

Shell overview

Linux 的基本单元：「shell」

「shell」（也被称为终端、控制台或命令行）是一个基于文本的用户界面，通过它把命令发送给机器。在 Linux 中，shell 的默认语言是 bash。与主要在 Windows 内部进行点击操作的 Windows 用户不同，Linux 开发者坚持使用键盘把命令输入到 shell。对于那些没有编程背景的人来说，这种转变一开始也许会不自然，但是在 Linux 中开发的好处很容易超过最初的学习投资。

学习几个重要的概念

和成熟的编程语言相比，bash 只需要学习几个主要的概念。这一步完成之后，之后 bash 的学习就只剩下记忆了。更清楚地说就是：要学好 bash，只需要记住 20—30 个命令（command）以及其中最常用的参数（argument）就可以了。

对于非开发者而言，Linux 很令人费解，因为开发者似乎能随意且不费力地使用深奥的终端命令。其实是因为他们只记住了少量的命令——对于更复杂的问题，他们（和所有普通人一样）也需要谷歌一下。

以下就是 bash 中的主要概念。

命令语法

bash 中的命令是区分大小写的，且遵循 {命令}{参数} 的语法结构。

例如，在『grep -inr』中，grep 是命令（搜索文本的一个字符串），-inr 是标记（flag）或参数（随 grep 默认运行而变化）。理解这个命令的唯一方法是使用谷歌搜索，或输入『man grep』命令。我推荐同时学习命令和其中最常用的参数，否则单独学习每一个标记的作用是很费力的。

目录相对地址

当前目录：.

上一级目录的上一级目录：..

用户的主目录：~

文件的系统根目录：/

例如，为了从当前目录换到上一级目录，需要输入：「cd..」。类似地，为了复制位于「/path/to/file.txt」文件到上一级目录中，需要输入「cp /path/to/file.txt..」（请注意命令末尾的点）。这些例子中使用的都是相对路径，可以使用绝对路径替换。

标准输入 (STDIN) /标准输出 (STDOUT)

任何输入和提交（通过键入 ENTER）到窗口的命令都被称为标准输入（standard input, STDIN）。

任何程序打印（print）到终端的东西（例如，一份文件中的文本）都被称为标准输出（standard output, STDOUT）。

管道 (PIPING)

1 |

一种管道，其左方是一个命令的 STNOUT，将作为管道右方的另一个命令的 STDIN。

例如：echo 'test text' | wc -l

2 >

大于号，作用是取一个命令 STDOUT 位于左方，并将其写入/覆写（overwrite）入右方的一个新文件。

例如：ls > tmp.txt

3 >>

两个大于号，作用是取一个命令 STDOUT 位于左方，并将其追加到右方的一个新的或现有文件中。

例如：date >> tmp.txt

通配符 (WILDCARDS)

这类似于 SQL 中的 % 符号，例如，使用「WHERE first_name LIKE 『John%』」搜索所有以 John 起始的名字。

在 bash 中，相应的命令是「John*」。如果想列出一个文件夹中所有以「.json」结尾的文件，可以输入：「ls *.json」。

TAB 键自动完成

如果我们输入一个命令并按下 TAB 键，那么 Bash 将自动完成该命令。但是，我们也应该使用一些如 zsh 或 fish 工具来自动完成，因为我们很难记住各种命令及它们的参数。更准确地说，这些工具会基于我们的命令行历史自动完成命令语句。

退出

有时候我们会卡在一些程序中并不知道如何退出它们。这在 Linux 新手中是很常见的问题，这也会大大损害新手的积极性。一般来说，退出命令会和字母「q」有一些关系，所以记住以下的退出命令或快捷键就十分有用了。

- Bash

CTRL+c

q

exit

- Python

`quit()`

`CTRL+d`

- Nano: `CTRL+x`

- Vim: `<Esc> :q!`

常用 Bash 命令

以下是在 Linux 中最常用到的指令，在使用新系统进行开发时，记住这些指令对于快速上手非常重要。

- `cd {directory}`: 转换当前目录
- `ls -lha`: 列出目录文件（详细信息）
- `vim` or `nano`: 命令行编辑器
- `touch {file}`: 创建一个新的空文件
- `cp -R {original_name} {new_name}`: 复制一个文件或目录（包含内部所有文件）
- `mv {original_name} {new_name}`: 移动或重命名文件
- `rm {file}`: 删除文件
- `rm -rf {file/folder}`: 永久删除文件或文件夹（小心使用）
- `pwb`: 打印当前工作目录
- `cat` or `less` or `tail` or `head -n10 {file}`: 文件的标准输出内容
- `mkdir {directory}`: 创建一个空的目录
- `grep -inr {string}`: 在当前目录或子目录的文件中搜索一个字符串
- `column -s, -t <delimited_file>`: 在 columnar 格式中展示逗号分隔文件

- `ssh {username}@{hostname}`: 连接到远程机器中
- `tree -LhaC 3`: 向下展示三级目录结构 (带有文件大小信息和隐藏目录信息)
- `htop` (or `top`): 任务管理器
- `pip install --user {pip_package}`: Python 安装包管理器, 安装包到 `~/.local/bin` 目录下
- `pushd . ; popd ; dirs ; cd -`: 在堆栈上 push/pop/view 一个目录, 并变回最后一个目录
- `sed -i "s/{find}/{replace}/g" {file}`: 替代文件中的一个字符串
- `find . -type f -name '*.txt' -exec sed -i "s/{find}/{replace}/g" {} \;`: 替换当前目录和子目录下后缀名为.txt 文件的一个字符串
- `tmux new -s session, tmux attach -t session`: 创建另一个终端会话界面而不创建新的窗口 [高级命令]
- `wget {link}`: 下载一个网页或网页资源
- `curl -X POST -d "{key: value}" http://www.google.com`: 发送一个 HTTP 请求到网站服务器
- `find <directory>`: 递归地列出所有目录和其子目录的内容

高级 & 不常用的指令

保留一个有用命令列表以备不需也是非常必要的, 即使这些情况不常发生 (如某个进程阻塞了几个网络端口)。以下我们将列出几个不常用命令:

- `lsof -i :8080`: 列出打开文件的描述符 (-i 是网络接口的标记)
- `netstat | head -n20`: 列出当前打开的 Internet/UNIX 接口 (socket) 以及相关信息
- `dstat -a`: 输出当前硬盘、网络、CPU 活动等信息
- `nslookup <IP address>`: 找到远程 IP 地址的主机名
- `strace -f -e <syscall> <cmd>`: 跟踪程序的系统调用 (-e 标记用于过滤某些系统调用)

- `ps aux | head -n20`: 输出目前活动的进程
- `file <file>`: 检查文件类型 (例如可执行文件、二进制文件、ASCII 文本文件)
- `uname -a`: 内核信息
- `lsb_release -a`: 系统信息
- `hostname`: 检视你的机器的主机名 (即其他电脑可以搜索到的名称)
- `pstree`: 可视化分支进程
- `time <cmd>`: 执行一个命令并报告用时
- `CTRL + z ; bg ; jobs ; fg`: 从当前 `tty` 中传递一个进程到后台再返回前台
- `cat file.txt | xargs -n1 | sort | uniq -c`: 统计文件中的独特字 (unique words) 数量
- `wc -l <file>`: 计算文件的行数
- `du -ha`: 在磁盘上显示目录及其内容的大小
- `zcat <file.gz>`: 显示压缩文本文件的内容
- `scp <user@remote_host> <local_path>`: 将文件从远端复制到本地服务器, 或反过来
- `man {command}`: 为一个命令显示 manual (说明文档), 但是通常这样不如谷歌搜索好用

原文链接: <http://alexpitalia.com/posts/2017/6/26/learning-linux-bash-to-get-things-done>