

快速使用云环境

: Tags	如何在云主机上编程
Person	叶露宿
Project	开发环境配置
科目	Linux操作系统
语言	C



本文档地址 <http://suyelu.pro/linux-coding>

也可以直接访问 <http://suyelu.pro> 进入主页后自行查找本文档。

目录

[使用到的命令清单](#)

[Tabby使用的一些Tips](#)

[文件及目录相关命令](#)

[系统及相关命令](#)

[vim的简单使用](#)

[如何进入及退出vim](#)

[vim的四种模式](#)

[如何在文档中快速跳转](#)

[如何进行编辑](#)

[如何删除](#)

[如何复制及粘贴](#)

[撤销与恢复](#)

[搜索](#)

[如何在云主机上编程](#)

[Man手册的使用](#)

[man手册汉化](#)

[彩色man手册](#)

使用到的命令清单

文件及目录相关

命令名	作用
ls (list)	列出目录下的内容
cd (change directory)	切换目录
mkdir (make directory)	创建目录
rm (remove)	删除文件或目录
cat (concatenate)	获取文件内容
pwd	

系统及其他

命令名	作用
exit	退出连接
reboot	重启系统
apt update	更新软件仓库缓存，为更新和下载软件做准备
apt install xxx	下载xxx软件

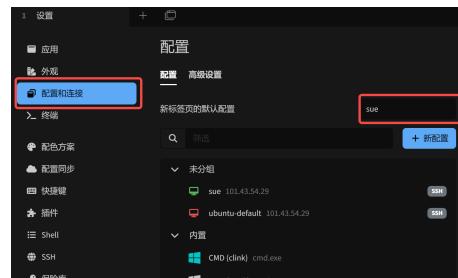
编程相关

命令名	作用
gcc	编译源代码
vim	打开编辑器
./a.out	执行可执行程序
man xxx	查看xxx的手册

Tabby使用的一些Tips

1. 如何让tabby自动连接到云主机

▼ 图



2. 如何快速在多个终端之间切换

- ctrl + shift + t 打开新终端
- ctrl + TAB 切换到下一个终端
- Ctrl + shift + left 切换到左侧终端
- ctrl + shift + right 切换到右侧终端

3. 如何将多个终端放在一个窗口，并在之间切换

将鼠标移动至终端标题序号位置，待鼠标样式切换为 ，拖拽到其他窗口的边缘四个位置，可将其放置在此窗口的相应位置

将其重新拖拽到上方标题栏，可重新恢复为独立的终端

最大化活动窗格 (pane-maximize)	Ctrl-Alt-Enter ×
聚焦上面的窗格 (pane-nav-up)	Ctrl-Alt-Up ×
聚焦下面的窗格 (pane-nav-down)	Ctrl-Alt-Down ×
聚焦左方的窗格 (pane-nav-left)	Ctrl-Alt-Left ×
聚焦右方的窗格 (pane-nav-right)	Ctrl-Alt-Right ×
聚焦上一个窗格 (pane-nav-previous)	Ctrl-Alt-[×
聚焦下一个窗格 (pane-nav-next)	Ctrl-Alt-] ×

4. 如何在窗口中批量执行命令 `ctrl + shift + I` 、 `ctrl + shift + alt + I`

5. 如何传输文件 `sz` `rz` `sftp`

文件及目录相关命令

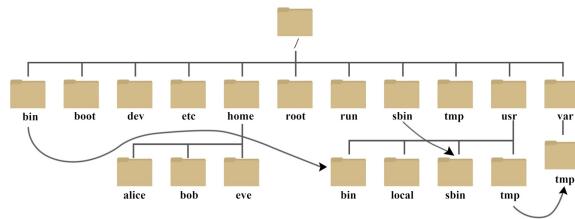
1. 几个特殊的目录介绍：`.` `..` `~` `-` `/`

- a. `.` 当前目录
- b. `..` 上级目录，也就是父目录
- c. `~` 当前用户的家目录
- d. `-` 上次工作目录
- e. `/` 根目录

2. 命令的基本构成

- 命令主体：命令名（谓语动词）
- 命令的承受者：参数（宾语）
- 命令执行的方式：选项（状语）

Linux目录树



如何判断命令执行成功与否？

每个命令都有自己的返回值，正如我们写程序的时候的 `return 0;` 返回的是成功的 `0`

- 返回值为 `0`：命令执行成功
- 返回这 `不为0`：命令执行失败

```

sue@IsoftStone ~ % ls
sue@IsoftStone ~ % [0] → [0]

```

3. `ls` - 列出目录下的内容

```

ls #ls不加任何参数，则使用缺省参数`.`，也就是当前目录
ls / #将列出根目录下的所有文件及目录
ls -a #列出当前目录下的所有文件，包括隐藏文件
ls -l #以长列表的形式展示文件及目录的详细信息
ls -al #以长列表的形式展示全部文件及目录的详细信息

```

4. mkdir - 创建目录

```
mkdir new_dir #在当前位置创建一个new_dir的目录  
mkdir /tmp/new_dir_1 #在/tmp下创建new_dir_1的目录  
mkdir -p /tmp/123/456/789 #创建/var/123/456下的789目录，但是如果所有上级目录中，有不存在的，则一并创建
```

5. 绝对路径和相对路径

绝对路径和相对路径是指在命令行中指定文件或目录时所使用的两种不同的路径表示方式。

- 绝对路径：指从根目录开始的完整路径。如：`/home/user/file.txt`
- 相对路径：指相对于当前工作目录的路径。如：`./file.txt` 表示当前目录下的 file.txt。

在使用命令行时，大多数情况下建议使用相对路径。因为相对路径更加简洁，而且更方便移植，不需要考虑在不同机器上的根目录位置不同，而绝对路径则需要考虑这一点。

5. cd - 切换当前工作目录

```
cd #cd不加参数，则会回到自己的家目录  
cd - # 回到上次工作目录  
cd .. # 切换到父目录  
cd / #切换到根目录  
cd ~ #切换到家目录  
pwd #打印当前工作目录
```

6. rm - 删除文件或目录

```
rm file.txt # 删除文件file.txt  
rm -r dir # 删除目录dir及其下的所有文件和目录  
rm -f file.txt # 强制删除文件file.txt，不提示确认
```

7. cat - 获取文件内容

```
cat file.txt # 将file.txt的内容输出到屏幕上  
cat file1.txt file2.txt # 将file1.txt和file2.txt的内容按照顺序输出到屏幕上
```

系统及相关命令

1. exit - 退出连接

```
exit # 退出连接
```

2. reboot - 重启系统

```
reboot # 重启系统
```

3. `apt update` - 更新软件仓库缓存，为更新和下载软件做准备

```
apt update # 更新软件仓库缓存
```

4. `apt install xxx` - 下载xxx软件

```
apt install xxx # 下载xxx软件
```

vim的简单使用

如何进入及退出vim

1. `:w` Write
2. `:wq` Write and Quit
3. `:q` Quit
4. `:q!` Quit force

vim的四种模式

1. 普通模式（默认进入模式）

在普通模式下，可以使用多个命令对文本进行处理，比如复制、剪切、粘贴、插入等。此外，还可以使用 `:command` 命令执行特定的操作。

2. 命令模式

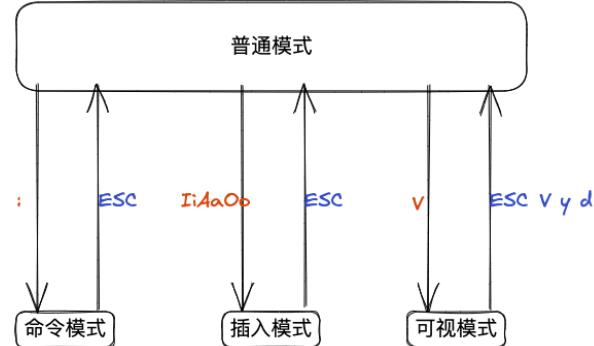
在普通模式下输入 `:` 进入命令模式，可以执行一些高级命令，比如查找、替换、保存、退出等操作。

3. 插入模式

插入模式是用来输入文本的模式。在普通模式下，按下 `i` 或 `a` 等命令即可进入插入模式。在插入模式下，可以输入文本，但不能执行命令。要返回普通模式，需要按下 `ESC` 键。

4. 可视模式

可视模式用于选择文本。在普通模式下，按下 `v` 或 `V` 即可进入可视模式。在可视模式下，可以使用 `h`、`j`、`k`、`l` 等命令选择文本，还可以使用 `y`、`d` 等命令复制、剪切选中的文本。



如何在文档中快速跳转

1. 普通模式和可视模式下

- a. `gg` 到达文件首行
- b. `G` 到达文件末行
- c. `5gg` 到达第五行
- d. `5G` 到达第五行

2. 插入模式下

- a. 使用方向键

如何进行编辑

在 `vim` 中，编辑器普通模式、插入模式和可视模式下，可以执行不同的编辑操作。

1. 普通模式

在普通模式下，可以使用 `d`, `x` 等命令对文档内容进行删除，`y` 来复制，`p` 来粘贴。

此外，还可以使用 `:command` 命令执行特定的操作。

2. 插入模式

插入模式是用来输入文本的模式。在普通模式下，按下 `i` 或 `a` 等命令即可进入插入模式。在插入模式下，可以输入文本，但不能执行命令。要返回普通模式，需要按下 `ESC` 键。

3. 可视模式

可视模式用于选择文本。在普通模式下，按下 `v` 或 `V` 即可进入可视模式。在可视模式下，可以使用 `h`、`j`、`k`、`l` 等命令选择文本，还可以使用 `y`、`d` 等命令复制、剪切选中的文本。

如何删除

1. 普通模式下

- a. `dd` 删除一行
- b. `5dd` 删除5行
- c. `d5G` 删除到第5行
- d. `dw` 删除当前单词
- e. `d$` 删除到行末
- f. `d^` 删除到行首
- g. `x` 删除当前字符

2. 插入模式下

- a. 使用 `backspace` 删除即可

3. 可视模式下

- a. 选中后使用 `d` 可删除

如何复制及粘贴

在 `vim` 中，仅可在普通或可视模式下进行复制和粘贴

1. 普通模式

- a. `yy` 复制一行
- b. `5yy` 复制5行
- c. `y5G` 复制到第5行
- d. `p` 粘贴当当前光标之后
- e. `P` 粘贴到之前

2. 可视模式

- a. 选中后使用 `y` 复制
- b. 使用 `p`、`P` 来粘贴

撤销与恢复

在普通模式下，可以使用 `u` 命令撤销（`undo`）上一步操作，`ctrl + r` 来重做（`redo`）。

搜索

在普通模式下，可以使用 `/` 命令搜索文本。输入要搜索的文本，然后按下 `Enter` 键即可，按 `n` 查找下一个（`next`），`N` 查找上一个。



如果想更详细的学习 `vim`，建议你直接使用 `vim` 官方提供的教学文档，在终端执行 `vimtutor` 试试吧~ 什么？这个文档是英文的！那么试一下这个命令吧：`vimtutor zh`

如何在云主机上编程

1. 使用 `vim xxx.c` 命令可以打创建 `.c` 文件,编写代码

```
1 //include <stdio.h>
2 int main () {
3     printf("Hello World");
4     return 0;
5 }
```

2. 使用 `gcc xxx.c -o xxx` 对文件进行编译

```
user@laptop: ~ % vim HelloWorld.c
user@laptop: ~ % gcc HelloWorld.c -o Hello.exe
user@laptop: ~ % ls
HelloWorld.c          Hello.exe
user@laptop: ~ % ./Hello.exe
Hello World
user@laptop: ~ %
```

3. 执行可执行程序

```
user@laptop: ~ % ./HelloWorld
Hello World
user@laptop: ~ %
```

Man手册的使用

man是Linux下的一个非常重要的命令，用于查看系统或程序的手册。使用man命令可以查看命令的用法、参数、返回值等信息，非常实用。

man命令的格式为：

```
man [选项] [命令或程序或函数]
```

常用的选项有：

- **k**：查询命令的简要说明
- **f**：查询命令的详细说明
- **a**：显示所有与查询项相关的手册页

例如，要查询ls命令的用法，可以使用以下命令：

```
man ls
```

这样就可以查看到ls命令的用法、参数、返回值等信息。

使用man命令需要注意的是，man手册通常会分为多个章节，每个章节都有不同的内容。常用的章节有：

- 第一章：用户命令
- 第二章：系统调用
- 第三章：C库函数
- 第四章：特殊文件
- 第五章：文件格式和约定
- 第六章：游戏和屏保
- 第七章：杂项
- 第八章：系统管理命令

可以使用以下命令查询指定章节的手册页：

```
man [章节号] [命令或程序]
```

例如，要查询scanf的C库函数手册页，可以使用以下命令：

```
man 3 scanf
```

这样就可以查看到ls命令的C库函数手册页了。



在vim中，可以直接使用 shift + K 来查看当前光标所在关键字的Man手册，如果是要加上man手册的章节的话，可以先输入数字，再输入shift + K；
退出内置手册，直接按q即可

man手册汉化

- 要在Ubuntu 20.04上汉化man手册，可以执行以下命令：

```
sudo apt install manpages-zh
```

- 使用 vim ~/.zshrc 打开.zshrc文件，在最后一行之后添加以下内容

```
export LC_ALL=en_US.UTF-8
export LANG=en_US.UTF-8

alias cman="man -M /usr/share/man/zh_CN/"
```

- 使用 cman 命令查看汉化版 MAN 手册

```
source ~/.zshrc
cman ls
```

彩色man手册

- 使用 vim ~/.zshrc 打开zsh配置文件
- 找到plugins开始的一行，填写并修改为以下内容

```
plugins=(git.zsh-syntax-highlighting.colored-man-pages)
```

- 关闭并重新打开连接即可获得彩色版man手册

SCANF(3) Linux Programmer's Manual SCANF(3)

NAME `scanf`, `fscanf`, `sscanf`, `vscanf`, `vsprintf`, `vfscanf` - input format conversion

SYNOPSIS

```
#include <stdio.h>

int scanf(const char *format, ...);
int fscanf(FILE *stream, const char *format, ...);
int sscanf(const char *str, const char *format, ...);

#include <stdarg.h>

int vscanf(const char *format, va_list ap);
int vsprintf(const char *str, const char *format, va_list ap);
int vfscanf(FILE *stream, const char *format, va_list ap);
```

Feature Test Macro Requirements for glibc (see `feature_test_macros(7)`):

```
vscanf(), vsprintf(), vfscanf():
    _ISOC99_SOURCE || _POSIX_C_SOURCE >= 200112L
```

LS(1) General Commands Manual LS(1)

NAME ls, dir, vdir - 列目录内容

摘要 ls [选项] [文件名...]

POSIX 标准选项： [-CFRacdilqrstu1]

GNU 选项 (短格式)：
[-1abcfgiklmnopqrstuvwxyzABCDFGLNQRSUX] [-w cols] [-T cols] [-I pattern]
[-full-time] [--format={long,verbose,commas,across,vertical,single-column}]
[--sort={none,time,size,extension}]
[-time=[atime,access,use,ctime,status]]
[--color=[none,auto,always]] [-help] [--version] [--]

描述 (DESCRIPTION)
程序 1 先列出非目录的文件项，然后是每一个目录中的“可显示”文件。如果，没有选项之外的参数 [注释：即文件名部分为空] 出现，缺省为 “.” (当前目录)。选项 “-d” 使得目录与非目录项同样对待。除非 “-a” 选项出现，文件名以 “.” 开始的文件不属“可显示”文件。

以当前目录为准，每一组文件 (包括非目录文件项，以及每一内含文件的目
Manual page 1(1) line 1 (press h for help or q to quit)