

Bash 行操作

网道 (WangDoc.com) , 互联网文档计划

目录 [隐藏]

1. 简介
2. 光标移动
3. 清除屏幕
4. 编辑操作
5. 自动补全
6. 其他快捷键

1. 简介

Bash 内置了 Readline 库，具有这个库提供的很多“行操作”功能，比如命令的自动补全，可以大大加快操作速度。

这个库默认采用 Emacs 快捷键，也可以改成 Vi 快捷键。

```
$ set -o vi
```

下面的命令可以改回 Emacs 快捷键。

```
$ set -o emacs
```

📄 Bash 脚本教程

- 📄 1. 简介
- 📄 2. 基本语法
- 📄 3. 模式扩展
- 📄 4. 引号和转义
- 📄 5. 变量
- 📄 6. 字符串操作
- 📄 7. 算术运算
- 📄 8. 操作历史
- 📄 9. 行操作
- 📄 10. 目录堆栈
- 📄 11. 脚本入门
- 📄 12. read 命令
- 📄 13. 条件判断
- 📄 14. 循环
- 📄 15. 函数
- 📄 16. 数组
- 📄 17. set 命令, shopt 命令

如果想永久性更改编辑模式（Emacs / Vi），可以将命令写在 `~/.inputrc` 文件，这个文件是 Readline 的配置文件。

```
set editing-mode vi
```

本章介绍的快捷键都属于 Emacs 模式。Vi 模式的快捷键，读者可以参考 Vi 编辑器的教程。

Bash 默认开启这个库，但是允许关闭。

```
$ bash --noediting
```

上面命令中，`--noediting` 参数关闭了 Readline 库，启动的 Bash 就不带有行操作功能。

2. 光标移动

Readline 提供快速移动光标的快捷键。

- `Ctrl + a`：移到行首。
- `Ctrl + b`：向行首移动一个字符，与左箭头作用相同。
- `Ctrl + e`：移到行尾。
- `Ctrl + f`：向行尾移动一个字符，与右箭头作用相同。
- `Alt + f`：移动到当前单词的词尾。
- `Alt + b`：移动到当前单词的词首。

上面快捷键的 Alt 键，也可以用 ESC 键代替。

3. 清除屏幕

`Ctrl + l` 快捷键可以清除屏幕，即将当前行移到屏幕的第一行，与 `clear` 命令作用相同。

📖 18. 脚本除错

📖 19. mktemp 命令, trap 命令

📖 20. 启动环境

📖 21. 命令提示符

🔗 链接

📄 本文源码

📁 代码仓库

📬 反馈

4. 编辑操作

下面的快捷键可以编辑命令行内容。

- `Ctrl + d` : 删除光标位置的字符 (delete) 。
- `Ctrl + w` : 删除光标前面的单词。
- `Ctrl + t` : 光标位置的字符与它前面一位的字符交换位置 (transpose) 。
- `Alt + t` : 光标位置的词与它前面一位的词交换位置 (transpose) 。
- `Alt + l` : 将光标位置至词尾转为小写 (lowercase) 。
- `Alt + u` : 将光标位置至词尾转为大写 (uppercase) 。

使用 `Ctrl + d` 的时候，如果当前行没有任何字符，会导致退出当前 Shell，所以要小心。

剪切和粘贴快捷键如下。

- `Ctrl + k` : 剪切光标位置到行尾的文本。
- `Ctrl + u` : 剪切光标位置到行首的文本。
- `Alt + d` : 剪切光标位置到词尾的文本。
- `Alt + Backspace` : 剪切光标位置到词首的文本。
- `Ctrl + y` : 在光标位置粘贴文本。

同样地，Alt 键可以用 Esc 键代替。

5. 自动补全

命令输入到一半的时候，可以按一下 Tab 键，Readline 会自动补全命令或路径。比如，输入 `cle`，再按下 Tab 键，Bash 会自动将这个命令补全为 `clear`。

如果符合条件的命令或路径有多个，就需要连续按两次 Tab 键，Bash 会提示所有符合条件的命令或路径。

除了命令或路径，Tab 还可以补全其他值。如果一个值以 `$` 开头，则按下 Tab 键会补全变量；如果以 `~` 开头，则补

全用户名；如果以 `@` 开头，则补全主机名（hostname），主机名以列在 `/etc/hosts` 文件里面的主机为准。

自动补全相关的快捷键如下。

- `Tab`：完成自动补全。
- `Alt + ?`：列出可能的补全，与连接两次 `Tab` 键作用相同。
- `Alt + /`：尝试文件路径补全。
- `Ctrl + x /`：先按 `Ctrl + x`，再按 `/`，等同于 `Alt + ?`，列出可能的文件路径补全。
- `Alt + !`：命令补全。
- `Ctrl + x !`：先按 `Ctrl + x`，再按 `!`，等同于 `Alt + !`，命令补全。
- `Alt + ~`：用户名补全。
- `Ctrl + x ~`：先按 `Ctrl + x`，再按 `~`，等同于 `Alt + ~`，用户名补全。
- `Alt + $`：变量名补全。
- `Ctrl + x $`：先按 `Ctrl + x`，再按 `$`，等同于 `Alt + $`，变量名补全。
- `Alt + @`：主机名补全。
- `Ctrl + x @`：先按 `Ctrl + x`，再按 `@`，等同于 `Alt + @`，主机名补全。
- `Alt + *`：在命令行一次性插入所有可能的补全。
- `Alt + Tab`：尝试用 `.bash_history` 里面以前执行命令，进行补全。

上面的 `Alt` 键也可以用 `ESC` 键代替。

6. 其他快捷键

- `Ctrl + j`：等同于回车键（LINEFEED）。
- `Ctrl + m`：等同于回车键（CARRIAGE RETURN）。
- `Ctrl + o`：等同于回车键，并展示操作历史的下一个命令。
- `Ctrl + v`：将下一个输入的特殊字符变成字面量，比如回车变成 `^M`。

- `Ctrl + [` : 等同于 `ESC`。
- `Alt + .` : 插入上一个命令的最后一个词。
- `Alt + _` : 等同于 `Alt + .` 。

上面的 `Alt + .` 快捷键，对于很长的文件路径，有时会非常方便。因为 Unix 命令的最后一个参数通常是文件路径。

```
$ mkdir foo_bar  
$ cd #按下 Alt + .
```

上面例子中，在 `cd` 命令后按下 `Alt + .`，就会自动插入 `foo_bar`。

[🔍 操作历史](#)

[📁 目录堆栈 🔍](#)

本教程采用知识共享 署名-相同方式共享 3.0协议。

分享本文      

联系: contact@wangdoc.com