 3
     2  1 2 3
[root@180-143 test]# wc -l test
2 test
```

`grep` `egrep`

强大的文本搜索工具

grep （global search regular expression(RE) and print out the line，全面搜索正则表达式并把行打印出来）是一种强大的文本搜索工具，它能使用正则表达式搜索文本，并把匹配的行打印出来。用于过滤/搜索的特定字符。可使用正则表达式能配合多种命令使用，使用上十分灵活。

> 选项

```
-A <显示行数>   --after-context=<显示行数>   # 除了显示符合范本样式的那一行之外，并显示该行之后的内容。
-B<显示行数>   --before-context=<显示行数>   # 除了显示符合样式的那一行之外，并显示该行之前的内容。
-o # 只输出文件中匹配到的部分。
-n --line-number          # 在显示符合范本样式的那一列之前，标示出该列的编号。
-E --extended-regexp              # 将范本样式为延伸的普通表示法来使用，意味着使用能使用扩展正则表达式。
-i --ignore-case    # 忽略字符大小写的差别。
```

> 正则表达式

```
^    # 锚定行的开始 如：'^grep'匹配所有以grep开头的行。
$    # 锚定行的结束 如：'grep$' 匹配所有以grep结尾的行。
.    # 匹配一个非换行符的字符 如：'gr.p'匹配gr后接一个任意字符，然后是p。
*    # 匹配零个或多个先前字符 如：'*grep'匹配所有一个或多个空格后紧跟grep的行。
.*   # 一起用代表任意字符。
[]   # 匹配一个指定范围内的字符，如'[Gg]rep'匹配Grep和grep。
[^]  # 匹配一个不在指定范围内的字符，如：'[^A-Z]rep' 匹配不包含 A-Z 中的字母开头，紧跟 rep 的行
\(..\)  # 标记匹配字符，如'\(love\)'，love被标记为1。
\<      # 锚定单词的开始，如:'\<grep'匹配包含以grep开头的单词的行。
\>      # 锚定单词的结束，如'grep\>'匹配包含以grep结尾的单词的行。
x\{m\}  # 重复字符x，m次，如：'0\{5\}'匹配包含5个o的行。
x\{m,\}   # 重复字符x,至少m次，如：'o\{5,\}'匹配至少有5个o的行。
x\{m,n\}  # 重复字符x，至少m次，不多于n次，如：'o\{5,10\}'匹配5--10个o的行。
\w    # 匹配文字和数字字符，也就是[A-Za-z0-9]，如：'G\w*p'匹配以G后跟零个或多个文字或数字字符，然后是p。
\W    # \w的反置形式，匹配一个或多个非单词字符，如点号句号等。
\b    # 单词锁定符，如: '\bgrep\b'只匹配grep。
```

grep命令常见用法

在文件中搜索一个单词，命令会返回一个包含 **“match_pattern”** 的文本行：

```
grep match_pattern file_name
grep "match_pattern" file_name
```

在多个文件中查找：

```
grep "match_pattern" file_1 file_2 file_3 ...
```

输出除之外的所有行 **-v** 选项：

```
grep -v "match_pattern" file_name
```

标记匹配颜色 **--color=auto** 选项：

```
grep "match_pattern" file_name --color=auto
```

使用正则表达式 **-E** 选项：

```
grep -E "[1-9]+"
# 或
egrep "[1-9]+"
```

只输出文件中匹配到的部分 **-o** 选项：

```
echo this is a test line. | grep -o -E "[a-z]+\."
line.

echo this is a test line. | egrep -o "[a-z]+\."
line.
```

统计文件或者文本中包含匹配字符串的行数 **-c** 选项：

```
grep -c "text" file_name
```

搜索命令行历史记录中 输入过 `git` 命令的记录：

```
history | grep git
```

grep递归搜索文件

在多级目录中对文本进行递归搜索：

在多级目录中对文本进行递归搜索：

```
grep "text" . -r -n
# .表示当前目录。
```

忽略匹配样式中的字符大小写：

```
echo "hello world" | grep -i "HELLO"
# hello
```

`tr`

> 将字符进行替换压缩和删除

> 选项

```
-d或—delete：删除所有属于第一字符集的字符；
```

> 实例

将输入字符由大写转换为小写：

```
echo "HELLO WORLD" | tr 'A-Z' 'a-z'
hello world
```

'A-Z' 和 'a-z'都是集合，集合是可以自己制定的，例如：'ABD-}'、'bB.,'、'a-de-h'、'a-c0-9'都属于集合，集合里可以使用'\n'、'\t'，可以可以使用其他ASCII字符。

使用tr删除字符：

```
echo "hello 123 world 456" | tr -d '0-9'
hello  world
```

###

`vimdiff`

`dos2unix`

将DOS格式文本文件转换成Unix格式

> 实例

最简单的用法就是dos2unix直接跟上文件名：

```
dos2unix file
```

diff

4.文件压缩及解压缩命令（4个）

`tar`

unzip gzip zip

5.信息显示命令（ 12 个）

`uname` `hostname` `uptime` `stat` `du` `df` `top` `date` `free`

dmesg cal

6.搜索文件命令（ 4 个）

`which`

> 查找并显示给定命令的绝对路径

which命令 用于查找并显示给定命令的绝对路径，环境变量PATH中保存了查找命令时需要遍历的目录。which指令会在环境变量$PATH设置的目录里查找符合条件的文件。也就是说，使用which命令，就可以看到某个系统命令是否存在，以及执行的到底是哪一个位置的命令。

```
[root@localhost ~]# which pwd
/bin/pwd

[root@localhost ~]# which adduser
/usr/sbin/adduser
```

`whereis`

> 查找二进制程序、代码等相关文件路径

```
[root@180-143 test]# whereis pwd
pwd: /usr/bin/pwd /usr/include/pwd.h /usr/share/man/man1/pwd.1.gz /usr
```

locate

7.用户管理命令（ 10 个）

`useradd` `userdel` `passwd` `id` `su` `visudo` `sudo`

usermod groupadd chage

8.基础网络操作命令（ 10 个）

`ifconfig`

`netstat`

`ping` icmp arp 地址解析协议 内网通信基于arp协议进行广播

`route`

`telnet`

`nc`

`ssh`

`scp`

`wget`

ifup ifdown

9.深入网络操作命令（ 6 个）

`nslookup`

`dig`

```
[root@localhost ~]# cat /etc/resolv.conf
# Generated by NetworkManager
nameserver 223.5.5.5
nameserver 114.114.114.11
```

`traceroute`

`mtr`

`tcpdump`

`nmap`

`lsof`

10.有关磁盘与文件系统的命令（ 10 几个）

`mount` `umount`

`df` `du`

fdisk

resize2fs

fsck dd dumpe2fs dump parted
mkfs partprobe e2fsck mkswap swapon sync

11.关机和查看系统信息的命令（ 3 个）

shutdown halt init

12.系统管理相关命令（ 8 个）

`uptime` `top` `free` `vmstat` `mpstat` `iostat`

sar(sysstats) chkconfig

13.系统安全相关命令（ 10 个）

`chmod` `chown` `sudo`

chattr lsattr passwd

chgrp chage su umask

14.查看系统用户登陆信息的命令（ 7 个）

`whoami` `who` `w` `last`

lastlog users finger

15.其它（ 19 个）

`echo` `rpm` `yum` `` `date `` `clear` `history` `nohup`

`watch`

xargs

bc

expr

time

printf alias unalias
nc exec export unset type

16.系统性能监视高级命令(12 个)

内存:top free vmstat mpstat iostat sar
CPU:top vmstat mpstat iostat sar
I/O:vmstat mpstat iostat sar
进程:ipcs ipcrm lsof strace lstrace
负载:uptime

17.关机/重启/注销命令（ 7）

关机重启:shutdown init halt poweroff reboot
注销退出：logout exit ctl+d ——>快捷键(生产常用)

18.进程管理：（ 16 个）

kill,killall,pkill：杀掉进程

Runing 僵尸 不可中断D 终止模式 挂起

ps：查看进程

pstree：显示进程状态树

crontab：设置定时

bg：后台运行

fg：挂起程序

jobs：显示后台程序

pgrep：查找匹配条件的进程

strace：跟踪一个进程的系统调用
vmstat：报告虚拟内存统计信息

19.非常危险的系统命令（ 5 个）：

mv rm

fdisk parted dd

20.linux 系统四位剑客（ 3 个）

`grep egrep` `sed` `awk` `find`

grep（ egrep） sed awk