

Linux常见指令

1.ls指令

语法：ls [选项][目录或文件]

功能：对于目录，该命令列出该目录下的所有子目录与文件。对于文件，将列出文件名以及其他信息。

常用选项：

- a 列出目录下的所有文件，包括以 . 开头的隐含文件。
- d 将目录象文件一样显示，而不是显示其下的文件。如：ls -d 指定目录
- i 输出文件的 i 节点的索引信息。如 ls -ai 指定文件
- k 以 k 字节的形式表示文件的大小。ls -alk 指定文件
- l 列出文件的详细信息。
- n 用数字的 UID,GID 代替名称。（介绍 UID， GID）
- F 在每个文件名后附上一个字符以说明该文件的类型，"*"表示可执行的普通文件；"/"表示目录；"@ "表示符号链接；|"表示FIFOs；"="表示套接字(sockets)。（目录类型识别）
- r 对目录反向排序。
- t 以时间排序。
- s 在l文件名后输出该文件的大小。（大小排序，如何找到目录下最大的文件）
- R 列出所有子目录下的文件。(递归)
- 1 一行只输出一个文件。

举例：

| ls -l或者ll

```
total 120
drwxrwxr-x 3 junmoxiao junmoxiao 4096 May 28 16:01 a1
-rwxrwxr-x 1 junmoxiao junmoxiao 12456 May 21 18:30 a.out
drwxrwxr-x 3 junmoxiao junmoxiao 4096 Apr 29 09:53 b1
-rw-rw-r-- 1 junmoxiao junmoxiao 53 May 28 19:50 code.c
-rw-rw-r-- 1 junmoxiao junmoxiao 0 Apr 29 00:50 c.txt
drwxrwxr-x 3 junmoxiao junmoxiao 4096 May 28 14:55 d1
-rw-rw-r-- 1 junmoxiao junmoxiao 247 May 28 15:57 d1.zip
-rw-rw-r-- 1 junmoxiao junmoxiao 95 May 20 17:25 hello.cpp
-rw-rw-r-- 1 junmoxiao junmoxiao 5477 May 28 11:59 hellolinux.c
-rw-rw-r-- 1 junmoxiao junmoxiao 1 May 20 17:14 hello.py
```

编辑

| ls

```
a1      b1      c.txt   d1.zip   hellolinux.c
a.out   code.c   d1      hello.cpp hello.py
```

编辑

2.pwd指令

语法：pwd

功能：显示用户当前所在目录

常用选项：无

举例：

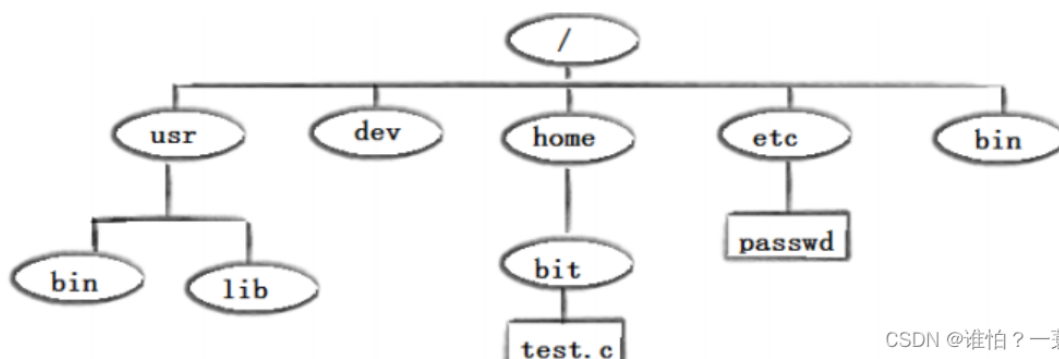
| pwd

```
/home/junmoxiao CSDN @谁怕？—蓑烟雨任平生
```

编辑

3.cd指令

Linux系统中，磁盘上文件和目录被组成一颗目录树，每个节点都是目录或文件。



CSDN @谁怕？一蓑烟雨任平生

编辑

语法：cd [目录名]

功能：改变工作目录。将当前目录改变到指定的目录下。

举例：

cd .. : 返回上级目录

cd /home/litao/linux/ : 绝对路径

cd ../day02/ : 相对路径

cd ~ : 进入用户家目录

cd - : 返回最近访问目录

4.touch命令

语法：touch [选项]... [文件]...

功能：touch命令参数可更改文档或目录的日期时间，包括时间和更改时间，或者新建一个不存在的文件。

常用选项：

- a 或--time=atime或--time=access或--time=use只更改存取时间。
- c 或--no-create 不建立任何文档。
- d 使用指定的日期时间，而非现在的时间。
- f 此参数将忽略不予处理，仅负责解决BSD版本touch指令的兼容性问题。
- m 或--time=mtime或--time=modify 只更改变动时间。
- r 把指定文档或目录的日期时间，统统设成和参考文档或目录的日期时间相同。
- t 使用指定的日期时间，而非现在的时间。

举例：

```
touch test.txt
```

5.mkdir指令

语法：mkdir [选项] [dirname]...

功能：在当前目录下创建一个名为 “dirname”的目录

常用选项：

- p, --parents 可以是一个路径名称。此时若路径中的某些目录尚不存在,加上此选项后,系统将自动建立好那些尚不存在的目录,即一次可以建立多个目录;

举例：

```
mkdir -p test/test1：递归建立多个目录
```

6.rmdir指令 && rm指令：

rmdir是一个与mkdir相对应的命令。mkdir是建立目录，而rmdir是删除命令。

语法：rmdir [-p][dirName]

适用对象：具有当前目录操作权限的所有使用者

功能：删除空目录

常用选项：

-p 当子目录被删除后如果父目录也变成空目录的话，就连带父目录一起删除。

rm命令可以同时删除文件或目录

语法：rm [-f-i--r-v][dirName/dir]

适用对象：所有使用者

功能：删除文件或目录

常用选项：

-f 即使文件属性为只读(即写保护)，亦直接删除

-i 删除前逐一询问确认

-r 删除目录及其下所有文件

举例：

rm -r a1

rm -r -i b1

```
[junmoxiao@VM-24-6-centos ~]$ rm -r -i b1
rm: descend into directory 'b1'
```

编辑

7.man指令

Linux的命令有很多参数,我们不可能全记住，我们可以通过查看联机手册获取帮助。

访问Linux手册页的命令是：**man语法：**man [选项]命令

如果Linux下没有man，则进行安装：

root下安装

```
| yum install -y man-pages
```

普通用户下安装：

```
| sudo yum install -y man-pages
```

常用选项：

-k 根据关键字搜索联机帮助

num 只在第num章节找

-a 将所有章节的都显示出来，比如 man printf 它缺省从第一章开始搜索，知道就停止，用a选项，当按下q退出，他会继续往后面搜索，直到所有章节都搜索完毕。

解释一下,man手册分为8章

1 是普通的命令

2 是系统调用,如open,write之类的(通过这个，至少可以很方便的查到调用这个函数，需要加什么头文件)

3 是库函数,如printf,fread4是特殊文件,也就是/dev下的各种设备文件

5 是指文件的格式,比如passwd, 就会说明这个文件中各个字段的含义

6 是给游戏留的,由各个游戏自己定义

7 是附件还有一些变量,比如向environ这种全局变量在这里就有说明

8 是系统管理用的命令，这些命令只能由root使用;例如ifconfig

举例：

man string

```
STRING(3)                                     Linux Programmer's Manual                                     STRING(3)

NAME
  strcpy, strcasecmp, strcat, strchr, strcmp, strcoll, strcpy, strcspn, strdup, strfry, strlen, strncat, strncmp, strncpy, strncasecmp, strpbrk, strchr,
  strsep, strspn, strstr, strtok, strxfrm, index, rindex - string operations

SYNOPSIS
  #include <strings.h>

  int strcasecmp(const char *s1, const char *s2);

  int strncasecmp(const char *s1, const char *s2, size_t n);

  char *index(const char *s, int c);

  char *rindex(const char *s, int c);

  #include <string.h>

  char *strcpy(char *dest, const char *src);

  char *strcat(char *dest, const char *src);

  char *strchr(const char *s, int c);

  int strcmp(const char *s1, const char *s2);

  int strcoll(const char *s1, const char *s2);

  char *strcpy(char *dest, const char *src);

  size_t strcspn(const char *s, const char *reject);

  char *strdup(const char *s);

  char *strfry(char *string);

Manual page string(3) line 1 (press h for help or q to quit) CSDN @维帕?—冀翔雨在平生
```

编辑

8.cp指令

语法：cp [选项] 源文件或目录 目标文件或目录

功能：复制文件或目录

说明：cp指令用于复制文件或目录，如同时指定两个以上的文件或目录，且最后的目的地是一个已经存在的目录，则它会把前面指定的所有文件或目录复制到此目录中。若同时指定多个文件或目录，而最后的目的地并非一个已存在的目录，则会出现错误信息。

常用选项：

- f 或 --force 强行复制文件或目录，不论目的文件或目录是否已经存在
- i 或 --interactive 覆盖文件之前先询问用户
- r 递归处理，将指定目录下的文件与子目录一并处理。若源文件或目录的形态，不属于目录或符号链接，则一律视为普通文件处理

| -R 或 --recursive递归处理，将指定目录下的文件及子目录一并处理

举例：

cp 为拷贝的操作，cp 指令默认拷贝的文件

| cp 文件名 路径：拷贝文件

```
[junmoxiao@VM-24-6-centos ~]$ tree cp1
cp1
|-- cp2
|   |-- cp3
|   `-- hello.py
`-- hello.cpp
```

CSDN @谁怕？—蓑烟雨任平生

编辑

| cp hello.cpp cp2

```
[junmoxiao@VM-24-6-centos ~]$ tree cp1
cp1
|-- cp2
|   |-- cp3
|   |-- hello.cpp
|   `-- hello.py
`-- hello.cpp
```

CSDN @谁怕？—蓑烟雨任平生

编辑

| cp -r 目录名 路径：r 为递归，递归拷贝文件，即拷贝目录


```
[junmoxiao@VM-24-6-centos ~]$ ll testcp
total 0
```

CSDN @谁怕？一蓑烟雨任平生

编辑

```
cp -r cp1 testcp
```

```
[junmoxiao@VM-24-6-centos ~]$ ll testcp
total 4
drwxrwxr-x 3 junmoxiao junmoxiao 4096 May 29 14:46 cp1
[junmoxiao@VM-24-6-centos ~]$ tree testcp
testcp
|-- cp1
|   |-- cp2
|   |   |-- cp3
|   |   |-- hello.cpp
|   |   `-- hello.py
|   `-- hello.cpp
```

CSDN @谁怕？一蓑烟雨任平生

编辑

`cp -f` 文件名 路径：强制拷贝。平常状况下，路径下有同名目录会给出覆盖提示；`-f` 为暴力，为不给是否覆盖提示，直接覆盖。

但是有的情况下哦可能不会强制覆盖，那可能是因为`cp`的把问题，可以用`alias`查看`cp`的别名

```
alias cp
```

```
alias cp='cp -i'
```

这里的 `cp -i` 我们上面见过，它的意思是覆盖文件之前先询问用户，所以 `cp` 本质是 `cp -i`，无论怎样都会先提问用户，比如我们的 `cp -f` 其实就被转换为了 `cp -i -f`，这里我们可以通过 **取消别名** 的方式，来解决这个问题：

| unalias cp

9.mv指令

mv命令是move的缩写，可以用来移动文件或者将文件改名（move (rename) files），是Linux系统下常用的命令，经常用来备份文件或者目录。

语法: mv [选项] 源文件或目录 目标文件或目录

功能：

- 1.视mv命令中第二个参数类型的不同（是目标文件还是目标目录），mv命令将文件重命名或将其移至一个新的目录中。
- 2.当第二个参数类型是文件时，mv命令完成文件重命名，此时，源文件只能有一个（也可以是源目录名），它将所给的源文件或目录重命名为给定的目标文件名。
- 3.当第二个参数是已存在的目录名称时，源文件或目录参数可以有多个，mv命令将各参数指定的源文件均移至目标目录中。

-f：force强制的意思，如果目标文件已经存在，不会询问而直接覆盖

- i：若目标文件（destination）已经存在时，就会询问是否覆盖！

举例：

| mv 文件名 路径：剪切文件到指定路径

```
[junmoxiao@VM-24-6-centos ~]$ tree mv1
mv1
|-- cp1
|   |-- cp2
|   |   |-- cp3
|   |   |-- hello.cpp
|   |   `-- hello.py
|   `-- hello.cpp
|-- mv2
|   |-- mv3
```

5 directories, 3 files

CSDN @谁怕？—蓑烟雨任平生

编辑

| mv cp1 mv2

```
[junmoxiao@VM-24-6-centos ~]$ tree mv1
mv1
|-- mv2
|   |-- cp1
|   |   |-- cp2
|   |   |   |-- cp3
|   |   |   |-- hello.cpp
|   |   |   `-- hello.py
|   |   `-- hello.cpp
|   |-- mv3
```

CSDN @谁怕？—蓑烟雨任平生

编辑

| mv 目录名 路径：剪切目录到指定路径

| mv 文件名 路径 新文件名：剪切文件到指定路径并重命名

总结：

- mv 文件名 不存在文件名：重命名
- mv 文件名 路径：剪切
- mv 文件名 目录 + 不存在文件名：剪切并重命名文件
- mv 目录名 目录 + 不存在目录名：剪切并重命名目录

10 .cat命令

语法：cat [选项] [文件]

功能：查看目标文件的内容

常用选项：

- b 对非空输出行编号
- n 对输出的所有行编号
- s 不输出多行空行

举例：

编辑 11.more指令

```
[junmoxiao@VM-24-6-centos ~]$ cat hellolinux.c
#include<stdio.h>
int main()
{

    printf("hello linux\n");
    printf("hello linux\n");
    printf("hello linux\n");
    printf("hello linux\n");
    printf("hello linux\n");
    printf("hello linux\n");
}
```

CSDN @谁怕？一蓑烟雨任平生

语法：more [选项] [文件]

功能：more命令，功能类似cat

常用选项：

-n 对输出的所有行编号

q 退出more

测试 more 指令前，我们先写一个命令行脚本：

```
cnt=0; while [ $cnt -le 1000 ]; do echo "hello $cnt"; let  
cnt++; done >log.txt
```

意思是往 mylog.txt 中写入 hello 1 ~ hello 1000。

使用 cat 查看一下：

```
hello 967  
hello 968  
hello 969  
hello 970  
hello 971  
hello 972  
hello 973  
hello 974  
hello 975  
hello 976  
hello 977  
hello 978  
hello 979  
hello 980  
hello 981  
hello 982  
hello 983  
OSDN@谁怕？一蓑烟雨任平生
```

编辑

cat 只看到了结尾部分，说明 cat 适用于观看短文本。

长文本可以使用more指令观看：

| more 文件名：从头部查看文本内容，按回车逐行翻阅。

12.less指令

语法：less[参数] 文件

功能：less与more类似，但使用less可以随意浏览文件，而more仅能向前移动，却不能向后移动，而且less在查看之前不会加载整个文件。

选项：

| -i 忽略搜索时的大小写
-N 显示每行的行号
/字符串：向下搜索“字符串”的功能
?字符串：向上搜索“字符串”的功能
n：重复前一个搜索（与 / 或 ? 有关）
N：反向重复前一个搜索（与 / 或 ? 有关）
q:quit

补充：

less工具是对文件或其它输出进行分页显示的工具，应该说是linux查看文件内容的工具，功能强大。

less的用法比起 more 更加的有弹性。在 more 的时候，我们并没有办法向前面翻，只能往后面看。

使用 less，就可以使用 [pageup] [pagedown] 等按键的功能来往前往后翻看文件，更容易用来查看一个文件的内容。

less 拥有更多的搜索功能，不止可以向下搜，也可以向上搜。

| less 文件名：查看文件内容支持上下翻阅

13.head指令

head 与 **tail** 就像它的名字一样的浅显易懂，它是用来显示开头或结尾某个数量的文字区块，**head** 用来显示档案的 开头至标准输出中，而 **tail** 想当然尔就是看档案的结尾。

语法：head [参数] ... [文件]...

功能：head用来显示档案开头至标准输出中，默认head命令打印其相应的文件的开头10行。

选项：

└ -n<行数> 显示的行数

编辑

14.tail指令

tail 命令从指定点开始将文件写到标准输出.使用tail命令的-f选项可以方便的查阅正在改变的日志文件,tail - f filename会把filename里最尾部的内容显示在屏幕上,并且不但刷新,使你看到最新的文件内容。

语法：tail[必要参数][选择参数][文件]

功能：用于显示指定文件末尾内容，不指定文件时，作为输入信息进行处理。常用查看日志文件。

选项：

└ -f 循环读取

└ -n<行数> 显示行数

在学习完head和tail指令后，我们怎样提取文件中的第m行到n行呢？

方法一：借助借助中间文件 + 重定向

我们以linux.c文件为例子

```

1: linux.c+
1 #include<stdio.h>
2 int main()
3 {
4     printf("hello linux\n");
5     printf("hello linux\n");
6     printf("hello linux\n");
7     printf("hello linux\n");
8     printf("hello linux\n");
9     printf("hello linux\n");
10    printf("hello linux\n");
11    printf("hello linux\n");
12    printf("hello linux\n");
13    printf("hello linux\n");
14    printf("hello linux\n");
15    printf("hello linux\n");
16    printf("hello linux\n");
17    printf("hello linux\n");
18    printf("hello linux\n");
19    printf("hello linux\n");
20    printf("hello linux\n");
21    printf("hello linux\n");
22    printf("hello linux\n");
23    printf("hello linux\n");
24    printf("hello linux\n");
25    printf("hello linux\n");
26    printf("hello linux\n");
27    printf("hello linux\n");
28    printf("hello linux\n");
29    printf("hello linux\n");
30    printf("hello linux\n");
31    printf("hello linux\n");
32    printf("hello linux\n");
33    printf("hello linux\n");
34    printf("hello linux\n");
35    printf("hello linux\n");

```

CSDN @谁怕？—蓑烟雨任平生

编辑

提取第10行到30行

step 1 : head -30 linux.c > tmp.c —— 将前30内容重定向到 tmp.c 中

step2 : tail -20 tmp.c —— tail 截取 tmp.txt 的后20行


```
[junmoxiao@VM-24-6-centos ~]$ tail -20 tmp.c
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
```

CSDN @谁怕？一蓑烟雨任平生

编辑

方法2：管道

首先，介绍一下管道。我们生活中，管道是用来传输资源的，对于Linux中的管道也是。

| 就是管道的标志。计算机中最重要的资源是数据，| 就会被 shell 解释为链接两条指令用来传输数据的指令。我们可以把管道也当做一种文件。

先看管道的例子：

我们原本统计行需要用 `wc -l mylog.txt`，并且会打印文件名：

```
wc -l mylog.txt
1001 mylog.txt
```

但是学习了管道后，我们可以使用 `cat mylog.txt | wc -l`，把 `cat` 的数据写到管道里，之后用 `wc -l` 到管道中读取，这样也不显示文件名。

```
cat mylog.txt | wc -l
```

```
1001
```

好了，介绍完管道和用法，我们回到正题：

`cat tmp.c` 将内容放到管道中，`head -520` 截取文件中前520行，再把这些放入管道中，`tail -20` 截取这些的后20行

指令为：

```
cat linux.c | head -520 | tail -20
```

```
[junmoxiao@VM-24-6-centos ~]$ cat linux.c | head -520 | tail -20
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
return 0;
}
```

CSDN @谁怕？一蓑烟雨任平生

编辑

15.时间相关指令

date显示

date 指定格式显示时间：date +%Y:%m:%d

date 用法：date [OPTION]... [+FORMAT]

1.在显示方面，使用者可以设定欲显示的格式，格式设定为一个加号后接数个标记，其中常用的标记列表如下：

- %H：小时(00..23)
- %M：分钟(00..59)
- %S：秒(00..61)
- %X：相当于 %H:%M:%S
- %d：日 (01..31)
- %m：月份 (01..12)
- %Y：完整年份 (0000..9999)
- %F：相当于 %Y-%m-%d

2.在设定时间方面

date -s //设置当前时间，只有root权限才能设置，其他只能查看。

date -s 20080523 //设置成20080523，这样会把具体时间设置成空00:00:00

date -s 01:01:01 //设置具体时间，不会对日期做更改

date -s "01:01:01 2008-05-23" //这样可以设置全部时间

date -s "01:01:01 20080523" //这样可以设置全部时间

date -s "2008-05-23 01:01:01" //这样可以设置全部时间

date -s "20080523 01:01:01" //这样可以设置全部时间

3.时间戳

时间→时间戳：date +%s

时间戳→时间：date -d@1508749502

Unix时间戳（英文为Unix epoch, Unix time, POSIX time 或 Unix timestamp）是从1970年1月1日（UTC/GMT 的午夜）开始所经过的秒数，不考虑闰秒。

举例：

| date

```
[junmoxiao@VM-24-6-centos ~]$ date
Mon May 29 21:27:21 CST 2023
```

编辑

| date +%Y-%m:%d/%H:%M:%S：特殊格式显示时间

```
[junmoxiao@VM-24-6-centos ~]$ date +%Y-%m:%d/%H:%M:%S
2023-05:29/21:29:01
```

编辑

| date +%s：显示时间戳时间

```
[junmoxiao@VM-24-6-centos ~]$ date +%s
1685366988
```

编辑

| date +%Y-%m:%d/%H:%M:%S -d@时间戳：将时间戳转换为格式时间

```
[junmoxiao@VM-24-6-centos ~]$ date +%Y-%m:%d/%H:%M:%S -d@1685366988
2023-05:29:21:29:48
```

CSDN @谁怕？一蓑烟雨任平生

编辑

16. cal指令：

cal命令可以用来显示公历（阳历）日历。公历是现在国际通用的历法，又称格列历，通称阳历。“阳历”又名“太阳历”，系以地球绕行太阳一周为一年，为西方各国所通用，故又名“西历”。

命令格式：cal [参数][月份][年份]

功能：用于查看日历等时间信息，如只有一个参数，则表示年份(1-9999)，如有两个参数，则表示**常用选项：**

- 3 显示系统前一个月，当前月，下一个月的月历
- j 显示在当年中的第几天（一年日期按天算，从1月1号算起，默认显示当前月在一年中的天数）
- y 显示当前年份的日历

举例：

cal ：默认查看本月日历

```
[junmoxiao@VM-24-6-centos ~]$ cal
      May 2023
Su Mo Tu We Th Fr Sa
    1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31
```

CSDN @谁怕？一蓑烟雨任平生

编辑

| cal 年份：查看一年的日历

```
[junmoxiao@VM-24-6-centos ~]$ cal 2023
2023

    January                February                March
Su Mo Tu We Th Fr Sa    Su Mo Tu We Th Fr Sa    Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7      1  2  3  4          1  2  3  4
 8  9 10 11 12 13 14      5  6  7  8  9 10 11      5  6  7  8  9 10 11
15 16 17 18 19 20 21      12 13 14 15 16 17 18      12 13 14 15 16 17 18
22 23 24 25 26 27 28      19 20 21 22 23 24 25      19 20 21 22 23 24 25
29 30 31                  26 27 28                  26 27 28 29 30 31

    April                  May                  June
Su Mo Tu We Th Fr Sa    Su Mo Tu We Th Fr Sa    Su Mo Tu We Th Fr Sa
                   1      1  2  3  4  5  6          1  2  3
 2  3  4  5  6  7  8      7  8  9 10 11 12 13      4  5  6  7  8  9 10
 9 10 11 12 13 14 15      14 15 16 17 18 19 20      11 12 13 14 15 16 17
16 17 18 19 20 21 22      21 22 23 24 25 26 27      18 19 20 21 22 23 24
23 24 25 26 27 28 29      28 29 30 31          25 26 27 28 29 30
30

    July                  August                September
Su Mo Tu We Th Fr Sa    Su Mo Tu We Th Fr Sa    Su Mo Tu We Th Fr Sa
                   1      1  2  3  4  5          1  2
 2  3  4  5  6  7  8      6  7  8  9 10 11 12      3  4  5  6  7  8  9
 9 10 11 12 13 14 15      13 14 15 16 17 18 19      10 11 12 13 14 15 16
16 17 18 19 20 21 22      20 21 22 23 24 25 26      17 18 19 20 21 22 23
23 24 25 26 27 28 29      27 28 29 30 31          24 25 26 27 28 29 30
30 31

    October                November                December
Su Mo Tu We Th Fr Sa    Su Mo Tu We Th Fr Sa    Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7      1  2  3  4          1  2
 8  9 10 11 12 13 14      5  6  7  8  9 10 11      3  4  5  6  7  8  9
15 16 17 18 19 20 21      12 13 14 15 16 17 18      10 11 12 13 14 15 16
22 23 24 25 26 27 28      19 20 21 22 23 24 25      17 18 19 20 21 22 23
29 30 31                  26 27 28 29 30          24 25 26 27 28 29 30
                                   31

CSDN @谁怕？一蓑烟雨任平生
```

编辑

| cal -3 ：显示系统前一个月，当前月，下一个月的月历

编辑 17.find、which、whereis指令

```
[junmoxiao@VM-24-6-centos ~]$ cal -3
      April 2023      May 2023      June 2023
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
                1      1 2 3 4 5 6                1 2 3
 2  3  4  5  6  7  8   7  8  9 10 11 12 13   4  5  6  7  8  9 10
 9 10 11 12 13 14 15  14 15 16 17 18 19 20  11 12 13 14 15 16 17
16 17 18 19 20 21 22  21 22 23 24 25 26 27  18 19 20 21 22 23 24
23 24 25 26 27 28 29  28 29 30 31          25 26 27 28 29 30
30                                     CSDN @谁怕? 一蓑烟雨任平生
```

find

Linux下find命令在目录结构中搜索文件，并执行指定的操作。

Linux下find命令提供了相当多的查找条件，功能很强大。由于find具有强大的功能，所以它的选项也很

多，其中大部分选项都值得我们花时间来了解一下。

即使系统中含有网络文件系统(NFS)，find命令在该文件系统中同样有效，只你具有相应的权限。

在运行一个非常消耗资源的find命令时，很多人都倾向于把它放在后台执行，因为遍历一个大的文件系

统可能会花费很长的时间(这里是指30G字节以上的文件系统)。

语法： find pathname -options

功能： 用于在文件树中查找文件，并作出相应的处理（可能访问磁盘）

常用选项：

-name 按照文件名查找文件

find在查找时，第一次查找较慢，第二次很快，Linux会把查找的结果缓存起来。

find 路径 -name 查找名称：把从这个路径开始的所有 带有查找名称 的文件查找出来

```
[junmoxiao@VM-24-6-centos ~]$ find -name python
./VimForCpp/vim/bundle/YouCompleteMe/third_party/ycmd/third_party/parso/parso/python
./VimForCpp/vim/bundle/YouCompleteMe/third_party/ycmd/ycmd/tests/python
./VimForCpp/vim/bundle/YouCompleteMe/third_party/ycmd/ycmd/completers/python
./VimForCpp/vim/bundle/YouCompleteMe/python
./VimForCpp/vim/bundle/python-mode/ftplugin/python
./VimForCpp/vim/bundle/LeaderF/autoload/leaderf/python
```

CSDN @谁怕？一蓑烟雨任平生

编辑

which

which 指令：可以查看指令路径

比如查看 clear 的路径：

| which clear

```
[junmoxiao@VM-24-6-centos ~]$ which clear
/bin/clear
```

CSDN @谁怕？一蓑烟雨任平生

编辑

我们常用的指令都在 /usr/bin 位置

whereis

whereis 搜索指定名称的文件/程序/归档文件

| whereis python

```
[junmoxiao@VM-24-6-centos ~]$ whereis python
python: /usr/bin/python3.6m /usr/bin/python3.6 /usr/bin/python /usr/bin/python2.7-config /usr/bin/python2.7 /usr/lib/python3.6 /usr/lib/python2.7 /usr/lib64/python3.6 /usr/lib64/python2.7 /etc/python /usr/include/python3.6m /usr/include/python2.7 /usr/share/man/man1/python.1.gz
```

CSDN @谁怕？一蓑烟雨任平生

编辑

18. grep指令：

语法：grep [选项] 搜寻字符串文件

功能：在文件中搜索字符串，将找到的行打印出来

常用选项：

- 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：反向选择，亦即显示出没有 '搜寻字符串' 内容的那一行

grep 为文本行过滤，说白了就是将指定文本内容，按照特定关键字，按行筛选。

举例：

grep '字符串' 文件：筛选出 含有该字符串 的内容

```
[junmoxiao@VM-24-6-centos ~]$ grep 'linux' linux.c
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
printf("hello linux\n");
```

CSDN @谁怕？—袁烟雨任平生

编辑

grep -v '字符串' 文件：筛选出 不含有该字符串 的内容

```
[junmoxiao@VM-24-6-centos ~]$ grep -v 'linux' linux.c
#include<stdio.h>
int main()
{
    return 0;
}
```

CSDN @谁怕？—袁烟雨任平生

编辑

| grep -i '字符串' 文件：忽略大小写 正向匹配

| grep -ivn '字符串' 文件：忽略大小写反向匹配并给上行号

| cat test.txt | grep 'a'：通过管道来匹配 a

19. zip/unzip指令：

语法：zip 压缩文件.zip 目录或文件

功能：将目录或文件压缩成zip格式

常用选项：

| -r 递归处理，将指定目录下的所有文件和子目录一并处理

举例：

| 将test2目录压缩：zip test2.zip test2/*

| 解压到tmp目录：unzip test2.zip -d /tmp

| zip -r 压缩文件.zip 目录或文件名：压缩一个文件

```
[junmoxiao@VM-24-6-centos testzip]$ ll
total 4
drwxrwxr-x 3 junmoxiao junmoxiao 4096 May 29 22:42 t1
[junmoxiao@VM-24-6-centos testzip]$ zip -r tmp.zip t1
  adding: t1/ (stored 0%)
  adding: t1/t2/ (stored 0%)
  adding: t1/t2/t3/ (stored 0%)
[junmoxiao@VM-24-6-centos testzip]$ ll
total 8
drwxrwxr-x 3 junmoxiao junmoxiao 4096 May 29 22:42 t1
-rw-rw-r-- 1 junmoxiao junmoxiao  442 May 29 22:48 tmp.zip
```

编辑

| unzip 压缩文件 -d 路径：解压到指定目录

20.tar指令

语法：tar [-cxtzjvf] 文件与目录 ... 参数：

常用选项：

- c ：建立一个压缩文件的参数指令(create 的意思)；
- x ：解开一个压缩文件的参数指令！
- t ：查看 tarfile 里面的文件！
- z ：是否同时具有 gzip 的属性？亦即是否需要用 gzip 压缩？
- j ：是否同时具有 bzip2 的属性？亦即是否需要用 bzip2 压缩？
- v ：压缩的过程中显示文件！这个常用，但不建议用在背景执行过程！
- f ：使用档名，请留意，在 f 之后要立即接档名喔！不要再加参数！
- C ： 解压到指定目录

tar 是一个神奇的指令，它既可以 压缩，又可以 解压。

tar 打包后文件的后缀为 tgz

举例：

- tar -czf dst src ：将 src 中的内容压缩到 dst 中。
- tar 命令默认可以文件(包括目录)中的文件全部打包。
- tar -czf 的 - 可以省略，但是建议不要省略。

- tar -xzf dst：将压缩文件 dst 指定到当前路径
- tar -xzf 的 - 可以省略，但是建议不要省略。

- tar -ztvf dst：不解压文件，查看压缩包 dst 内有什么。

| tar xzf dst -C 路径：将压缩文件 dst 解压到指定路径

21.bc指令

tar [-cxtzjv] 文件与目录 参数：

bc命令可以很方便的进行浮点运算。

在平时，按下 bc 后会阻塞：

此时就可以进行计算，**bc 相当于 Linux 中的计算器**：

而我们平常也可以通过这种方式，进行 **数据计算**：

| echo "114+514"|bc

echo 将114+514输送到管道中，bc 直接取管道中的值进行计算。

| 到这篇文章内容就结束了，希望以上内容对大家能有帮助，谢谢！！！！