

linux面试常考命令

- 1、显示日期与时间的命令：date
- 2、显示日历的命令：cal
- 3、简单好用的计算器：bc
- 4、热键“命令补全或文件补齐”：Tab
- 5、热键“中断目前程序”：Ctrl+C
- 6、热键“键盘输入结束（End of File）相当与输入exit”：Ctrl+D
- 7、在线求助：man 【command】或 info 【command】，按q退出
- 8、数据同步写入磁盘：sync
- 9、关机命令：shutdown -h 【now | 10(分钟) | 20: 25】
- 10、重启命令：reboot
- 11、改变文件所属用户组：chgrp

12、改变文件所有者：chown [-R] 'groupname' 'dirname/filename'

```
chgrp -R usergroup /home/yang/a.txt
```

13、改变文件的属性：chmod

```
chmod [-R] 帐号名称 文件或目录
```

```
chmod [-R] 帐号名称: 组名 文件或目录
```

```
chmod yangyuqin:root /home/hadoop/bb.txt
```

14、改变权限：chmod

```
chmod [-R] xyz 文件或目录
```

u,g,o表示user,group,other这3种身份的权限，a表示all，即全部的身份。+加入，-除去，=设置。r读，w写，x执行，4读，2写，1执行。

```
chmod 777 /home/hadoop/a.txt
```

```
chmod u+rwx,g+rx,o+r /home/hadoop/a.txt
```

15、目录

1) . 代表此层目录

2) .. 代表上一层目录

3) - 代表前一个工作目录

4) ~ 代表“目前用户身份”所在的主文件夹

5) ~ account 代表account这个用户的主文件夹（account是个帐号名称）

16、切换目录：cd

```
cd 【相对路径或绝对路径】
```

17、显示目前所在的目录：pwd

pwd [-P]

-P 如果是连接文件，会不以连接文件的数据显示，而是显示正确的完整路径，即获取连接文件的真正路径。

18、新建新目录：mkdir

mkdir [-mp] 目录名称

-m : 配置文件的权限，mkdir -m 711 test1;

-p : 帮助你直接将所需要的目录（包含上层目录）递归创建起来，如果没有这个参数，只能一层一层建立目录,mkdir -p /home/a/b/c/d.

19、删除“空”的目录：rmdir

rmdir [-p] 目录名称

20、查看文件与目录：ls

ls [-aAdfFhilnrRSt] 目录名称

ls [--color={never,auto,always}] 目录名称

ls [--full-time] 目录名称

-a: 全部的文件，连同隐藏文件（开头为.的文件）一起列出来（常用）

-d: 仅列出目录本身，而不是列出目录内的文件数据（常用）

-f: 直接列出结果，而不进行排序(ls默认会以文件名排序)

-F: 根据文件、目录等信息给予附加数据结构，例如：*: 代表可执行文件，/: 代表目录，=: 代表socket文件，|: 代表FIFO文件

-h: 将文件容量以人类较易读的方式(例如GB，KB等)列出来

-l: 列出长数据串，包含文件的属性与权限等数据（常用）

-r: 将排序结果反向输出，例如：原本文件名由小到大，反向则为由大到小

-R: 连同子目录内容一起列出来，等于该目录下的所有文件都会显示出来

-S: 以文件容量大小排序，而不是用文件名排序

-t: 依时间排序，而不是用文件名

--color=never: 不要依据文件特性给予颜色显示

--color=always: 显示颜色

- color=auto: 让系统自行依据设置来判断是否给予颜色
- full-time: 以完整时间模式 (包含年、月、日、时、分) 输出
- time={atime, ctime}: 输出访问时间或改变权限属性时间 (ctime) , 而非内容更改时间 (modification time)

21、复制文件或目录: cp

- cp [-adfilprsu] 源文件(source) 目标文件(destination)
- cp [option] source1,source2,source3, ... directory
- a: 相当于-pdr 的意思
- d: 若源文件为连接文件的属性(link file), 则复制连接文件属性而非文件本身
- p: 连同文件的属性一起复制过去, 而非使用默认属性(备份常用)
- r: 递归持续复制, 用于目录的复制行为(常用)
- i: 若目标文件(destination)已经存在时, 在覆盖时会先询问操作的进行(常用)
- l: 进行硬链接 (hard link) 的连接文件创建, 而非复制文件本身
- s: 复制成为符号链接文件(symbolic link), 即"快捷方式"文件
- u: 若destination 比source旧才更新destination

注意: 如果源文件有两个以上, 则最后一个目的文件一定要是“目录”才行

22、移除文件或目录: rm

- rm [-fir] 文件或目录
- f: 就是force的意思, 忽略不存在的文件, 不会出现警告信息
- i: 互动模式, 在删除前会询问用户是否操作
- r: 递归删除。最常用在目录的删除。但是很危险!!!

23、移动文件或目录, 或更名: mv

- mv [-fiu] source destination
- mv [options] source1,source2,source3,.....,directory

24、直接查看文件内容: cat (concatenate)

- cat [-AbEnTv]
- A: 相当于-vET的整合参数, 可列出一些特殊字符, 而不是空白而已
- n: 打印出行号, 连同空白行也会有行号, 与-b的参数不同
- b: 列出行号, 仅针对非空白行做行号显示, 空白行不标行号
- E: 将结尾的断行字符\$显示出来
- T: 将[Tab]按键以^I显示出来
- v: 列出一些看不出来的特殊字符

25、反向列示: tac

26、添加行号打印: nl

27、可翻页查看(只向后): more, 按q停止

28、可翻页查看(可向前或向后): less, 按q停止

29、取出前面几行: head

head [-n number] 文件

head -n 20 /home/test

head -n -100 /home/test //不打印后面100行的数据, 只是显示前面的数据

30、取出后面几行: tail

tail [-n number] 文件

tail -n 20 /home/test

tail -n +100 /home/test //只是列出100行以后的数据

31、文件时间:

1) modification time(mtime): 当该文件的“内容数据”更改时, 就会更新这个时间。内容数据指的是文件的内容, 而不是文件的属性或权限。

2) status time(ctime): 当该文件的“状态”(status)改变时, 就会更新这个时间, 举例来说, 像是权限与属性被更改了, 都会更新这个时间。

3) access time(ctime): 当“该文件的内容被取用”时, 就会更新这个读取时间(access)。举例来说, 我们使用cat去读取文件的时候, 就会更新该文件的ctime。

32、修改文件时间或创建新文件: touch

33、查看文件类型: file

34、寻找“执行文件”: which

which [-a] command

-a: 将所有由PATH目录中可以找到的命令均列出, 而不只第一个被找到的命令名称

which ipconfig

which hadoop

35、从whereis数据库文件中寻找特定的文件: whereis

whereis [-bmsu] 文件或目录

-b: 只找二进制格式的文件

-m: 只找在说明文件manual路径下的文件

-s: 只找source源文件
-u: 查找不在上述三个选项当中的其他特殊文件
whereis ifconfig

36、从以创建的数据库/var/lib/mlocate/数据库中寻找特定的文件: locate

locate [-ir] keyword

-i: 忽略大小写的差异

-r: 后面可接正则表达式的显示方式

locate passwd //从系统中找出与passwd相关的文件名

37、从文件系统寻找特定的文件: find

find [PATH] [option] [action]

参数:

1) 与时间有关的参数: 共有-atime、-ctime、-mtime, 以-mtime说明

-mtime n: n为数字, 意义为n天之前的“1天之内”被更改过的文件

-mtime +n: 列出在n天之前 (不含n天本身) 被更改过的文件名

-mtime -n: 列出在n天之内 (含n天本身) 被更改过的文件名

-newer file: file为一个存在的文件, 列出比file还要新的文件名

find / -mtime 0 //将过去系统上24小时内有改动(mtime)的文件列出, 数字0代表目前的时间, 所以是从现在开始到24小时前

find / -mtime 3 //列出3天前的24小时内有改动的文件

find /etc -newer /etc/passwd //寻租/etc目录下的文件, 如果文件日期比/etc/passwd新的就列出来, 可用于分辨两个文件之间的新旧关系

2) 与用户或用户组名有关的参数:

-uid n: n为数字, 这个数字是用户的帐号ID, 即UID, 这个UID 是记录在/etc/passwd里面与帐号名称对应的数字

-gid n: n为数字, 这个数字是用户组名的ID, 即GID, 这个GID 记录在/etc/group中

-user name: name为用户帐号名称, 例如dmtsai

-group name: name为用户组名, 例如users

-nouser: 寻找文件的所有者不存在/etc/passwd的人

-nogroup: 寻找文件的所有用户组不存在与/etc/group的人

find /home -user vbird //查找/home下面属于bird用户的文件, 可将属于某个用户的所有文件都找出来

find / -nouser //查找系统中不属于任何人的文件

3) 与文件权限及名称有关的参数:

-name filename: 查找文件名为filename的文件

-size [+|-]SIZE: 查找比SIZE 还要大(+)或小(-)的文件。这个SIZE的规格有: c: 代表byte, k: 代表1024bytes。所以, 要找比50KB还要大的文件, 就是“-size +50k”

-type TYPE: 查找文件的类型为TYPE的, 类型主要有: 一般正规文件(f)、设备文件(b, c)、目录(d)、连接文件(l)、socket(s)及FIFO(p)等属性

-perm mode: 查找文件权限“刚好等于”mode的文件, 这个mode为类似chmod的属性值, 举例来说, -rwxr-xr-x的属性为0755

-perm -mode: 查找文件权限“必须要全部包含mode的权限”的文件, 举例来说, 我们要查找-rwxr--r--, 即0744的文件, 使用-perm -0744, 当一个文件的权限为-rwxr-xr-x, 即755时, 也会被列出来, 因为-rwxr-xr-x的属性已经包括了-rwxr--r--的属性了

-perm +mode: 查找文件权限“包含任一mode的权限”的文件, 举例来说, 我们查找-rwxr-xr-x, 即-perm +0755, 但一个文件是-rw-----也会被列出来, 因为它有-rw.....的属性存在

find / -name passwd //找出文件名为passwd的这个文件

find /var -type s //找出/var目录下文件类型为socket的文件名

find / -perm +0755 //查找文件当中含有-rwxr-xr-x中的任一属性

4) 其他可进行的操作:

-exec command: command为其他命令, -exec后面可再接其他的命令来处理查找到的结果

-print: 将结果打印到屏幕上, 这个操作是默认值

find /etc -name passwd -exec cat {} \; //将找到文件名为passwd的文件作为命令的输入

38、列出文件系统的整体磁盘使用量: df

df [-ahikHTm] [目录或文件名]

-a: 列出所有的文件系统, 包括系统特有的/proc等文件系统;

-k: 以KB的容量显示各文件系统

-m: 以MB的容量显示各文件系统

-h: 以人们较易阅读的GB、MB、KB等格式自行显示

-H: 以M=1000K 替代M=1024K 的进位方式

-T: 连同该分区的文件系统名称(例如ext3)也列出

-i: 不用硬盘容量, 而已inode的数量来显示

df //将系统内所有的文件系统列出来, 在Linux下如果df没有加任何参数, 那么默认会将系统内所有的(不含特殊内存内的文件系统与swap)都以1KB的容量列出来。

df -h //将容量结果以易读的容量格式显示出来

df -aT //将系统内所有的特殊文件格式及名称都列出来

df -h /etc //将/etc下面的可用的磁盘容量以易读的容量格式显示, 这样就可以知道某个目录下还有多少容量可以使用了

df -ih //将目前各个分区当中可用的inode数量列出来

39、评估文件系统的磁盘使用量（常用于评估目录所占容量）： du

du [-ahskm] 文件或目录名称

-a: 列出所有的文件与目录容量, 因为默认仅统计目录下面的文件量而已

-h: 以人们较易读的容量格式(G/M)显示

-s: 列出容量而已, 而不列出每个各别的目录占用容量

-S: 不包含子目录下的总计, 与-s有点差别

-k: 以KB列出容量显示

-m: 以MB列出容量显示

du //列出当前目录下的所有文件容量, 但仅会显示目录容量(不含文件)

du -a //除了显示目录容量, 还会显示文件容量

du -sm .* //检查当前目录下面每个目录所占用的容量, 可检查某个目录下那个子目录占用最大的容量

40、连接文件：ln

1) 硬连接或实际连接：hard link


```
ln /etc/crontab .
```

2) 符号连接, 也即是快捷方式: symbolic link

```
ln -s /etc/crontab1 crontab2
```

41、变量的显示: echo

```
echo $PATH
```

```
echo helloworld
```

42、查看环境变量与常见环境变量说明: env

43、查看所有变量 (含环境变量与自定义变量): set

44、查看历史命令: history

45、提示符的设置: PS1

46、显示结果的语系变量: locale

47、命令别名设置: alias

```
alias lm='ls -l | more'
```

48、管道命令: pipe

```
ls -al /etc | less
```

49、将同一行里面的数据进行分解, 取出想要的数: cut

cut -d'分隔字符' -f fields //用于分隔字符

cut -c 字符范围 //用于排列整齐的信息

-d: 后面接分隔字符, 与-f一起使用

-f: 依据-d的分隔字符将一段信息切割成为数据, 与-f取出第几段的意思

-c: 以字符(characters)的单位取出固定字符区间

echo \$PATH | cut -d ':' -f 5 //将PATH变量取出, 以:为分割符, 取出第五个路径

echo \$PATH | cut -d ':' -f 3,5 //将PATH变量取出, 以:为分割符, 取出第3到第5个路径

export | cut -c 12- //将export输出的信息取得第12字符以后的所有字符串, 但是在处理多空格相连的数据时会比较复杂

50、分析一行信息, 若匹配想找的数据则将该行取出来: grep

grep [-acinv] [--color=auto] '查找字符串' filename

-a: 将binary文件以text文件的方法查找数据

-c: 计算找到“查找字符串”的次数

-i: 忽略大小写的不同, 所以大小写视为相同

-n: 顺便输出行号

-v: 顺便输出行号

--color=auto: 可以将找到的关键字部分加上颜色显示

history | grep 'ls'

51、排序命令: sort

sort [-fbMnrtuk] [file or stdin]

-f: 忽略大小写的差异, 例如A与a视为编码相同

-b: 忽略最前面的空格符部分

-M: 以月份的名字来排序, 例如JAN,DEC等的排序方法

-n: 使用“纯数字”进行排序 (默认是以文字类型来排序的)

-r: 反向排序

-u: 就是uniq, 相同的数据中, 仅出现一行代表

-t: 分隔符, 默认是用[Tab]键来分隔

-k: 以那个区间 (field) 来进行排序的意思

```
cat /etc/passwd | sort
```

```
cat /etc/passwd | sort -t ':' -k 3
```

52、排序完成但是将重复的数据仅列出一个显示: uniq

```
uniq [-ic]
```

-i: 忽略大小写字符的不同

-c: 进行计数

```
last | cut -d ' ' -f 1 | sort | uniq -c
```

53、统计文件的字、行、字符: wc

```
wc [-lwm]
```

-l: 仅列出行

-w: 仅列出多少字 (英文单字)

-m: 多少字符

```
cat /etc/man.config | wc //输出三个数字分别为行、字数、字符数
```

54、数据的替换、删除、新增、选取工具: sed

```
sed [-nefr] [动作]
```

1) 参数:

-n: 使用安静 (silent) 模式。在一般sed的用法中, 所有来自STDIN的数据一般都会被列出到屏幕上。但如果加上-n参数后, 则只有经过sed特殊处理的那一行 (或者操作) 才会被列出来

- e: 直接在命令模式上进行sed的动作编辑
- f: 直接将sed的动作写在一个文件内, -f filename则可以在filename内的sed动作
- r: sed的动作支持的是扩展型正则表达式的语法 (默认是基础正则表达式语法)
- i: 直接修改读取的文件内容, 而不是由屏幕输出

2) 动作说明: [n1 [,n2]] function

n1,n2: 不见得会存在, 一般代表选择进行动作的行数, 举例来说, 如果我的动作是需要10到20行之间进行的, 则“10, 20[动作行为]”

function常用的参数:

- a: 新增, a的后面可以接字符串, 而这些字符串会在新的一行出现(目前的下一行)
- c: 整行替换, c的后面可以接字符串, 这些字符串可以替换n1,n2之间的行
- d: 删除, 因为是删除, 所以d后面通常不接任何参数
- i: 插入, i的后面可以接字符串, 而这些字符串会在新的一行出现 (目前上一行)
- p: 打印, 也就是将某个选择的数据打印出来, 通常p会与参数sed -n一起运行
- s: 部分替换, 可以直接进行替换的工作, 通常这个s的动作可以搭配正则表达式! 例如1,20s/old/new/g

3) 以行为单位的新增/删除功能

nl /etc/passwd | sed '2,5d' //将/etc/passwd的内容列出并且打印行号, 同时, 将第2 ~ 5行删除

4) 在第二行后面 (即加在第3行) 加上“drink tea?” 字样

nl /etc/passwd | sed '2a drink tea?'

5) 在第二行前面 (即加在第2行) 插入“drink tea?” 字样

nl /etc/passwd | sed '2i drink tea?'

6) 在第二行后面加入两行字, 例如"Drink tea or " 与"drink beer? "

-> nl /etc/passwd | sed '2a Drink tea or \

-> drink beer? '

7) 以行为单位的替换与显示功能

nl /etc/passwd | sed '2,5c No 2-5 number' //将第2-5行的内容替换成为“No 2-5 number”

8) 仅打印部分数据

`nl /etc/passwd | sed -n '5,7p'` // -n表示只会打印处理后的数据，如果不加-n，那处理后的数据会打印两次

9) 部分数据的查找并替换的功能

`sed 's/要被替换的字符串/新的字符串/g'`

`hadoop@hadoop-ThinkPad:~$ /sbin/ifconfig eth0`

`eth0 Link encap:Ethernet HWaddr b8:88:e3:3c:89:f9`

`inet addr:10.130.26.21 Bcast:10.130.26.127 Mask:255.255.255.128`

`inet6 addr: fe80::ba88:e3ff:fe3c:89f9/64 Scope:Link`

`UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1`

`RX packets:143999 errors:0 dropped:1 overruns:0 frame:0`

`TX packets:88285 errors:0 dropped:0 overruns:0 carrier:0`

`collisions:0 txqueuelen:1000`

`RX bytes:39104286 (39.1 MB) TX bytes:39602666 (39.6 MB)`

`sbin/ifconfig eth0 | grep 'inet addr' | sed 's/^\.*addr://g' | sed 's/Bcast.*$//g'` //从ifconfig文件中抽取出本机的ip地址

`cat /etc/man.config | grep 'MAN' | sed 's/#.*$//g' | sed '/^$/d'`

55、数据处理工具：awk

sed常常作用于一整行的处理，awk则比较倾向于将一行分成数个“字段”来处理，awk相当适合处理小型的数据处理。

`awk '条件类型1{动作1} 条件类型2{动作2} ... ' filename`

awk 主要是处理每一行的字段内的数据，而默认的字段的分隔符为空格键或[tab]键

awk的处理流程是：

- 1) 读入第一行，并将第一行的数据填入\$0,\$1,\$3等变量当中，\$0代表一整行数据
- 2) 依据条件类型的限制，判断是否需要进行后面的动作
- 3) 做完所有的动作与条件类型
- 4) 所还有后续的“行”的数据，则重复上面1~3的步骤，直到所有的数据都读完为止

awk的内置变量：

NF : 每一行 (\$0) 拥有的字段总数

NR : 目前awk所处理的是“第几行”数据

FS : 目前的分隔字符，默认是空格键

awk运算符：

>: 大于 <: 小于 >=: 大于或等于 <=: 小于或等于 ==: 等于 !=: 不等于

last -n 5 | awk '{print \$1 "\t" \$3}' //打印第一列和第3列

last -n 5 | awk '{print \$1 "\t lines: " NR "\t colums:" NF}' //打印第一列，第几行，每一行的列数

cat /etc/passwd | awk '{FS=":"} \$3 < 10 {print \$1 "\t" \$3}' //分隔为:, 打印第三列小于10以下的数据，并且仅列出帐号与第三列

新建一个文件: pay.txt

```
Name  1st  2nd  3th
VBird 23000 24000 25000
DMTsai 21000 20000 23000
Bird2 43000 42000 41000
```

第一行只是说明，所以第一行不要进行加总 (NR==1时处理)

第二行以后就会有加总的情况出现 (NR》=2以后处理)

hadoop@hadoop-ThinkPad:~\$ cat pay.txt | awk 'NR==1{printf "%10s %10s %10s %10s %10s\n",\$1,\$2,\$3,\$4,"Total"}

NR>=2{total=\$2+\$3+\$4

printf "%10s %10d %10d %10d %10.2f\n",\$1,\$2,\$3,\$4,total}'

```
Name    1st    2nd    3th    Total
VBird   23000   24000   25000  72000.00
DMTsai   21000   20000   23000  64000.00
Bird2    43000   42000   41000 126000.00
```

56、文件比较工具: diff

diff [-bBi] from-filr to-file

57、新增用户：useradd

useradd [-u UID] [-g 初始化用户组] [-G 次要用户组] [-mM] [-c 说明栏] [-d 主文件夹绝对路径] [-s shell] 用户帐号名

useradd vbird1

useradd -u 123 -g users vbird2

58、设置密码：passwd

passwd 123456

59、删除用户：userdel

userdel vbird1

60、新增用户组：groupadd

groupadd [-g gid] [-r] 用户组名

61、删除用户组：groupdel

groupdel [groupname]

62、切换身份：su

su [-lm] [-c 命令] [username]

-：单纯使用 - 如“su -”，代表使用login-shell的变量文件读取方式来登录系统，若用户名称没有加上去，则代表切换为root身份

-l：与 - 类似，但后面需要加欲切换的用户帐号，也是login-shell的方式

-m：-m与-p是一样的，表示使用目前的环境设置，而不读取新用户的配置文件

-c：仅进行一次命令

63、切换到root身份: sudo

64、查看当前用户的id: id

65、查看目前登录在系统上的用户: who或w

66、每个帐号的最近登录时间: lastlog

67、直接将命令丢到后台中“执行”: &

```
tar -zvf /tmp/etc.tar.gz /etc &
```

68、将目前的工作丢到后台中“暂停’: Ctrl+z

69、查看目前的后台工作状态: jobs

```
jobs [-lrs]
```

-l: 除了列出job number与命令串之外, 同时列出PID的号码

-r: 仅列出正在后台run的工作

-s: 仅列出正在后台当中暂停 (stop) 的工作

70、管理后台当中的工作: kill

```
kill -signal %jobnumber
```

kill -l // -l列出目前kill能够使用的信号(signal)有哪些, signal代表给予后面接的那个工作什么样的指示。

-1: 重新读取一次参数的配置文件 (reload)

-2: 代表与由键盘输入[ctrl]-c同样的操作;

-9: 立即强制删除一个工作

-15: 以正常的程序方式终止一项工作, 与-9不同

kill -9 %2; jobs

71、查看某个时间点的进行状况: ps

ps aux //查看系统所有的进程数据

ps -lA //也是能够查看所有系统的数据

ps axjf //连同部分进行树状态

-A: 所有的进程均显示出来, 与-e具有相同的作用

-a: 不与termial有关的所有进程

-u: 有效用户相关的进程

x: 通常与a这个参数一起使用, 可列出较完整的信息

输出格式:

l: 较长、较详细地将该PID的信息列出

j: 工作的格式 (jobs format)

-f: 做一个更为完整的输出

1) 仅查看自己的bash相关进程: ps -l

2) 查看系统所有进程: ps aux

ps aux | grep 'syslog'

72、查看内存使用情况: free

free [-b|-k|-m|-g] [-t]

-b: 直接输入free时, 显示的是KB, 我们可以使用b (bytes)、m (MB)、k (KB) 及g (GB) 来显示单位

-t: 在输出的最终结果中显示物理内存与swap的总量

73、查看系统与内核相关信息：uname

```
uname [-asrmpi]
```

-a：所有系统相关的信息

74、查看系统启动时间与工作负载：uptime

75、跟踪网络：netstat

76、查看网卡信息：ifconfig

```
ifconfig eth0 192.168.0.1 netmask 255.255.255.0 //配置网卡ip地址
```

77、远程登录linux主机：ssh

```
ssh [-l login_name] [-p port] [user@]hostname
```

```
ssh 192.168.0.11
```

```
ssh -l root 192.168.0.11 或 ssh root@192.168.0.11 //指定用户
```

```
ssh -l root -p 12333 216.230.230.114
```

78、远程拷贝：scp

```
scp -r /home/administrator/test/ root@192.168.1.100:/root/ //拷贝本机/home/administrator/test整个目录至远程主机192.168.1.100的/root目录下
```

```
scp /home/administrator/Desktop/old/driver/test/test.txt root@192.168.1.100:/root/ //拷贝单个文件至远程主机
```

```
scp -r root@192.168.62.10:/root/ /home/administrator/Desktop/new/ //远程文件/文件夹下载
```

79、查看服务：service

```
service --status-all
```

