

---

# 目录

说明	1.1
----	-----

---

## I Linux命令

dpkg	2.1
useradd	2.2
cat	2.3
sed	2.4

---

## II 个人笔记

C语言笔记	3.1
adb刷机	3.2
树莓派	3.3
其他	3.4

---

- 说明

## 说明

1. linux目录记录个人学习linux时用到的一些命令
2. notes目录记录个人感兴趣的一些笔记

- [dpkg命令](#)
  - [dpkg软件包相关文件介绍](#)
  - [参考实例](#)

## dpkg命令

dpkg命令的英文全称是“Debian package”，dpkg是Debian的一个底层包管理工具，主要用于对已下载到本地和已安装的软件包进行管理。

dpkg这个机制最早由Debian Linux社区所开发出来的，通过dpkg的机制，Debian提供的软件就能够简单的安装起来，同时能提供安装后的软件信息，实在非常不错。只要派生于Debian的其它Linux distributions大多使用dpkg这个机制来管理，包括B2D，Ubuntu等。

## dpkg软件包相关文件介绍

`/etc/dpkg/dpkg.cfg` dpkg包管理软件的配置文件 【Configuration file with default options】

`/var/log/dpkg.log` dpkg包管理软件的日志文件 【Default log file (see /etc/dpkg/dpkg.cfg(5) and option --log)】

`/var/lib/dpkg/available` 存放系统所有安装过的软件包信息 【List of available packages.】

`/var/lib/dpkg/status` 存放系统现在所有安装软件的状态信息

`/var/lib/dpkg/info` 记安装软件包控制目录的控制信息文件

参数	说明
-i	安装软件包
-r	删除软件包
-l	显示已安装软件包列表
-L	显示于软件包关联的文件
-c	显示软件包内文件列表

## 参考实例

安装包：

```
dpkg -i package.deb
```

删除包：

```
dpkg -r package.deb
```

`dpkg -r package-name # --remove`, 移除软件包，但保留其配置文件

`dpkg -P package-name # --purge`, 清除软件包的所有文件 (removes everything, including conffiles)

列出当前已安装的包：

```
dpkg -l
```

每条记录对应一个软件包，注意每条记录的第一、二、三个字符，这就是软件包的状态标识，后边依次是软件包名称、版本号和简单描述。

1. 第一字符为期望值(Desired=Unknown/Install/Remove/Purge/Hold)，它包括：
  - u Unknown状态未知,这意味着软件包未安装,并且用户也未发出安装请求.
  - i Install用户请求安装软件包.
  - r Remove用户请求卸载软件包.
  - p Purge用户请求清除软件包.
  - h Hold用户请求保持软件包版本锁定.
2. 第二列,是软件包的当前状态(Status=Not/Inst/Conf-files/Unpacked/half-conf/Half-inst/trig-aWait/Trig-pend)
  - n Not软件包未安装.
  - i Inst软件包安装并完成配置.
  - c Conf-files软件包以前安装过,现在删除了,但是它的配置文件还留在系统中.
  - u Unpacked软件包被解包,但还未配置.
  - f half-conf试图配置软件包,但是失败了.
  - h Half-inst软件包安装,但是但是没有成功.
  - w trig-aWait触发器等待
  - t Trig-pend触发器未决
3. 第三列标识错误状态,第一种状态标识没有问题,为空. 其它符号则标识相应问题 (Err?=(none)/Reinst-required (Status,Err: uppercase=bad))
  - h 软件包被强制保持,因为有其它软件包依赖需求,无法升级.
  - r Reinst-required, 软件包被破坏,可能需要重新安装才能正常使用(包括删除).
  - x 软件包被破坏,并且被强制保持.
4. 案例说明： ii —— 表示系统正常安装了该软件  
 pn —— 表示安装了该软件，后来又清除了  
 un —— 表示从未安装过该软件  
 iu —— 表示安装了该软件，但是未配置  
 rc —— 该软件已被删除，但配置文件仍在
5. dpkg子命令 为了方便用户使用， dpkg不仅提供了大量的参数选项, 同时也提供了许多子命令。比如：  
 dpkg-deb、dpkg-divert、dpkg-query、dpkg-split、dpkg-statoverride、start-stop-daemon

列出deb包的内容：

```
dpkg -c package.deb
```

配置

```
dpkg --configure package
```

查询

```
dpkg -l package-name-pattern # --list, 查看系统中软件包名符合pattern模式的软件包
```

```
dpkg -L package-name # --listfiles, 查看package-name对应的软件包安装的文件及目录
```

```
dpkg -p package-name # --print-avail, 显示包的具体信息
```

```
dpkg -s package-name # --status, 查看package-name (已安装) 对应的软件包信息
```

```
dpkg -S filename-search-pattern # --search, 从已经安装的软件包中查找包含filename的软件包名称
```

更多dpkg的使用方法可在命令行里使用 `man dpkg` 来查阅 或直接使用 `dpkg --help`。

- useradd命令详解
  - 语法
  - 参数说明
  - 实例

## useradd命令详解

[原文链接](#)

### 语法

```
useradd [-mMnr] [-c <备注>] [-d <登入目录>] [-e <有效期限>] [-f <缓冲天数>] [-g <群组>] [-G <群组>] [-s <shell>] [-u <uid>] [用户帐号]
```

或

```
useradd -D [-b] [-e <有效期限>] [-f <缓冲天数>] [-g <群组>] [-G <群组>] [-s <shell>]
```

### 参数说明

- -c<备注> 加上备注文字。备注文字会保存在passwd的备注栏位中。
- -d<登入目录> 指定用户登入时的起始目录。
- -D 变更预设值。
- -e<有效期限> 指定帐号的有效期限。
- -f<缓冲天数> 指定在密码过期后多少天即关闭该帐号。
- -g<群组> 指定用户所属的群组。
- -G<群组> 指定用户所属的附加群组。
- -m 自动建立用户的登入目录。
- -M 不要自动建立用户的登入目录。
- -n 取消建立以用户名为名的群组。
- -r 建立系统帐号。
- -s 指定用户登入后所使用的shell。
- -u 指定用户ID。

### 实例

添加一般用户

```
useradd tt
```

为添加的用户指定相应的用户组

```
useradd -g root tt
```

创建一个系统用户

```
useradd -r tt
```

useradd

为新添加的用户指定home目录

```
useradd -d /home/myd tt
```

建立用户且制定ID

```
useradd caojh -u 544
```

禁止用户wsj0051登录

```
usermod -s /sbin/nologin wsj0051
```

恢复用户wsj0051登录

```
usermod -s /bin/bash wsj0051
```

其他

#添加一个不能登录的用户

```
useradd -d /usr/local/apache -g apache -s /bin/false apache
```

要拒绝系统用户登录，可以将其 shell 设置为 /usr/sbin/nologin 或者 /bin/false。

```
usermod -s | --shell /usr/sbin/nologin username
```

或者

```
usermod -s | -shell /bin/false username
```

说明及比较：

```
/bin/false
```

/bin/false 什么也不做只是返回一个错误状态，然后立即退出。将用户的 shell 设置为 /bin/false，用户会无法登录，并且不会有任何提示。

```
/usr/sbin/nologin
```

nologin 会礼貌的向用户显示一条信息，并拒绝用户登录：

```
This account is currently not available.
```

有一些软件，比如一些 ftp 服务器软件，对于本地非虚拟账户，只有用户有有效的 shell 才能使用 ftp 服务。这时候就可以使用 nologin 使用户即不能登录系统，还能使用一些系统服务，比如 ftp 服务。/bin/false 则不行，这是二者的重要区别之一。

```
/etc/nologin
```

如果存在 /etc/nologin 文件，则系统只允许 root 用户登录，其他用户全部被拒绝登录，并向他们显示 /etc/nologin 文件的内容。

图 - 笔记

- cat命令
  - 参数说明
  - cat主要有三大功能
  - 实例

## cat命令

### 参数说明

- -n 或 --number: 由 1 开始对所有输出的行数编号。
- -b 或 --number-nonblank: 和 -n 相似, 只不过对于空白行不编号。
- -s 或 --squeeze-blank: 当遇到有连续两行以上的空白行, 就替换为一行的空白行。
- -v 或 --show-nonprinting: 使用 ^ 和 M- 符号, 除了 LFD 和 TAB 之外。
- -E 或 --show-ends : 在每行结束处显示 \$。
- -T 或 --show-tabs: 将 TAB 字符显示为 ^I。
- -A, --show-all: 等价于 -vET。
- -e: 等价于"-vE"选项;
- -t: 等价于"-vT"选项;

cat命令是linux下的一个文本输出命令, 通常是用于观看某个文件的内容的;

### cat主要有三大功能

1. 一次显示整个文件。

```
$ cat filename
```

2. 从键盘创建一个文件。只能创建新文件,不能编辑已有文件.

```
$ cat > filename
```

cat > test.sh << EOF 以EOF作为输入结束, 之前存在的内容会被覆盖掉。  
 cat << EOF >> test.sh 将内容追加到 test.sh 的后面, 不会覆盖掉原有的内容  
 EOF只是标识, 不是固定的

```
cat << HHH > iii.txt
```

输入内容:

```
> sdlkfjksl
> sdkjfkl
> asdlfj
> HHH
```

这里的“HHH”就代替了“EOF”的功能。结果是相同的。引用 `cat iii.txt` 可以看到结果:

```
sdlkfjksl
sdkjfkl
asdlfj
```

3. 将几个文件合并为一个文件。将file1 file合并到file

```
$ cat file1 file2 > file
```

把 textfile1 的文档内容加上行号后输入 textfile2 这个文档里：

```
cat -n textfile1 > textfile2
```

把 textfile1 和 textfile2 的文档内容加上行号（空白行不加）之后将内容附加到 textfile3 文档里：

```
cat -b textfile1 textfile2 >> textfile3
```

## 实例

清空 /etc/test.txt 文档内容：

```
cat /dev/null > /etc/test.txt
```

cat 也可以用来制作镜像文件。例如要制作软盘的镜像文件，将软盘放好后输入：

```
cat /dev/fd0 > OUTFILE
```

相反的，如果想把 image file 写到软盘，输入：

```
cat IMG_FILE > /dev/fd0
```

注：

1. OUTFILE 指输出的镜像文件名。
2. IMG\_FILE 指镜像文件。
3. 若从镜像文件写回 device 时，device 容量需与相当。
4. 通常用制作开机磁片。

- **sed命令**
  - 命令格式
  - 选项
  - **sed常用命令**
  - **sed替换标记**
  - **实例**
    - 全面替换标记g
    - 定界符
    - 删除操作：d命令
    - 子串匹配标记\1
    - 选定行的范围：, (逗号)
    - 多点编辑：e命令

## sed命令

sed是一种流编辑器，它是文本处理中非常好的工具，能够完美的配合正则表达式使用，功能不同凡响。处理时，把当前处理的行存储在临时缓冲区中，称为“模式空间”(pattern space)，接着用sed命令处理缓冲区中的内容，处理完成后，把缓冲区的内容送往屏幕。接着处理下一行，这样不断重复，直到文件末尾。文件内容并没有改变，除非你使用重定向存储输出。Sed主要用来自动编辑一个或多个文件，可以将数据行进行替换、删除、新增、选取等特定工作，简化对文件的反复操作，编写转换程序等。

## 命令格式

sed的命令格式： sed [options] 'command' file(s);

sed的脚本格式： sed [options] -f scriptfile file(s);

## 选项

- e：直接在命令行模式上进行sed动作编辑，此为默认选项；
- f：将sed的动作写在一个文件内，用-f filename 执行filename内的sed动作；
- i：直接修改文件内容；
- n：只打印模式匹配的行；
- r：支持扩展表达式；
- h或--help：显示帮助；
- V或--version：显示版本信息。

## sed常用命令

- a\ 在当前行下面插入文本；
- i\ 在当前行上面插入文本；
- c\ 把选定的行改为新的文本；
- d\ 删除，删除选择的行；
- D\ 删除模板块的第一行；

s 替换指定字符;

h 拷贝模板块的内容到内存中的缓冲区;

H 追加模板块的内容到内存中的缓冲区;

g 获得内存缓冲区的内容，并替代当前模板块中的文本;

G 获得内存缓冲区的内容，并追加到当前模板块文本的后面;

l 列表不能打印字符的清单;

n 读取下一个输入行，用下一个命令处理新的行而不是用第一个命令;

N 追加下一个输入行到模板块后面并在二者间嵌入一个新行，改变当前行号码;

p 打印模板块的行。 P(大写) 打印模板块的第一行;

q 退出Sed;

b label 分支到脚本中带有标记的地方，如果分支不存在则分支到脚本的末尾;

r file 从file中读行;

t label if分支，从最后一行开始，条件一旦满足或者T，t命令，将导致分支到带有标号的命令处，或者到脚本的末尾;

T label 错误分支，从最后一行开始，一旦发生错误或者T，t命令，将导致分支到带有标号的命令处，或者到脚本的末尾;

w file 写并追加模板块到file末尾;

W file 写并追加模板块的第一行到file末尾;

! 表示后面的命令对所有没有被选定的行发生作用;

= 打印当前行号;

## sed替换标记

g 表示行内全面替换;

p 表示打印行;

w 表示把行写入一个文件;

x 表示互换模板块中的文本和缓冲区中的文本;

y 表示把一个字符翻译为另外的字符（但是不用于正则表达式）;

\1 子串匹配标记;

& 已匹配字符串标记;

## 实例

替换文本中的字符串：

```
sed 's/book/books/' file
```

-n 选项和 p 命令一起使用表示只打印那些发生替换的行：

```
sed -n 's/test/TEST/p' file
```

直接编辑文件选项-i，会匹配file文件中每一行的第一个book替换为books

```
sed -i 's/book/books/g' file
```

## 全面替换标记g

使用后缀 /g 标记会替换每一行中的所有匹配：

```
sed 's/book/books/g' file
```

当需要从第N处匹配开始替换时，可以使用 /Ng：

```
echo sksksksksk | sed 's/sk/SK/2g'  
skSKSKSKSK  
echo sksksksksk | sed 's/sk/SK/3g'  
skskSKSKSK  
echo sksksksksk | sed 's/sk/SK/4g'  
skskskSKSKSK
```

## 定界符

以上命令中字符 / 在sed中作为定界符使用，也可以使用任意的定界符

```
sed 's:test:TEXT:g'  
sed 's|test|TEXT|g'
```

定界符出现在样式内部时，需要进行转义：

```
sed 's/\/bin\/\usr\/local\/bin/g'
```

## 删除操作：d命令

删除空白行：

```
sed '/^$/d' file
```

删除文件的第2行：

```
sed '2d' file
```

删除文件的第2行到末尾所有行：

```
sed '2,$d' file
```

删除文件最后一行：

```
sed '$d' file
```

删除文件中所有开头是test的行：

```
sed '/^test/'d file
```

已匹配字符串标记& 正则表达式 \w+ 匹配每一个单词，使用 [&] 替换它，& 对应于之前所匹配到的单词：

```
echo this is a test line | sed 's/\w+/[&]/g'
[this] [is] [a] [test] [line]
```

所有以192.168.0.1开头的行都会被替换成它自己加localhost：

```
sed 's/^192.168.0.1/&localhost/' file
192.168.0.1localhost
```

## 子串匹配标记\1

匹配给定样式的其中一部分：

```
echo this is digit 7 in a number | sed 's/digit \([0-9]\)/\1/'
this is 7 in a number
```

命令中 digit 7，被替换成了 7。样式匹配到的子串是 7，(..) 用于匹配子串，对于匹配到的第一个子串就标记为 \1，依此类推匹配到的第二个结果就是 \2，例如：

```
echo aaa BBB | sed 's/(\([a-z]\+\)\ \([A-Z]\+\))/\2 \1/'
BBB aaa
```

love被标记为1，所有loveable会被替换成lovers，并打印出来：

```
sed -n 's/(love\)\able/\1rs/p' file
```

## 选定行的范围：, (逗号)

所有在模板test和check所确定的范围内的行都被打印：

```
sed -n '/test/,/check/p' file
```

打印从第5行开始到第一个包含以test开始的行之间的所有行：

```
sed -n '5,/^test/p' file
```

对于模板test和west之间的行，每行的末尾用字符串aaa bbb替换：

```
sed '/test/,/west/s/$/aaa bbb/' file
```

## 多点编辑：e命令

-e 选项允许在同一行里执行多条命令：

```
sed -e '1,5d' -e 's/test/check/' file
```

上面sed表达式的第一条命令删除1至5行，第二条命令用check替换test。命令的执行顺序对结果有影响。如果两个命令都是替换命令，那么第一个替换命令将影响第二个替换命令的结果。

和 -e 等价的命令是 --expression：

```
sed --expression='s/test/check/' --expression='/love/d' file
```

[更多详情](#)

- C语言学习笔记
  - C语言中的 32 个关键字
  - 二进制
  - 八进制
  - 十六进制
  - 将二进制、八进制、十六进制转换为十进制
  - 将十进制转换为二进制、八进制、十六进制
  - 二进制和八进制、十六进制的转换

# C语言学习笔记

## C语言中的 32 个关键字

int	float	double	char	short	long	signed	unsigned
if	else	switch	case	default	for	while	do
break	continue	return	void	const	sizeof	struct	typedef
static	extern	auto	register	enum	goto	union	volatile

## 二进制

二进制加法：1+0=1、1+1=10、11+10=101、111+111=1110

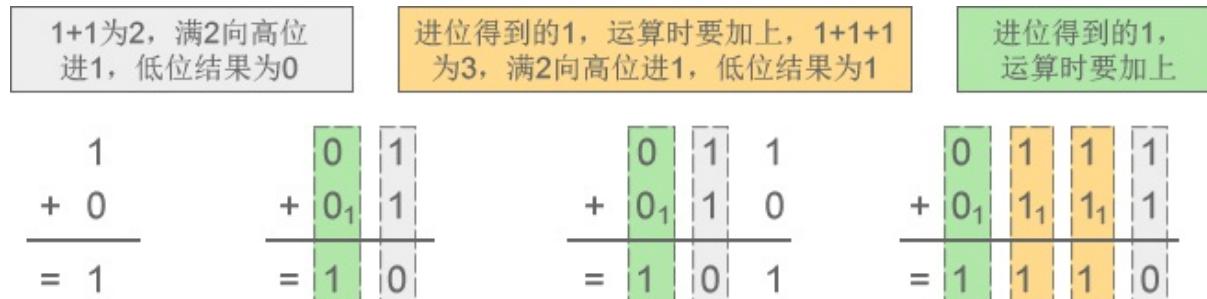


图 - 二进制加法示意图

二进制减法：1-0=1、10-1=1、101-11=10、1100-111=101

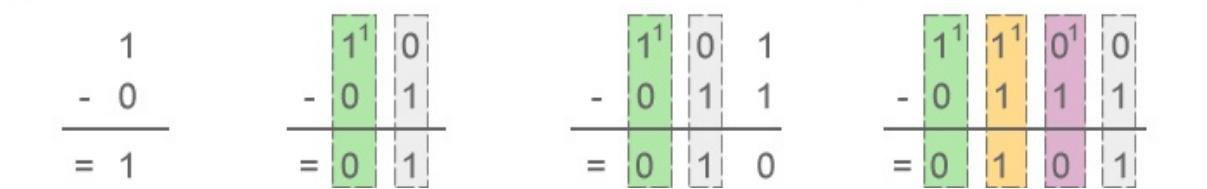
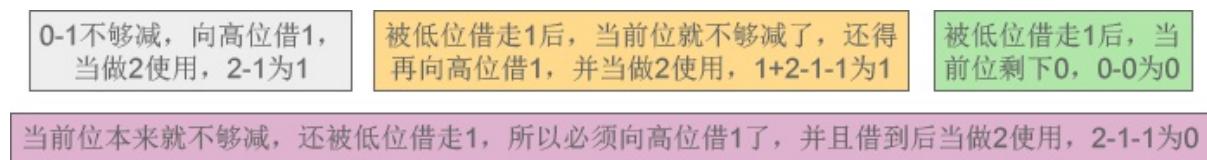


图 - 二进制减法示意图

## 八进制

八进制加法： $3+4=7$ 、 $5+6=13$ 、 $75+42=137$ 、 $2427+567=3216$

当前位满8，向高位进1，然后再减去8，就是当前位的和	进位得到的1，运算时要加上。当前位满8，向高位进1，然后再减去8，就是当前位的和	进位得到的1，运算时要加上	
$  \begin{array}{r}  3 \\  + 4 \\  \hline  = 7  \end{array}  $	$  \begin{array}{r}  0 \quad 5 \\  + 0_1 \quad 6 \\  \hline  = 1 \quad 3  \end{array}  $	$  \begin{array}{r}  0 \quad 7 \quad 5 \\  + 0_1 \quad 4 \quad 2 \\  \hline  = 1 \quad 3 \quad 7  \end{array}  $	$  \begin{array}{r}  2 \quad 4 \quad 2 \quad 7 \\  + 0_1 \quad 5_1 \quad 6_1 \quad 7 \\  \hline  = 3 \quad 2 \quad 1 \quad 6  \end{array}  $

图 - 八进制加法示意图

八进制减法： $6-4=2$ 、 $52-27=23$ 、 $307-141=146$ 、 $7430-1451=5757$

当前位不够减，向高位借1，当做8使用	被低位借走1后，当前位就不够减了，还得再向高位借1，并当做8使用	被低位借走的1，运算时要减去	
当前位本来就不够减，还被低位借走1，所以必须向高位借1了，并且借到后当做8使用			
$  \begin{array}{r}  6 \\  - 4 \\  \hline  = 2  \end{array}  $	$  \begin{array}{r}  5^1 \quad 2 \\  - 2 \quad 7 \\  \hline  = 2 \quad 3  \end{array}  $	$  \begin{array}{r}  3^1 \quad 0 \quad 7 \\  - 1 \quad 4 \quad 1 \\  \hline  = 1 \quad 4 \quad 6  \end{array}  $	$  \begin{array}{r}  7^1 \quad 4^1 \quad 3^1 \quad 0 \\  - 1 \quad 4 \quad 5 \quad 1 \\  \hline  = 5 \quad 7 \quad 5 \quad 7  \end{array}  $

## 十六进制

十六进制中，用A来表示10，B表示11，C表示12，D表示13，E表示14，F表示15，因此有0~F共16个数字，基数为16，加法运算时逢16进1，减法运算时借1当16。例如，数字0、1、6、9、A、D、F、419、EA32、80A3、BC00都是有效的十六进制。

十六进制加法： $6+7=D$ 、 $18+BA=D2$ 、 $595+792=D27$ 、 $2F87+F8A=3F11$

当前位满16，向高位进1，  
然后再减去16，就是  
当前位的和

进位得到的1，运算时要加上。  
当前位满16，向高位进1，  
然后再减去16，就是当前位的和

进位得到的1，  
运算时要加上

$\begin{array}{r} 6 \\ + 7 \\ \hline = D \end{array}$	$\begin{array}{r} 1 & 8 \\ + B_1 & A \\ \hline = D & 2 \end{array}$	$\begin{array}{r} 5 & 9 & 5 \\ + 7_1 & 9 & 2 \\ \hline = D & 2 & 7 \end{array}$	$\begin{array}{r} 2 & F & 8 & 7 \\ + 0_1 & F_1 & 8_1 & A \\ \hline = 3 & F & 1 & 1 \end{array}$
---	---	---	---

图 - 十六进制加法示意图

十六进制减法：D-3=A、52-2F=23、E07-141=CC6、7CA0-1CB1=5FEF

当前位不够减，向高位借1，当做16使用

被低位借走1后，当前位就不够减了，还得再向高位借1，并当做16使用

被低位借走的1，  
运算时要减去

当前位本来就不够减，还被低位借走1，所以必须向高位借1了，并且借到后当做16使用

$\begin{array}{r} D \\ - 3 \\ \hline = A \end{array}$	$\begin{array}{r} 5^1 & 2 \\ - 2 & F \\ \hline = 2 & 3 \end{array}$	$\begin{array}{r} E^1 & 0 & 7 \\ - 1 & 4 & 1 \\ \hline = C & C & 6 \end{array}$	$\begin{array}{r} 7^1 & C^1 & A^1 & 0 \\ - 1 & C & B & 1 \\ \hline = 5 & F & E & F \end{array}$
---	---	---	---

图 - 十六进制减法示意图

## 将二进制、八进制、十六进制转换为十进制

二进制、八进制和十六进制向十进制转换都非常容易，就是“按权相加”。所谓“权”，也即“位权”。

假设当前数字是 N 进制，那么：

- 对于整数部分，从右往左看，第 i 位的位权等于 $\$N^{i-1}\$\$$
- 对于小数部分，恰好相反，要从左往右看，第 j 位的位权为 $\$N^{-j}\$\$$ 。

更加通俗的理解是，假设一个多位数（由多个数字组成的数）某位上的数字是 1，那么它所表示的数值大小就是该位的位权。

转换成十进制的例子：

- 二进制： $\$1001 = 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0\$\$ = 8 + 0 + 0 + 1 = 9$ （十进制）
- 二进制： $\$101.1001 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4} = 4 + 0 + 1 + 0.5 + 0 + 0 + 0.0625 = 5.5625\$\$$ （十进制）
- 八进制： $\$302 = 3 \times 8^2 + 0 \times 8^1 + 2 \times 8^0\$\$ = 192 + 0 + 2 = 194$ （十进制）
- 八进制： $\$302.46 = 3 \times 8^2 + 0 \times 8^1 + 2 \times 8^0 + 4 \times 8^{-1} + 6 \times 8^{-2} = 192 + 0 + 2 + 0.5 + 0.09375 = 194.59375\$\$$ （十进制）
- 十六进制： $\$EA7 = 14 \times 16^2 + 10 \times 16^1 + 7 \times 16^0 = 3751\$\$$ （十进制）

## 将十进制转换为二进制、八进制、十六进制

将十进制转换为其它进制时比较复杂，整数部分和小数部分的算法不一样。

### 整数部分

十进制整数转换为 N 进制整数采用“除 N 取余，逆序排列”法。具体做法是：

- 将 N 作为除数，用十进制整数除以 N，可以得到一个商和余数；
- 保留余数，用商继续除以 N，又得到一个新的商和余数；
- 仍然保留余数，用商继续除以 N，还会得到一个新的商和余数；
- .....
- 如此反复进行，每次都保留余数，用商接着除以 N，直到商为 0 时为止。

把先得到的余数作为 N 进制数的低位数字，后得到的余数作为 N 进制数的高位数字，依次排列起来，就得到了 N 进制数字。

下图演示了将十进制数字 36926 转换成八进制的过程：

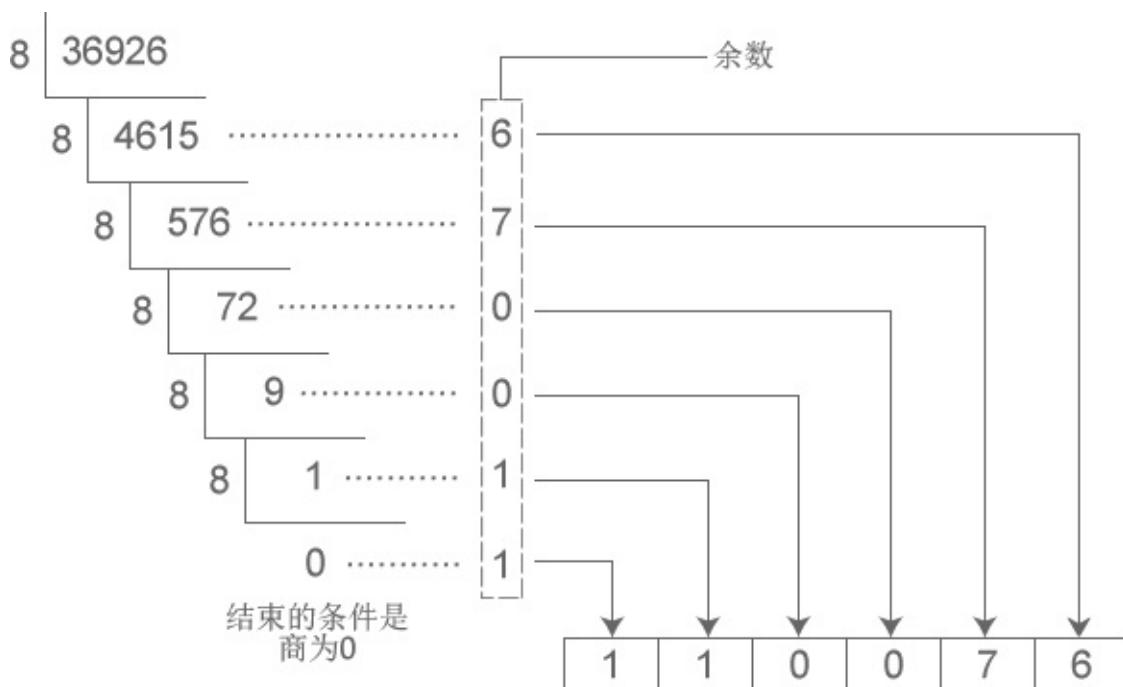


图 - 36926 转换成八进制

从图中得知，十进制数字 36926 转换成八进制的结果为 110076。

下图演示了将十进制数字 42 转换成二进制的过程：

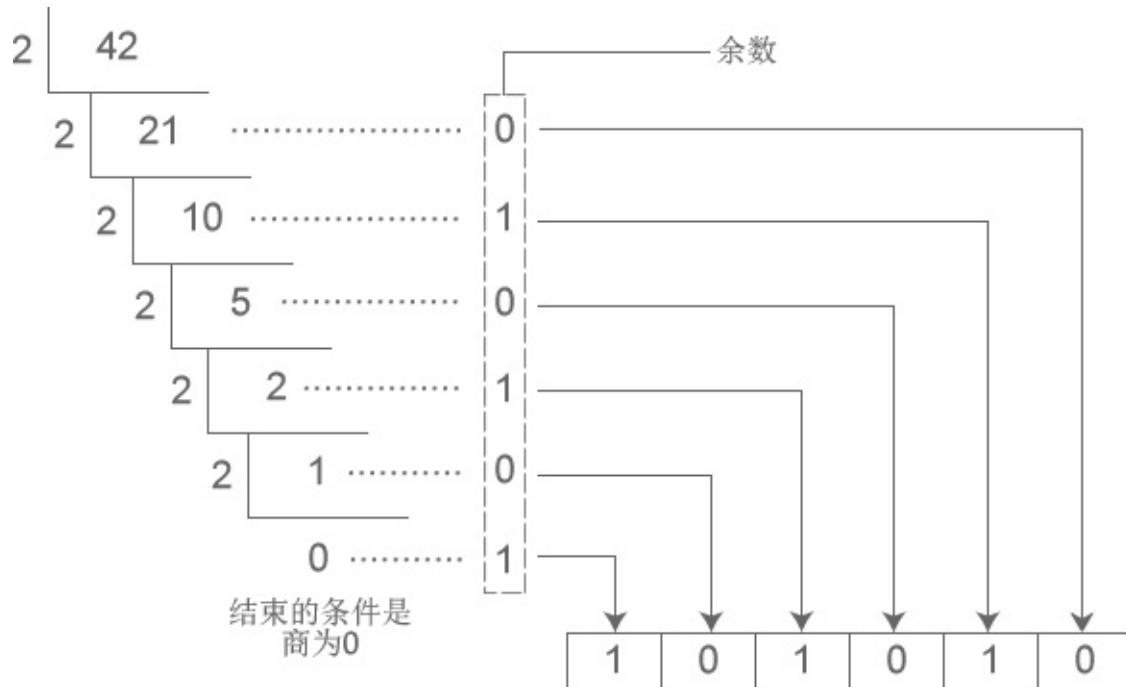


图 - 42 转换成二进制

从图中得知，十进制数字 42 转换成二进制的结果为 101010。

### 小数部分

十进制小数转换成 N 进制小数采用“乘 N 取整，顺序排列”法。具体做法是：

- 用 N 乘以十进制小数，可以得到一个积，这个积包含了整数部分和小数部分；
- 将积的整数部分取出，再用 N 乘以余下的小数部分，又得到一个新的积；
- 再将积的整数部分取出，继续用 N 乘以余下的小数部分；
- .....
- 如此反复进行，每次都取出整数部分，用 N 接着乘以小数部分，直到积中的小数部分为 0，或者达到所要求的精度为止。

把取出的整数部分按顺序排列起来，先取出的整数作为 N 进制小数的高位数字，后取出的整数作为低位数字，这样就得到了 N 进制小数。

下图演示了将十进制小数 0.930908203125 转换成八进制小数的过程：

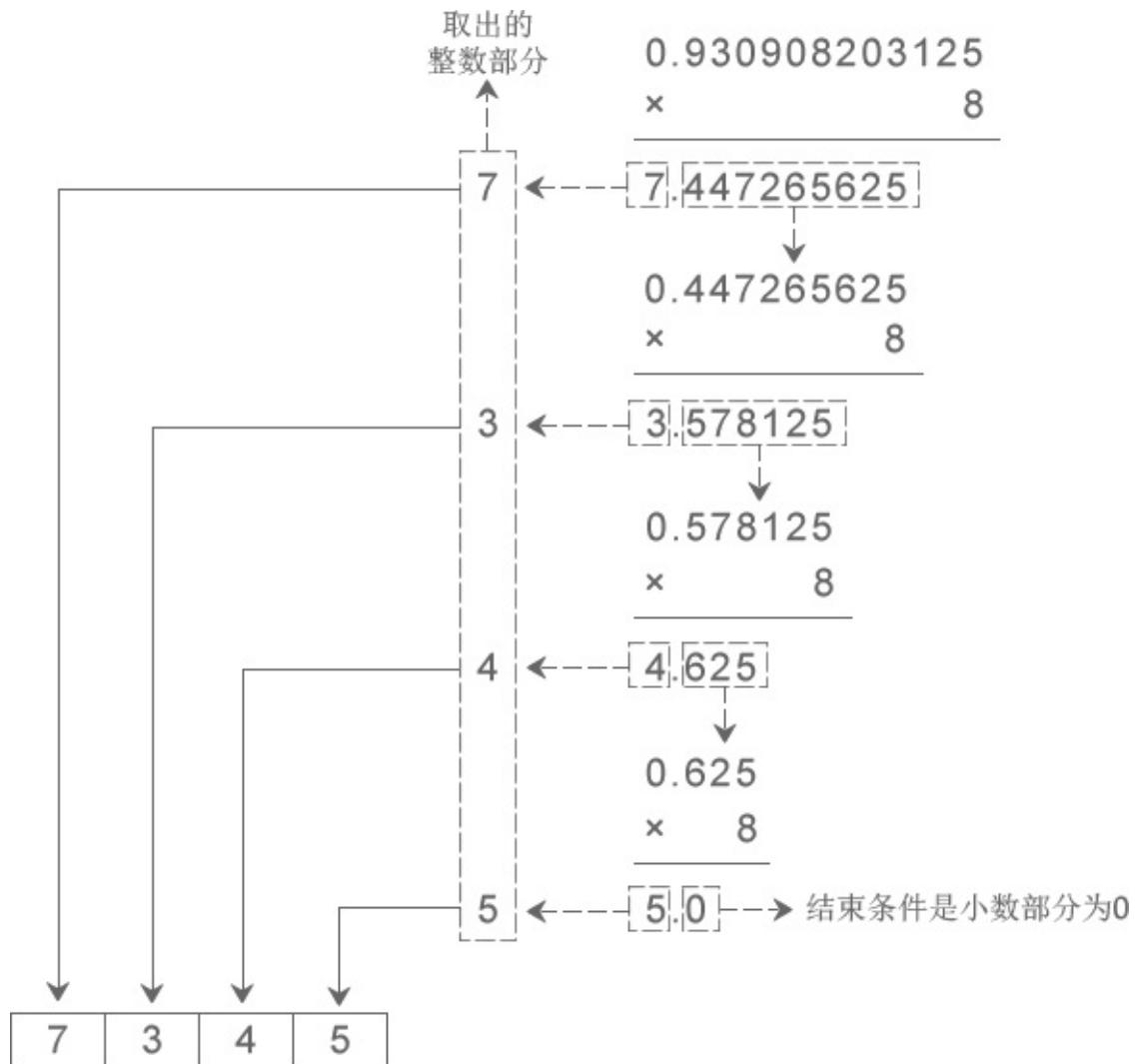


图 - 0.930908203125 转换成八进制

从图中得知，十进制小数 0.930908203125 转换成八进制小数的结果为 0.7345。

下表列出了前 17 个十进制整数与二进制、八进制、十六进制的对应关系：

十进制	0	1	2	3	4	5	6	7	8	9	10	11	12
二进制	0	1	10	11	100	101	110	111	1000	1001	1010	1011	1100
八进制	0	1	2	3	4	5	6	7	10	11	12	13	14
十六进制	0	1	2	3	4	5	6	7	8	9	A	B	C

## 二进制和八进制、十六进制的转换

其实，任何进制之间的转换都可以使用上面讲到的方法，只不过有时比较麻烦，所以一般针对不同的进制采取不同的方法。将二进制转换为八进制和十六进制时就有非常简洁的方法，反之亦然。

### 二进制整数和八进制整数之间的转换

二进制整数转换为八进制整数时，每三位二进制数字转换为一位八进制数字，运算的顺序是从低位向高位依次进行，高位不足三位用零补齐。下图演示了如何将二进制整数 1110111100 转换为八进制：

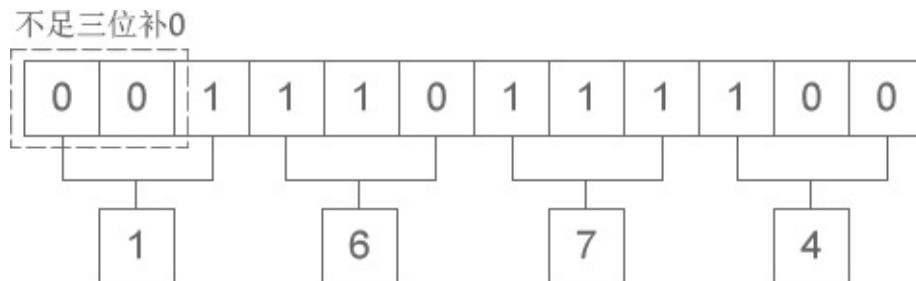


图 - 1110111100 转换为八进制

从图中可以看出，二进制整数 1110111100 转换为八进制的结果为 1674。

八进制整数转换为二进制整数时，思路是相反的，每一位八进制数字转换为三位二进制数字，运算的顺序也是从低位向高位依次进行。下图演示了如何将八进制整数 2743 转换为二进制：

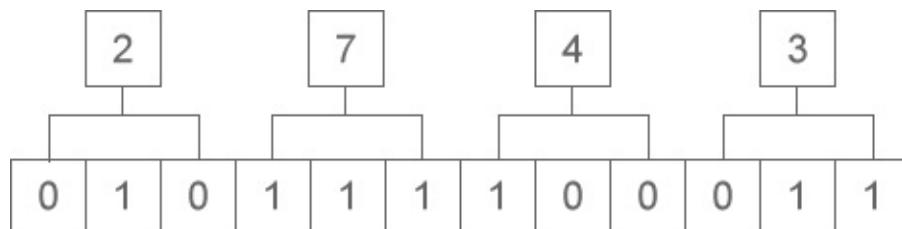


图 - 2743 转换为二进制

从图中可以看出，八进制整数 2743 转换为二进制的结果为 10111100011。

### 二进制整数和十六进制整数之间的转换

二进制整数转换为十六进制整数时，每四位二进制数字转换为一位十六进制数字，运算的顺序是从低位向高位依次进行，高位不足四位用零补齐。

下图演示了如何将二进制整数 10110101011100 转换为十六进制：

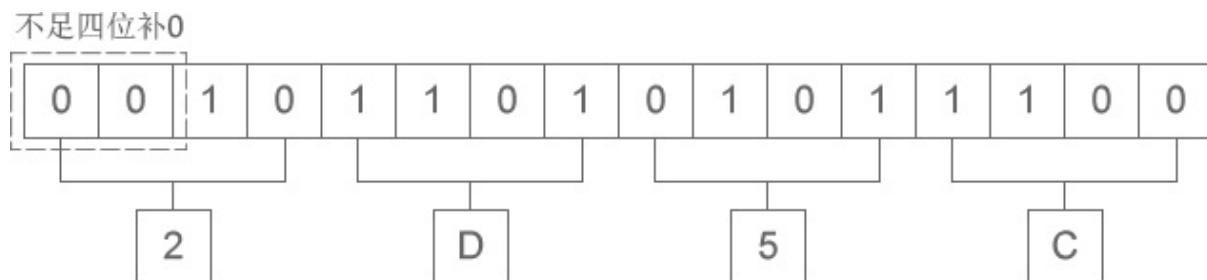
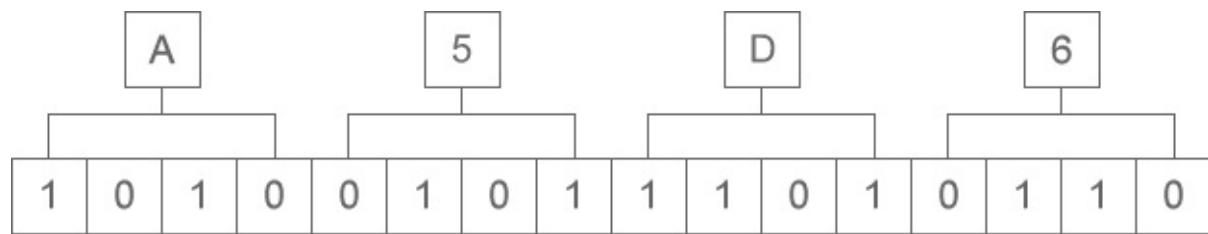


图 - 二进制整数 10110101011100 转换为十六进制

从图中可以看出，二进制整数 10 1101 0101 1100 转换为十六进制的结果为 2D5C。

十六进制整数转换为二进制整数时，思路是相反的，每一位十六进制数字转换为四位二进制数字，运算的顺序也是从低位向高位依次进行。下图演示了如何将十六进制整数 A5D6 转换为二进制：



从图中可以看出，十六进制整数 A5D6 转换为二进制的结果为 1010 0101 1101 0110。

- 在 linux 下使用 adb 刷机
  - 使用adb fastboot 线刷小米平板4Plus
  - 通过 fastboot 安装 Recovery
  - 通过 Recovery 安装魔趣ROM

## 在 linux 下使用 adb 刷机

### 使用adb fastboot 线刷小米平板4Plus

#### 1. 安装adb fastboot工具

```
sudo apt install adb fastboot
```

#### 2. 通过USB将您的设备连接到电脑，并验证手机连接

```
adb devices
```

#### 3. 重启到fastboot模式

```
adb reboot-bootloader
```

#### 4. 下载官方线刷包后解压[链接](#)

```
tar zxvf clover_images_V10.3.2.0.0DJCNXM_20190515.0000.00_8.1_cn_4388d99e5e.tgz
```

#### 5. 进入解压后的目录

```
cd clover_images_V10.3.2.0.0DJCNXM_20190515.0000.00_8.1_cn
```

#### 6. 开始刷机

- 修改为可执行 `chmod +x flash_all.sh`
- fastboot模式下再次验证手机连接 `fastboot devices`
- 执行刷机脚本<sup>1</sup> `sudo sh flash_all.sh`

## 通过 fastboot 安装 Recovery

#### 1. 下载 Recovery - 例如 [TWRP](#)。

- 您可以前往[魔趣下载站](#)的设备页面，点击**Recovery 下载**。
- 如果没有找到适用于您设备的 Recovery，请借助互联网搜索设备维护者或资深玩家发布的教程。

#### 2. 通过USB将您的设备连接到电脑。

#### 3. 在电脑上打开命令提示符（Windows）或终端（Linux 或 macOS）并输入：

```
adb reboot bootloader
```

您也可以通过组合键启动 fastboot 模式：

- 关闭设备后，按住音量调低 + 电源键，直到屏幕上出现“FASTBOOT”字样，然后松开。

#### 4. 一旦设备处于 fastboot 模式，请通过键入以下内容验证您的 PC 是否找到它：

```
fastboot devices
```

5. 将 Recovery 刷入到您的设备。

```
fastboot flash recovery twrp-x.x.x-x-x.img
```

6. 现在进入 Recovery 模式以验证安装：

- 关闭设备后，按住音量调高 + 电源键，直到进入 Recovery 模式，然后松开。

## 通过 Recovery 安装魔趣ROM

1. 下载你想要安装的魔趣ROM包。

- 可选项，下载第三方扩展包，例如 [OpenGapps](#)

2. 如果您尚未进入 Recovery 模式，请重启到 Recovery 模式。

- 关闭设备后，按住音量调高 + 电源键，直到进入 Recovery 模式，然后松开。

1. 该脚本删除所有数据，可以选择其他sh结尾的脚本，根据名字可以猜到大概意思 ↵

- 树莓派使用记录
  - 下载
  - 校验包，解压
  - **xz**烧录命令(该命令尝试失败)
  - **img**格式镜像烧录命令如下（亲测成功）
  - 卸载软件
  - 基础命令
  - **aptitude**

## 树莓派使用记录

### 下载

选择自己要安装的镜像下载

### 校验包，解压

```
shasum 2013-09-25-wheezy-raspbian.zip  
unzip 2013-09-25-wheezy-raspbian.zip
```

查看当前哪些设备已经挂载, `df -h` , 插入u盘或sd卡再执行一次 为了防止在写入镜像的时候有其他读取或写入, 我们需要卸载设备。两个分区都要卸载。

```
umount /dev/sdb1  
umount /dev/sdb2
```

### xz烧录命令(该命令尝试失败)

```
sudo xz -cd kali-2017.3-rpi3-nexmon.img.xz> /dev/sdb
```

查看烧录进度 `sudo pkill -USR1 -n -x xz`

### img格式镜像烧录命令如下（亲测成功）

```
sudo dd bs=4M if=2013-09-25-wheezy-raspbian.img of=/dev/sdb
```

查看烧录进度 `sudo pkill -USR1 -n -x dd`

### 卸载软件

#### 1. 卸载但不删除配置

```
apt-get remove packagename
```

#### 2. 卸载并删除配置

```
apt-get purge packagename
```

## 基础命令

[原链接](#)

安装软件 apt-get install softname1 softname2 softname3.....

卸载软件 apt-get remove softname1 softname2 softname3.....

卸载并清除配置 apt-get remove --purge softname1

更新软件信息数据库 apt-get update

进行系统升级 apt-get upgrade

搜索软件包 apt-cache search softname1 softname2 softname3.....

安装deb软件包 dpkg -i xxx.deb

删除软件包 dpkg -r xxx.deb

连同配置文件一起删除 dpkg -r --purge xxx.deb

查看软件包信息 dpkg -info xxx.deb

查看文件拷贝详情 dpkg -L xxx.deb

查看系统中已安装软件包信息 dpkg -l

重新配置软件包 dpkg-reconfigure xxx

清除所有已删除包的残余配置文件

```
dpkg -l |grep ^rc|awk '{print $2}' |sudo xargs dpkg -P
```

dpkg安装的可以用apt卸载，反之亦可。

## aptitude

aptitude update 更新可用的包列表

aptitude upgrade 升级可用的包

aptitude dist-upgrade 将系统升级到新的发行版

aptitude install pkgname 安装包

aptitude remove pkgname 删除包

aptitude purge pkgname 删除包及其配置文件

aptitude search string 搜索包

aptitude show pkgname 显示包的详细信息

aptitude clean 删除下载的包文件

aptitude autoclean 仅删除过期的包文件

- 个人笔记
  - Jenkins
  - Github加速
  - git
  - 网址导航

## 个人笔记

### Jenkins

1. 插件下载地址
  - hpi插件地址
  - Theme插件
2. 页面自定义 修改jar包 jenkins/WEB-INF/lib/jenkins-core-1.651.3.jar 文件中的 lib/layout/layout.jelly
3. 手机火狐插件 <https://addons.mozilla.org/zh-CN/android/addon/tampermonkey/>

### Github加速

4. 油猴脚本工具
5. 加速访问脚本地址: <https://greasyfork.org/zh-CN/scripts/397419>
6. 加速下载脚本地址: <https://greasyfork.org/scripts/412245>
7. Github 增强 - 高速下载: <https://greasyfork.org/zh-CN/scripts/412245-github-%E5%A2%9E%E5%BC%BA-%E9%AB%98%E9%80%9F%E4%B8%8B%E8%BD%BD>
8. 一键获取Github文件加速下载地址（浏览器扩展）
9. 扩展官网: <https://fhefh2015.github.io/Fast-GitHub/>
10. Chrome安装地址: <https://chrome.google.com/webstore/detail/mfnkflidjnladnkldfonnaicljppahpg>
11. 国内可访问Chrome安装地址1: <https://chrome.zzzmh.cn/info?token=mfnkflidjnladnkldfonnaicljppahpg>
12. 国内可访问Chrome安装地址2: <https://www.extfans.com/productivity/mfnkflidjnladnkldfonnaicljppahpg/>
13. Edge安装地址: <https://microsoftedge.microsoft.com/addons/detail/ljcefikaahacpphaioldelefadpmdp>

### git

让Git显示颜色

```
git config --global color.ui true
```

git设置别名

```
git config --global alias.lg "log --color --graph --pretty=format:'%Cred%h%Creset -%C(yellow)%d%Creset %s %Cgreen(%cr) %C(bold blue)<%an>%Creset' --abbrev-commit"
```

git图形化工具sourcetree

### 网址导航

- git
  - [github](#)
  - [gitee](#)
  - [gitlab](#)
  - [coding](#)
- 博客论坛
  - [csdn](#)
  - [开源中国](#)
  - [掘金](#)
  - [V2EX](#)
  - [segmentfault](#)
  - [语雀](#)
- 部署
  - [glitch](#)
  - [vercel](#)
- 邮箱
  - [联通邮箱](#)
  - [163邮箱](#)
  - [126邮箱](#)
  - [qq邮箱](#)
  - [新浪邮箱](#)
  - [Outlook邮箱](#)
  - [139邮箱](#)
- 主机域名
  - [Freenom域名](#)
  - [Vultr](#)
  - [腾讯云](#)
  - [Cloudflare](#)
- 书签
  - [下载office](#)
  - [油猴脚本](#)
  - [52破解](#)
  - [假52](#)
  - [清华源](#)
  - [MSDN](#)
- 论坛类
  - [知乎](#)
  - [简书](#)
  - [微博](#)
  - [领英](#)
- 云盘
  - [蓝奏云](#)
  - [天翼云盘](#)
  - [百度云](#)
  - [sync](#)
- 下载
  - [迅雷](#)
  - [音乐磁场](#)
  - [磁力搜索](#)
- 通知留言
  - [Leancloud](#)

- [Server酱](#)
- [WxPush](#)
- 账号登录
  - [360账号](#)
  - [火狐](#)
  - [豆瓣](#)
  - [少数派](#)
- 其他
  - [Chrome Ctx下载](#)
  - [Google Play下载](#)
  - [163study](#)
  - [Potplayer播放器](#)
  - [微pe工具箱](#)