

# C2 基本命令和开发工具

LINUX常用命令
命令格式
文件操作命令
文件处理命令
文件查看命令
权限管理命令
文件搜索命令
网络管理与通信命令
压缩解压命令
帮助命令
LINUX常用工具
Vi编辑器
命令模式
插入模式
底行模式
GCC编辑器
编译过程
文件编译
GDB调试工具

## LINUX常用命令

### 命令格式



## 2.1.1 命令格式



### 命令格式

Linux命令遵循一定的基本格式: **Command [options] [arguments]**

示例:



yx.boxuegu.com

## 文件操作命令

### 文件处理命令

#### ls

ls -a 查看包括隐藏文件在内的所有文件

```
[root@izwz94qplgxj28kn3hpn4hz ~]# ls -a
.  .bash_history  .bash_profile  .bashrc.swp  .cshrc  linux_learning  .oracle_jre_usage  .pydistutils.cfg  sa_recovery.log.bak  .tcshrc
.. .bash_logout  .bashrc        .cache       env.txt       .mysql_history  .pip               sa_recovery.log    .ssh              .viminfo
```

ls -l 以详细信息列出文件 可指定文件名

```
[root@izwz94qplgxj28kn3hpn4hz ~]# ls -l
total 1540
-r-x----- 1 root root    302 Apr 11 09:21 env.txt
drwxr-xr-x 2 root root   4096 Jun 29 14:08 linux_learning
-rw-r--r-- 1 root root 558161 Jul  3 09:25 sa_recovery.log
-rw-r--r-- 1 root root 1000071 Jun 15 06:50 sa_recovery.log.bak
```

```

[root@izwz94qplgxj28kn3hpn4hz ~]# ls -l linux_learning
total 120
-rwxr-xr-x 1 root root 8776 May 30 08:48 msgrcv
-rw-r--r-- 1 root root 648 May 30 08:48 msgrcv.c
-rwxr-xr-x 1 root root 8920 May 30 08:45 msgsend
-rw-r--r-- 1 root root 604 May 30 08:58 msgsend.c
-rwxr-xr-x 1 root root 8896 Jun 6 09:08 pthread_cancel
-rw-r--r-- 1 root root 395 Jun 6 09:08 pthread_cancel.c
-rwxr-xr-x 1 root root 8744 Jun 6 08:42 pthread_change_var
-rw-r--r-- 1 root root 339 Jun 6 08:42 pthread_change_var.c
-rwxr-xr-x 1 root root 8832 Jun 6 08:51 pthread_exit
-rw-r--r-- 1 root root 550 Jun 6 08:51 pthread_exit.c
-rwxr-xr-x 1 root root 8864 Jun 6 09:19 pthread_share
-rw-r--r-- 1 root root 456 Jun 6 09:19 pthread_share.c
-rwxr-xr-x 1 root root 8976 Jun 6 08:34 pthread_show_id
-rw-r--r-- 1 root root 448 Jun 6 08:31 pthread_show_id.c
-rw-r--r-- 1 root root 695 May 30 09:19 shm_w.c
-rw-r--r-- 1 root root 504 Jun 6 09:35 tcpserver.c
[root@izwz94qplgxj28kn3hpn4hz ~]# ls -l env.txt
-r-x----- 1 root root 302 Apr 11 09:21 env.txt
[root@izwz94qplgxj28kn3hpn4hz ~]#

```

## cd

cd ./xxx 切换到相对路径

cd /xxx 切换到绝对路径

cd ~ 切换到当前用户的家目录

## pwd

## touch

## mkdir

示例: `mkdir -pv ./testmkdir/itheima`



```

itheima@localhost:~
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
[itheima@localhost ~]$ mkdir -pv ./testmkdir/itheima
mkdir: 已创建目录 './testmkdir'
mkdir: 已创建目录 './testmkdir/itheima'
[itheima@localhost ~]$

```

说明:

①第2、3行信息为使用-v选项打印的创建过程。

②因使用-p选项, 所以会创建路径中不存在的目录。

## cp

-r 递归复制目录下的文件和子目录

示例: `cp -R dir ./testmkdir/itheima`

```
itheima@localhost:~  
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)  
[itheima@localhost ~]$ ls ./testmkdir/itheima  
[itheima@localhost ~]$ cp -R dir ./testmkdir/itheima  
[itheima@localhost ~]$ ls ./testmkdir/itheima  
dir  
[itheima@localhost ~]$
```

**mv**

移动文件或目录

**rm**

删除目录中的文件或目录

选项	说明
-f	强制删除文件或目录
-rf	选项-r与-f结合，删除目录中所有文件和子目录，并且不——确认
-i	在删除文件或目录时对要删除的内容逐一进行确认 (y/n)

**rmdir**

删除目录

```
itheima@localhost:~  
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)  
[itheima@localhost ~]$ mkdir -p ./itheima/itcast  
[itheima@localhost ~]$ rmdir -p itheima/itcast  
[itheima@localhost ~]$ rmdir itheima  
rmdir: 删除 "itheima" 失败: 没有那个文件或目录  
[itheima@localhost ~]$  
[itheima@localhost ~]$
```

文件查看命令

**cat**

打印文件内容到输出设备

**more**

分页显示文件内容

快捷键	说明
f/Space	显示下一页
Enter	显示下一行
q/Q	退出

head

查看文件的前n行

```
itheima@localhost:~  
文件(F) 编辑(E) 查看(V) 搜索 (S) 终端(T) 帮助(H)  
[ itheima@localhost ~]$ head -4 test.c  
#include <stdio.h>  
#include <stdlib.h>  
  
extern char **environ;  
[ itheima@localhost ~]$  
[ itheima@localhost ~]$
```

tail

查看文件的后n行

```
itheima@localhost:~  
文件(F) 编辑(E) 查看(V) 搜索 (S) 终端(T) 帮助(H)  
[ itheima@localhost ~]$ tail -4 test.c  
    unsetenv("MONKEY");  
    printf("MONKEY=%s\n", getenv("MONKEY"));  
    return 0;  
}  
[ itheima@localhost ~]$
```

权限管理命令

拥有者，同组用户，其他组用户，全部用户

权限	对应字符	文件	目录
读权限	r	可查看文件内容	可以列出目录中的内容
写权限	w	可修改文件内容	可以在目录中创建、删除文件
执行权限	x	可执行该文件	可以进入目录

su 将用户切换为root



方法如下:

A terminal window titled 'itheima@localhost:~' showing the execution of the 'su' command. The prompt changes from '[itheima@localhost ~]\$' to '[root@localhost itcast]#'. Annotations with blue boxes and arrows point to specific parts: '输入密码' (Enter password) points to the 'su' command; '执行命令' (Execute command) points to the prompt change; '退出' (Exit) points to the 'exit' command; '提示符号发生改变' (Prompt symbol changes) points to the change from '\$' to '#'. The terminal shows the command sequence: [itheima@localhost ~]\$ su, 密码: (password), [root@localhost itcast]# exit, exit, [itheima@localhost ~]\$.

A terminal window titled '[Lisi@redhat9]\$' showing the output of the 'ls -la' command. The output lists files and directories with their permissions, owner, group, size, date, and name. Annotations with yellow boxes and lines point to specific parts: '属组名' (Group name) points to the group column; '文件名' (File name) points to the file name column; '修改日期' (Modification date) points to the date column; '大小按字节计算' (Size in bytes) points to the size column; '属主名' (Owner name) points to the owner column; '链接数' (Link count) points to the link count column; '文件类型及访问权限' (File type and permissions) points to the permission column. The output is as follows:  
total 44  
drwx----- 4 lisi lisi 4096 Oct 8 .  
drwxr-xr-x 4 root root 4096 Sep 12 ..  
drwxrwxr-x 2 lisi lisi 4096 Oct 8 aa  
-rw-rw-r-- 1 lisi lisi 38 Oct 8 aa.txt  
-rw----- 1 lisi lisi 463 Oct 9 .bash\_history  
-rw-r--r-- 1 lisi lisi 24 Sep 12 .bash\_logout  
-rw-r--r-- 1 lisi lisi 191 Sep 12 .bash\_profile  
-rw-r--r-- 1 lisi lisi 124 Sep 12 .bashrc  
-rw-r--r-- 1 lisi lisi 118 Sep 12 .gtkrc  
[lisi@redhat9]\$

1-所属用户的权限 2-组群权限 3-其他用户的权限

-rwxrwxrwx为权限最开放的777

## chmod

变更文件或目录的权限

示例: **chmod u-x, g-x file**

A terminal window titled 'itheima@localhost:/home/itcast' showing the execution of the 'chmod' command. The prompt is '[root@localhost itcast]#'. The command sequence is: [root@localhost itcast]# ls -l file, [root@localhost itcast]# chmod u-x, g-x file, [root@localhost itcast]# ls -l file, [root@localhost itcast]#. The output of the first 'ls -l file' command is: -rwxrwxr-x. 1 itheima 501 7281 11月 14 17:35 file. The output of the second 'ls -l file' command is: -rw-rw-r-x. 1 itheima 501 7281 11月 14 17:35 file.

## chown

变更文件或目录的所有者

## 示例: `chown` root file

```
itheima@localhost:/home/itcast
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
[root@localhost itcast]# ls -l file
-rw-rw-r-x. 1 itheima 501 7281 11月 14 17:35 file
[root@localhost itcast]# chown root file
[root@localhost itcast]# ls -l file
-rw-rw-r-x. 1 root 501 7281 11月 14 17:35 file
[root@localhost itcast]#
```

## chgrp

变更文件或目录的所有者

## 示例: `chgrp` root itheima

```
itheima@localhost:/home/itcast
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
[root@localhost itcast]# ls -l itheima
-rw-r--r--. 1 itheima itcast 0 3月 30 15:04 itheima
[root@localhost itcast]# chgrp root itheima
[root@localhost itcast]# ls -l itheima
-rw-r--r--. 1 itheima root 0 3月 30 15:04 itheima
[root@localhost itcast]#
```

## 文件搜索命令

### which

查看命令所在路径

## 示例: `which` ls

```
itheima@localhost:~
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
[itheima@localhost ~]$ which ls
alias ls='ls --color=auto'
/bin/ls
[itheima@localhost ~]$
```

### find

接触搜索关键字查找文件或目录

find 搜索路径 [选项] 搜索关键字

选项	说明
-name	根据文件名查找
-size	根据文件大小查找
-user	根据文件所有者查找

### locate

借助搜索关键字查找文件或目录

locate [选项] 搜索关键字

locate速度远胜find

find搜索整个目录，locate搜索数据库/var/lib/locatedb

即便文件存在，数据库中没有记录，locate便搜索不到

## grep

在文件中搜索与字符串匹配的行并输出

grep 指定字符 源文件

示例1: `grep itheima /etc/passwd`



```
itheima@localhost:~  
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)  
[itheima@localhost ~]$ grep itheima /etc/passwd  
itheima:x:500:500:itcast:/home/itcast:/bin/bash  
[itheima@localhost ~]$
```

## 网络管理与通信命令

### ifconfig

### netstat

### ping

### write

使当前用户向另一个用户发送信息

### wall

使当前用户向所有用户发送信息

## 压缩解压命令

### gzip/gunzip

.gz 压缩后不保存源文件 若同时列出多个文件，则每个文件会被单独压缩

### zip/unzip



.zip 压缩后保存源文件

**bzip2/bunzip2**

.bz2 使用选项-k时保留源文件

**tar**

打包多个文件或目录，通常与压缩命令一起使用

选项	说明
-c	产生.tar打包文件
-v	打包时显示详细信息
-f	指定压缩后的文件名
-z	打包，同时通过gzip指令压缩备份文件，压缩后格式为.tar.gz
-x	从打包文件中还原文件。

例子: <https://www.cnblogs.com/newcaoguo/p/5896975.html>

```
[root@izwz94qplgj28kn3hpn4hz linux_learning]# tar -cf allc.tar *.c
[root@izwz94qplgj28kn3hpn4hz linux_learning]# ls
allc.tar  msgrcv.c  msgsend.c  pthread_cancel.c  pthread_change_var.c  pthread_exit.c  pthread_share.c  pthread_show_id.c  tcpserver.c
msgrcv   msgsend  pthread_cancel  pthread_change_var  pthread_exit  pthread_share  pthread_show_id  shm_w.c
```

**帮助命令**

**man**

选项	说明
-a	在所有的man帮助手册中搜索
-p	指定内容时，使用分页程序
-M	指定man手册搜索的路径

**info**

也是查看帮助手册，编写方式不一样

**whatis**

查询命令的功能

**whoami**

打印当前有效的用户名称

**LINUX常用工具**

**Vi编辑器**

命令模式

级别	操作符	说明
字符级	“左键”或字母“h”	使光标向字符的左边移动
	“右键”或字母“l”	使光标向字符的右边移动
行级	“上键”或字母“k”	使光标移动到上一行
	“下键”或字母“j”	使光标移动到下一行
	符号“\$”	使光标移动到当前行尾
	数字“0”	使光标移动到当前行首
单词级	字母“w”	使光标移动到下一个单词的首字母
	字母“e”	使光标移动到本单词的尾字母
	字母“b”	使光标移动到本单词的首字母

级别	操作符	说明
段落级	符号“}”	使光标移至段落结尾
	符号“{”	使光标移至段落开头
屏幕级	字母“H”	使光标移至屏幕首部
	字母“L”	使光标移至屏幕尾部
文档级	字母“G”	使光标移至文档尾行
	n+G	使光标移至文档的第n行

操作符	说明
字母“x”	删除光标所在的单个字符
字母“dd”	删除光标所在的当前行
n+dd	删除包括光标所在行的后边n行内容
d+\$	删除光标位置到行尾的所有内容

操作符	说明
字母“yy”	复制光标当前所在行
n+yy	复制包括光标所在行后的n行内容
y+e	从光标所在位置开始复制直到当前单词结尾
y+\$	从光标所在位置开始复制直到当前行结尾
y+{	从当前段落开始的位置复制到光标所在位置
p	将复制的内容粘贴到光标所在位置

插入模式

底行模式

`:set nu` 设置行号

`:set nonu` 取消行号

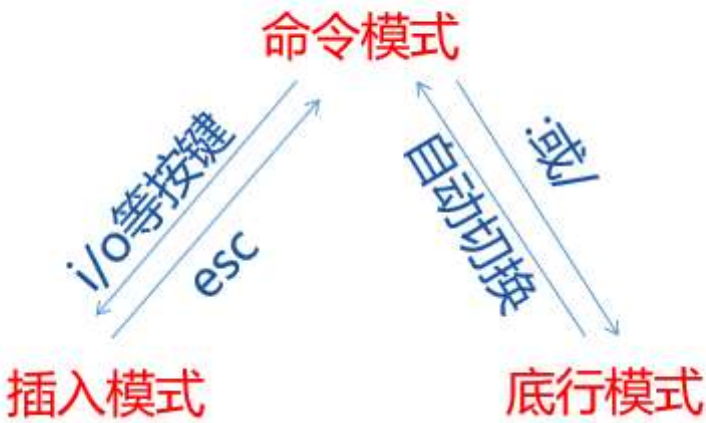
**:n** 使光标移动到第n行

**:/xx** 在文件中查找“xx”，若查找结果不为空，可以用“n”查找下一个，用“N”查找上一个

操作符	说明
:s/被替换内容/替换内容/	替换光标所在行的第一个目标
:s/被替换内容/替换内容/g	替换光标所在行的全部目标
%s/被替换内容/替换内容/g	替换整个文档中的全部目标
%s/被替换内容/替换内容/gc	替换整个文档中的全部目标，且每替换一个内容都有相应的提示

操作符	说明
:q	退出vi编辑器
:w	保存编辑后的内容
:wq	保存并退出vi编辑器
:q!	强行退出vi编辑器，不保存对文件的修改
:w!	对于没有修改权限的用户强行保存对文件的修改，并且修改后文件的所有者和所属组都有相应的变化
:wq!	强行保存文件并退出vi编辑器

Vi编辑器的三种模式之间可进行转换：



命令模式切换至插入模式：

操作符	说明
字母“a”	光标向后移动一位进入编辑模式
字母“s”	删除光标所在字母进入编辑模式
字母“o”	在当前行之下新起一行进入编辑模式
字母“A”	光标移动到当前行末尾进入编辑模式
字母“I”	光标移动到当前行行首进入编辑模式
字母“S”	删除光标所在行进入编辑模式
字母“O”	在当前行之上新起一行进入编辑模式

在命令行中对vi编辑器进行的设置，只对本次操作有效，若想长久保存vi设置，可通过对用户家目录中的.vimrc文件进行编辑实现。常用设置如表所示。

设置	说明
set number	设置行号
set autoindent	自动对齐
set smartindent	智能对齐
set showmatch	括号匹配
set tabstop=4	使用tab键时为4个空格
set mouse=a	鼠标支持
set cindent	使用C语言格式对齐

## GCC编辑器

由GUN开发的编程语言编译器，现可以编译C语言、Java、OC等多种变成语言

### 编译过程

#### 预处理 -e

- ❑ 删除代码中的“#define”，展开所有宏定义；
- ❑ 处理条件编译指令，如#if、#ifdef、#undef等；
- ❑ 将由“#include”包含的文件插入到预编译指令对应的位置，若文件中包含其它文件，同样进行替换；
- ❑ 删除代码中的注释；
- ❑ 添加行号和文件标识；
- ❑ 保留#pragma编译器指令。

#### 编译 -s

在编译阶段，GCC会对经过预处理的文件进行语法、词法和语义分析，确定代码实际要做的工作，若检查无误，则生成相应的汇编代码文件。

#### 汇编 -o

汇编过程将编译后生成的汇编代码转换为机器可以执行的命令，即二进制指令，每一个汇编语句几乎都会对应一条机器指令。

#### 链接

链接过程是组装各个目标文件的过程，在这个过程中会解决符号依赖和库依赖关系，最终生成可执行文件。

### 文件编译

#### 单文件编译

```
$ gcc hello.c
```

#### 多文件编译

```
$ gcc _main.c _add.c -o _main
```

## GDB调试工具



GDB可以逐条执行程序、操控程序的运行，并且随时可以查看程序中所有的内部状态，如各变量的值、传给函数的参数、当前执行的语句位置等，藉此判断代码中的逻辑错误。

## (1) 在代码中加入调试信息，并调试程序

```
$ gcc gdbtest.c -o app -g
```

```
(gdb) gdb app
```

## (2) 列出程序代码，并设置断点

```
list 行号
```

```
b 行号
```

```
b 22 if i = 5          #在22行设定带条件的断点
```

```
info b                #查看断点
```

```
disable Num           #取消断点
```