



Cygwin系列（五）：Shell命令行初体验



silaoA

爱好编程的电气工程师, silaoa.github.io

已关注

31 人赞同了该文章

发布于 2019-03-14 00:24，编辑于 2020-11-27 13:10

► 本文共5700余字，预计阅读时间15分钟，本文知乎链接：[Cygwin系列（五）：Shell命令行初体验](#)，本文同步发布于微信公众号。

前一篇文章[Cygwin系列（四）：一步一步搭建Cygwin最小系统中已建立了最小系统环境](#)，本篇将带你初步体验Cygwin的Shell命令行。

0x00 牛刀初试

利用setup安装完最小系统后，setup在桌面上建立了一个名为“Cygwin Terminal”的快捷方式。

桌面上Cygwin Terminal快捷方式

双击运行，弹出一个黑框框，初次运行还会自动帮用户做一些配置工作，如建立用户目录等。最后，打印一个形式如“xxx@yyy ~ \$”的提示符，紧接着是闪烁的光标，等待用户输入。如果直接按回车，黑框框里就换行继续打印出同样的提示符，同样跟着是闪烁的光标，继续等待用户输入。

那，可以做点什么呢？换句话说，我们能运行什么命令？先试试“ls /”，本人的输出结果如下图，注意到本人因做了额外配置，因此显示内容、颜色与你的可能不同，注意“s”和“/”之间有

空格隔开。



```
$ ls /
ls: 无法访问 '/mingw': No such file or directory
ls: 无法访问 '/f': No such file or directory
bin          Cygwin-Terminal.ico  home          mirror-local  setup.log.full
c            d                            lib           opt           setup-x86_64.exe
cygdrive     dev                          mingw         proc          tmp
Cygwin.bat   e                            mirror-list.txt sbin          usr
Cygwin.ico   etc                          setup.log     var
```

命令“ls /”的结果

再来解释这个命令，ls是命令名称，是list的简写，用于列出指定路径下的文件，而“/”就是指定的“路径”。那么，“/”指的哪个路径，如果稍稍留意输出结果，可以推测出“/”就是指Cygwin安装的顶层路径。好像没什么稀奇的，Windows下用Explorer也看到指定路径下的文件，而且更直观。

再继续，试试“help”，输出如下，结果比较长，截图只有一部分。这个命令貌似是告诉用户一些命令的用法。

```
$ help
GNU bash, 版本 4.4.12(3)-release (x86_64-unknown-cygwin)
这些 shell 命令是内部定义的。请输入 'help' 以获取一个列表。
输入 'help 名称' 以得到有关函数'名称'的更多信息。
使用 'info bash' 来获得关于 shell 的更多一般性信息。
使用 'man -k' 或 'info' 来获取不在列表中的命令的更多信息。

名称旁边的星号(*)表示该命令被禁用。

job_spec [&]
(( 表达式 ))
. 文件名 [参数]
:
[ 参数... ]
[[ 表达式 ]]
alias [-p] [名称[=值] ... ]
bg [任务声明 ...]
bind [-lpsvPSVX] [-m 键映射] [-f 文>
break [n]
builtin [shell 内建 [参数 ...]]
caller [表达式]
case 词 in [模式 [| 模式]...] 命令 ;;
cd [-L|[-P [-e]] [-@]] [目录]
command [-pVv] 命令 [参数 ...]
compgen [-abcdefgjkuv] [-o 选项] [->
complete [-abcdefgjkuv] [-pr] [-DE] >
compopt [-o|+o 选项] [-DE] [名称 ...>
continue [n]
history [-c] [-d 偏移量] [n] 或 hist>
if 命令; then 命令; [ elif 命令; the>
jobs [-lnprs] [任务声明 ...] 或 jobs>
kill [-s 信号声明 | -n 信号编号 | >
let 参数 [参数 ...]
local [option] 名称[=值] ...
logout [n]
mapfile [-d 分隔符] [-n 计数] [-O >
popd [-n] [+N | -N]
printf [-v var] 格式 [参数]
pushd [-n] [+N | -N | 目录]
pwd [-LP]
read [-ers] [-a 数组] [-d 分隔符] [->
readarray [-n 计数] [-O 起始序号] [->
readonly [-aAf] [名称[=值] ...] 或 r>
return [n]
select NAME [in 词语 ... ]; do 命令;>
set [--abefhkmnptuvxBCHP] [-o 选项名>
shift [n]
```

命令“help”的结果

初次体会到此结束，接下来并不想要你在黑框框里盲目乱敲，让我们暂且开启高空俯视视角，避免迷失在暗黑森林。

0x01 台前幕后

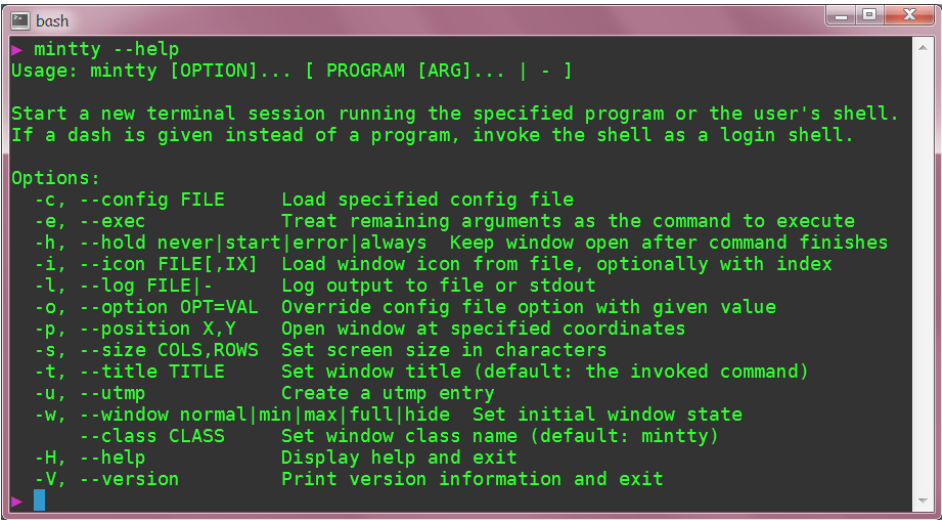
好奇心驱使我去了解双击“Cygwin Terminal”的快捷方式后，Windows帮我做了些什么。查看快捷方式的属性，其中“目标”字段的值为“D:\Cygwin64\bin\mintty.exe -e /bin/bash --login -i”，从这里可以推断，快捷方式实际运行的是mintty.exe，它就在Cygwin安装目录（D:\Cygwin64）的bin子目录中，那么mintty是什么？后边的“-e /bin/bash --login -i”又是什么意思？

查询mintty官网 <https://mintty.github.io>，上有文档介绍，mintty是用于Cygwin、MSYS、Msys2等项目的“终端模拟器（Terminal Emulator）”，顾名思义，它就是用来模拟终端的，但是终端自身仅具有基本输入输出能力，并不会执行程序指令。

再接着看，看到下面这张图，mintty的help输出中有一行“-e --exec Treat remaining arguments as the command to execute”，也就是“/bin/bash --login -i”是mintty启动后要执行的命令，这是要mintty启动bash.exe的节奏，还顺带传递了--login和-i选项，再一查

“bash”，大名鼎鼎的Bourne Again Shell。再结合mintty的help输出的头两行“Start a new session running the specified program or user's shell”，也就是说mintty启动的时候，会开

启一个终端会话运行指定的程序或用户Shell，上述 “/bin/bash --login -i” 正是 “指定的程序”。更多详情，mintty官网给了详细的手册页 mintty.github.io/mintty...。



mintty的help信息（图片来源：<https://mintty.github.io>）

进一步，打开Windows任务管理器，看到一个“应用” Terminal和一个“后台进程” bash.exe，右键Terminal，转到详细信息，正是mintty.exe。

查看到mintty和bash进程

至此，可以初步确定，**双击快捷方式实质是启动了mintty，而mintty又启动了后台进程bash，我们看到的前台窗口是mintty，而幕后则有bash在干活。**

经过一番操作（此处省略一万字.....），了解到终端、终端模拟器、Shell的概念，可以总结如下：

- ①终端自身并不执行用户输入的命令，它只是负责把输入的内容传给系统，并把结果呈现在窗口上给用户看；
- ②终端模拟器是一个模拟终端的图形界面软件，mintty就是一个终端模拟器实例；
- ③Shell是负责解释执行用户命令的解释器，bash就是一个Shell实例，它是由mintty在后台启动的。

从操作关系角度，可以简要绘制如下示意图，其中Cygwin核心是更深层幕后工作者，为mintty、bash的运行提供环境支撑。



用户、mintty、bash操作关系示意

0x02 概念与规则

好了，现在理清楚了，在mintty输入的内容实际都交给了Shell (bash) 在处理，输入的内容即为“命令”。俗话说“没有规矩不成方圆”，bash自有其一套语法规则，不合规则的输入内容不能视作合法命令，Shell (bash) 无法处理。

• 命令的形式

每条命令大体上遵循“<命令名称> <选项> <选项值> <参数>”的形式，命令名称 (command)、选项 (option)、选项值 (value)、参数 (argument) 之间用空格隔开。除了命令名称，其他部分都不是必需的，比如在本文最开头执行的“ls /”，其中ls是命令名称，/是参数，“help”就仅包含命令名称，“mintty --help”只包含命令名称和选项，没有选项值和参数。

计算机发明之初，存储器是很稀有的资源，为了尽量减少存储空间，也为了方便用户记忆，**命令名称都是单词简写**，十分简洁，这个传统基本上被保留下来。

命令名称描述要做什么 (Do What)，这个好理解，参数描述的是作用对象 (to Whom)，也比较好理解，而选项和选项值则具体描述怎么做 (How to Do)，这就比较复杂了。事实上，各个命令描述最多的地方就是选项和选项值。

选项有单个连字符 (-) 与字母组成的**短选项风格** (UNIX风格) 和两个连字符 (--) 与单词组成的**长选项风格** (GNU风格)，很多命令两者都支持，而且部分短选项和长选项等效，还有少数命令支持“+”与字母组成的选项，甚至与这些风格都不符的选项。

大部分命令都有“-v”和“-h”这两个选项，或等价的“--version”和“--help”，用于显示版本信息和帮助信息。熟悉应用程序，最主要地就是熟悉不同选项和不同参数配合作用，并加以掌握。对某个程序命令完全不了解或者是忘记了，都可以尝试“-h”、“--help”、“-v”、“--version”选项去了解。

• 有哪些命令

Shell能识别的命令大致分为3类：

①**内建(built-in)命令**，仅与Shell自身有关，不同的Shell (如sh、bash、zsh、fish等) 支持的内建命令；

②**用户定义函数 (function)、别名 (alias) 等**，用户可自定义一些Shell函数，给命名设置别名等；

③**外部命令**，额外安装的可执行程序越多，可供执行的外部命令就越多，比如ls就是外部命令。

要查询某个命令属于哪种，执行“type -a <命令名称>”即可，要看当前Shell支持哪些内建命令，执行“help”即可，其中type、help就是bash等Shell都会提供的内建命令。在刚安装完Cygwin最小系统时，bin子目录下大量可执行文件，除去一些DLL文件，其余都是Shell外部命令。

• 工作路径



文件都是存放在具体某个路径下的，Shell在工作的时候也是处在某个路径下，要查询当前工作路径（current working directory）是什么，可以通过内建命令“pwd”打印出来，内建命令“cd <路径>”可以切换工作路径。

Shell刚启动的时候，默认切换到当前用户目录作为初始工作路径，前文提到的提示符中，“~”就是当前用户目录，试下“pwd”如图，结果为“/home/<用户名>”。其中，home是Cygwin安装目录下的子目录，<用户名>是登录当前Shell的账号，在Cygwin中登录当前Shell的账号就是登录到Windows系统的用户名，这个子目录是首次启动Shell时自动建立的。

“~”指示当前用户目录

为什么要强调“当前”？因为UNIX/Linux是真正的多用户系统，允许多个用户同时登录到系统中取得使用权，系统给每个用户在/home目录下建立各自的用户目录，这样每个用户的环境彼此隔离、互不干扰。Windows是号称多用户系统，虽然也给每个用户建立各自的用户目录，在“C:\Users”下，但是Windows一次只允许一个用户登录。

0x03 命令急救箱

以上洋洋洒洒长篇大论，除了开头演示“ls /”、“help”和中间穿插介绍的type、cd、pwd，就没介绍几个命令用法。本节也只打算介绍几个关键命令，记住这几个，就可以去从零了解各个命令用法。

• help

help是Shell内建命令，可以查看各个内建命令的用法。如果不带参数，“help”输出当前Shell支持的所有内建命令；如果带参数，“help <参数1> <参数2> ...”则按参数逐个尝试查找帮助主题，通常所有参数应当为内建命令或者能匹配内建命令的模式，比如h*匹配所有以h开头的内建命令：help、history。

```
$ help help
help: help [-dms] [模式 ...]
    显示内建命令的相关信息。

    显示内建命令的简略信息。如果指定了 PATTERN 模式，
    给出所有匹配 PATTERN 模式的命令的详细帮助，否则打
    印一个帮助主题列表

    选项：
      -d      输出每个主题的简短描述
      -m      以伪 man 手册的格式显示使用方法
      -s      为每一个匹配 PATTERN 模式的主题仅显示一个用法
              简介

    参数：
      PATTERN  Pattern 模式指定一个帮助主题

    退出状态：
      返回成功，除非 PATTERN 模式没有找到或者使用了无效选项。
```

“help help”结果

• -h或--help选项

对于绝大多数外部命令，都支持至少“-h”和“--help”选项之一，用于显示该命令的用法，即使“-h”可能用作他途，但“--help”绝对是显示用法。


```

$ ls --help
用法: ls [选项]... [文件]...
List information about the FILES (the current directory by default).
Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

必选参数对长短选项同时适用。
-a, --all              不隐藏任何以 . 开始的项目
-A, --almost-all      列出除 . 及 .. 以外的任何项目
--author              与 -l 同时使用时列出每个文件的作者
-b, --escape           以八进制溢出序列表示不可打印的字符
--block-size=SIZE      scale sizes by SIZE before printing them; e.g.,
                        '--block-size=M' prints sizes in units of
                        1,048,576 bytes; see SIZE format below
-B, --ignore-backups   do not list implied entries ending with ~
-c                    with -lt: sort by, and show, ctime (time of last
                        modification of file status information);
                        with -l: show ctime and sort by name;
                        otherwise: sort by ctime, newest first
-C                    list entries by columns
--color[=WHEN]         colorize the output; WHEN can be 'always' (default
                        if omitted), 'auto', or 'never'; more info below
-d, --directory        list directories themselves, not their contents
-D, --dired             generate output designed for Emacs' dired mode
-f                    do not sort, enable -aU, disable -ls --color
-F, --classify          append indicator (one of */=>@|) to entries
--file-type            likewise, except do not append '*'
--format=WORD           across -x, commas -m, horizontal -x, long -l,

```

“ls --help” 结果

也有少数命令不支持 “-h” 或 “--help” 选项，而是在没有任何选项参数或者出错的时候显示用法。

• man

man是POSIX规定的帮助手册程序，是manual的简写。软件包中除了程序本身，还带了大量用户文档，在UNIX/Linux系统中，手册页是最常见的用户文档之一，man手册页用特殊的格式写成并压缩，通过man命令浏览。系统安装的软件包手册页通常在/usr/share/man/目录之下，按照章节分门别类地放在man1~man8子目录中。

参照上面的说明，试试 “man --help”，如下图man支持的选项很多，<章节>参数可以是1~8，指定man查找的章节。一般使用场景中不需要任何选项，直接 “man <主题>” 即可，man从1~8所有章节去查找，主题不限于命令名称，还可以是系统调用、函数库等。

```

$ man -h
用法: man[选项...] [章节] 手册页...

-C, --config-file=文件  使用该用户设置文件
-d, --debug             输出调试信息
-D, --default           将所有选项都重置为默认值
--warnings[=警告]      开启 groff 的警告

主要运行模式:
-f, --whatis            等同于 whatis
-k, --apropos           等同于 apropos
-K, --global-apropos    search for text in all pages
-l, --local-file        把“手册页”参数当成本地文件名来解读
-w, --where, --path, --location 输出手册页的物理位置
-W, --where-cat, --location-cat 输出 cat 文件的物理位置

-c, --catman            由 catman 使用，用来对过时的 cat
                        页重新排版
-R, --recode=编码       output source page encoded in ENCODING

寻找手册页:
-L, --locale=区域       定义本次手册页搜索所采用的区域设置
-m, --systems=系统      use manual pages from other systems

```

“man --help” 结果

在man查看手册页时，空格键向下翻页，PgUp/PgDn键向上/下翻页，↑/↓键向上/下移动一行，g/G跳转到页面顶部/底部，q退出，h查看帮助操作。更多详情，试试 “man man”。

• info



用户文档除了man手册页，还有info格式的手册页，由GNU项目引入，通过info（information的缩写）命令浏览，一般UNIX/Linux系统同时支持man和info两种风格的手册页。

Info手册页按照节点（node）组织，每个手册页文件是一个节点，手册页内支持链接到其他节点，如此组织犹如一张网，和网页类似。Info手册页一般存放在/usr/share/info目录下，其中dir文件是Top节点，相当于网站中的index首页。

老规矩，“info --help”试试怎么用，如图，info也支持较多选项，给出的例子中，直接“info <主题>”，和man类似。不加主题参数时，“info”默认打开Top节点。

```
$ info --help
用法: info [OPTION]... [MENU-ITEM...]

以 Info 格式阅读文档。

Frequently-used options:
-a, --all                use all matching manuals
-k, --apropos=STRING      look up STRING in all indices of all manuals
-d, --directory=DIR      add DIR to INFOPATH
-f, --file=MANUAL        specify Info manual to visit
-h, --help               display this help and exit
    --index-search=STRING go to node pointed by index entry STRING
-n, --node=NODENAME      specify nodes in first visited Info file
-o, --output=FILE        output selected nodes to FILE
-O, --show-options, --usage go to command-line options node
    --subnodes           recursively output menu items
-v, --variable VAR=VALUE assign VALUE to Info variable VAR
    --version            display version information and exit
-w, --where, --location  print physical location of Info file

The first non-option argument, if present, is the menu entry to start from;
it is searched for in all 'dir' files along INFOPATH.
If it is not present, info merges all 'dir' files and shows the result.
Any remaining arguments are treated as the names of menu
items relative to the initial node visited.

For a summary of key bindings, type H within Info.

Examples:
  info                show top-level dir menu
  info info-stdnd     show the manual for this Info program
  info emacs          start at emacs node from top-level dir
```

“info --help” 结果

在info浏览手册页时，空格向下翻页，PgUp/PgDn键向上/下翻页，↑/↓键向上/下移动一行，b/e跳转至当前节点页首（begin）/页末（end），q退出，Enter跟踪链接，p/n/u跳转至与当前节点关联的前一个（previous）/后一个（next）/上一级（up）节点，l回退至上次（last）页面，m/g跳转至本节点内菜单项，Ctrl-g取消当前要执行的动作。更多详情，试试“info info”。

0x04 其他提示

- UNIX、GNU、BSD的程序软件数不胜数，而且数量不断增长，我们**不可能记住每个程序怎么用，也不应试图去背、去记忆**，常用的命令和常见的用法只有一小部分，**熟能生巧**。不熟悉的话，试下--help选项、man、info，配合网上应用案例，也能摸清个大概了。
- UNIX、GNU、BSD的命令行程序功能单一而强悍，Shell可以巧妙灵活地将其组织起来，发挥出更加强大的功效，用多用熟悉了才能体会到。
- **英语真的比较重要，英语水平不需要很高，但要能够阅读手册页等文档。**
- Shell有很多特性，对于初学者来讲，最有用莫过于Tab键补全。在终端中输入命令、路径、文件等名称的时候，记得不准确，字符多了容易出错，Tab补全是一大福音。比如输入“m”，Shell会列出所有匹配以“m”开头的命令；再紧接着输入“a”，Shell列出所有匹配以“ma”开头的命令，这样逐渐缩减范围，直至唯一匹配。同样也适用于目录和文件补全，文件或目录名字很长时非常容易出错又难于检查，Tab键补全可以省去不少敲键盘的时间，还避免了错误，文件或目录名补全支持中文。
- **为练习本文内容，建议针对/bin目录下所有命令逐个尝试--help选项、查看man/info手册。**

参考

- en.wikipedia.org/wiki/C...
- <https://mintty.github.io>

更多阅读



- [上一篇：Cygwin系列（四）：一步一步搭建Cygwin最小系统](#)
- [下一篇：Cygwin系列（六）：使用Cygwin常见问题及应对](#)
- [伪码人专栏目录导航（持续更新...）](#)
- [Cygwin前传：从割据到互补](#)
- [Cygwin系列（九）：Cygwin学习路线](#)
- [微软WSL——Linux桌面版未来之光](#)
- [GNU Wget 爬虫？试一试](#)
- [silaoA的博客.https://silaoa.github.io](#)

如本文内容引起极度舒适，请转发分享，欢迎关注与留言交流

► 本文为原创文章，如需转载请私信知乎账号silaoA或联系公众号伪码人（We_Coder）。

都看这里了，不妨点个赞再走呗

发布于 2019-03-14 00:24，编辑于 2020-11-27 13:10

「真诚赞赏，手留余香」

赞赏

还没有人赞赏，快来当第一个赞赏的人吧！

[Cygwin](#) [Linux](#) [命令行界面 \(CLI\)](#)

写评论... | 你赞过作者的回答

**春风Shi里不如妮**

...

您好，我想问一下，为什么您的help界面中是中文呢？请问如果设置？

2021-03-26 · 作者回复了

● 回复 🍎 赞

**silaoA** 作者

...

这个与程序本地化有关，首先locale里有关语言的部分设置为zh_CN.utf8，其次程序本身得有中文消息支持，也就是在/usr/share/locale/zh_CN/LC_MESSAGES有相应的.mo文件。

2021-03-26

● 回复 🍎 赞

**正中央**

...

您好，假如我有一个linux软件，我可以把它安装在cygwin里面并运行吗，谢谢

2020-12-23 · 作者回复了

● 回复 🍎 赞

**silaoA** 作者

...

不可以。Linux下的二进制文件格式（elf）和Windows的不一样，cygwin是不识别的，cygwin只能做到源码级兼容，需要把源码在cygwin环境重新编译，生成Windows格式（pe）程序。

2020-12-24

● 回复 🍎 赞

文章被以下专栏收录

**伪码人 We_Coder**

专注Cygwin、Linux、电气等技术交流分享

**技术部落联盟**

涉及容器、大数据、人工智能、物联网等领域