

没有开花的树

于 2016-09-03 16:26:14 发布

38130 收藏 12

分类专栏: linux/centos

linux/centos 专栏收录读内容

0 订阅 12 篇文章 订阅专栏

在Linux命令行下，输入字符后，按两次Tab键，shell就会列出以这些字符打头的所有可用命令。如果只有一个命令匹配到，按一次Tab键就自动将这个命令补全。

比如，想更改密码，但只记得这个命令前几个字母是pass，这时候，按Tab键，shell就自动输出 passwd 命令，非常方便。

当然，除了命令补全，还有路径、文件名补全。这个在我们 cd 到特定目录时特别好用。

命令补全效果，如下：

```
[root@localhost ~]#
```

没有开花的树
http://blog.csdn.net/mycwg

那么，自己开发的程序，该怎么实现Tab自动补全？

补全命令说明

自动补全是Bash自带的一个强大的功能，允许通过编码指定命令参数如何补全。通常，补全脚本会放在/etc/bash_completion.d/ 目录下，方便统一启用所有补全脚本。

这里例子的命令为 foo

```
# cat /etc/bash_completion.d/foo.bash
_foo()
{
    local cur=${COMP_WORDS[COMP_CWORD]}
    COMPREPLY=( $(compgen -W "exec help test" -- $cur) )
}
complete -f _foo foo
```

如下，测试foo命令是否自动补全

```
# chmod +x /etc/bash_completion.d/foo.bash
# source /etc/bash_completion.d/foo.bash
# foo [Tab][Tab]
exec help test
```

以上，source是为了这个foo.bash在当前会话生效。默认情况下，这个补全脚本不会被执行到，也就是说，补全命令未激活。需要source激活这个脚本，就可以没有感知地使用了。

为了避免每次都要source一次，可以在bashrc加上这个命令。

bashrc全局配置在不同 linux 发行版可能位置不同，如下：

Centos	/etc/bashrc
Ubuntu	/etc/bash.bashrc

(如果只在当前帐号生效，只要配置 ~/.bashrc 即可)

在bashrc文件末尾加上 source /etc/bash_completion.d/foo.bash，这样，每次登录到linux后，就会激活这个补全脚本。

补全命令详解

前面给大家演示的例子，用到两个命令complete和compgen，下面分别介绍这两个命令。

complete (补全命令)

这是命令补全是核心的命令了，来看下这个命令的参数说明。

```
# help complete
complete: complete [-abcdefgjkuv] [-pr] [-DE] [-o option] [-A action] [-G globpat] [-W wordlist] [-F function] [-C
command] [-X filterpat] [-P prefix] [-S suffix] [name ...]
```

重点说明：

-F function	执行shell 函数，函数中生成COMPREPLY作为候选的补全结果
-C command	将 command 命令的执行结果作为候选的补全 结果
-G pattern	将匹配 pattern的文件名作为候选的补全结果
-W wordlist	分割 wordlist 中的单词，作为候选的补全结果
-p [name]	列出当前所有的补全命令
-r [name]	删除某个补全命令

演示下：

```
# complete -W 'word1 word2 word3 test' foo
# foo w<Tab>
# foo word<Tab>
# complete -p
complete -W 'word1 word2 word3 test' foo
complete -o filenames -F __udisks udisks
# complete -r foo
# complete -p
complete -o filenames -F __udisks udisks
```

目录

补全命令说明

补全命令详解

complete (补全命令)

compgen (筛选命令)

compost (修改补全命令设置)

内置补全变量

命令行补全实例

分类专栏

服务端架构

4篇

erlang技术与应用

66篇

skynet技术与应用

9篇

C/C++技术与应用

18篇

游戏开发

5篇

windows

15篇

linux/centos

12篇

jquery

10篇

apache/php

14篇

c/c++/vc++

25篇

erlang源码分析

7篇

erlang/immesia

67篇

svn/git

4篇

mysql

9篇

sqlite

3篇

protobuf

6篇

unity3d

4篇

flash/as3/flex

2篇

eclipse/IDEA

6篇

hadoop/hbase

1篇

skynet/lua

14篇

这个命令，用来筛选生成 匹配单词的 筛选补全结果

```
# help compgen
compgen: compgen [-abdefgjkuv] [-o option] [-A action] [-G globpat] [-W wordlist] [-F function] [-C command] [-X filterpat] [-P prefix] [-S suffix] [word]
```

重点说明:

-W wordlist	分割 wordlist 中的单词，生成筛选补全列表
-------------	---------------------------

```
# compgen -W 'word1 word2 test'
word1
word2
test
# compgen -W 'word1 word2 test' word
word1
word2
```

compopt (修改补全命令设置)

这个命令可以修改补全命令设置，注意了，这个命令必须在补全函数中使用，否则会报错。

```
# help compopt
compopt: compopt [-o+o option] [-DE] [name ...]
```

重点说明:

+o option	启用 option 配置
-o option	弃用 option 配置

例如，设置命令补全后不要多加空格，方法如下：

```
compopt -o nospace
```

内置补全变量

除了上面三个命令外，Bash还有几个内置变量来辅助补全功能，如下：

COMP_WORDS	类型为数组，存放当前命令行中输入的所有单词
COMP_CWORD	类型为整数，当前输入的单词在COMP_WORDS中的索引
COMP_REPLY	类型为数组，筛选的补全结果
COMP_WORDBREAK \$	类型为字符串，表示单词之间的分隔符
COMP_LINE	类型为字符串，表示当前的命令行输入字符
COMP_POINT	类型为整数，表示光标在当前命令行的哪个位置

命令行²补全实例

下面再结合前面三个补全命令（complete/compgen/compopt）和内置变量，写了例子说明下。

```
# cat /etc/bash_completion.d/foo.bash
foo()
{
    COMP_REPLY=(
        local cur=${COMP_WORDS[COMP_CWORD]};
        local cmd=${COMP_WORDS[COMP_CWORD-1]};
        case $cmd in
            'foo')
                COMP_REPLY=( $(compgen -W 'help test read' -- $cur) ); ;
            'test')
                local pro=( $(awk '{print $1}' /data/a.txt) )
                COMP_REPLY=( $(compgen -W "${pro[@]}" -- $cur) ); ;
            '+' )
                ;;
        esac
        if [[ "${COMP_WORDS[1]}" == "read" && ${COMP_CWORD} -eq 2 ]]; then
            local pro=($(pwd))
            cd /data
            compopt -o nospace
            COMP_REPLY=( $(compgen -d -f -- $cur) )
            cd $pro
        fi
        return 0
    }
complete -f _foo foo
```

例子中，foo有3个参数，分别是 help, read, test

read 测试遍历 /data 目录下所有文件

test 测试从文件中提取2级参数

help 只是演示，没有特殊作用

