

通配符： 文件名替换。 允许用户依据字符模式选择文件名

通配符

通配符	匹配项
*	匹配任意多个字符
?	匹配单一字符
[(!)characters]	匹配任意一个(不)属于字符集中字符
[:class:]	匹配任意一个属于指定字符类中的字符

以下表格对应👉最后一行（字符集为[] 中的内容 上面的字符集为 [:class:] ）

字符类	匹配项
[:album:]	匹配任意一个字母或者数字
[:alpha:]	匹配任意一个字母
[:digit:]	匹配任意一个数字
[:lower:]	匹配任意个小写字母
[:upper:]	匹配任意一个大写字母

以下为示例

通配符

模式	匹配项
*	所有文件
g*	以g开头的所有文件
b*.txt	以b开头，中间任意多个字符，并以.txt结尾的所有文件
Data???	Data后面跟三个字符的所有文件
[abc]*	以abc中任意个开头的所有文件
Back_[1-9]	以back.开头，后面跟一个数字的所有文件
[:upper:]*	以大写字母开头的所有文件
[!:digit:]*	不以数字开头的所有文件
*[:lower:]123]	以小写字母或123结尾的所有文件

Linux文件目录

/bin

包含系统启动和运行所必需的二进制文件
包含linux内核、最初的RAM磁盘映像（系统启动时，驱动程序会用到），以及启动加载程序

/boot

/boot/grub/grub.conf 或 menu.lst 用来配置启动加载程序
/boot/vmlinuz，linux内核

/dev

一个包含设备节点的特殊目录。内核将所有能够识别的设备存放在这个目录中
包含所有系统层面的配置文件，同时也包含一系列的shell脚本
该目录中包含的内容都应该是可读文件

/etc

比较有趣的文件夹

/etc/passwd 用户账号列表。 shadow 用户密码列表
/etc/crontab 自动化任务运行的时间（还存有SHELL PATH的路径）
/etc/fstab 存储设备以及相关挂载点的列表

Linux命令

查看系统空间

df 查看硬盘驱动器的可用空间
free 查看可用内存

cd快捷方式

cd 工作目录到主目录
cd - 当前目录变成前一个目录
cd ~username 到username的目录

文件操作

创建目录

mkdir *directory* ... (参数后面带有三个点号表示参数重复)

查看文件

ls命令

选项	长选项	含义
-a	—all	列出所有文件
-d	—directory	列出目录内容
-F	—classify	每个名字末尾加上类型指示符
-h	—human-readable	显示文件大小
-r	—reverse	反序输出（正常按照字母升序）
-S		按照文件大小排序输出
-t		按照修改时间排序

查看文件类型

```
file filename 查看文件类型
```

查看文件内容

```
less 查看文本文件的程序 （与vim 使用方式差不多
```

复制文件和目录

```
cp item1 item2
```

选项	含义
-a —archive	复制文件和目录及其属性
-i --interactive	覆盖一个已存在文件，会进行提示
-r --recursive	递归的复制目录及其内容。复制目录需要
—u —update	将文件从一个目录复制到另一个目录时，只会复制那些不存在的文件
-v --version	复制文件，显示消息性消息

移动和重命名文件

```
mv item1 item2
将文件（或目录）item1 移动（或重命名）为item2
选项 i u v 与上面的性质一样
```

删除文件和目录

```
rm item ...
```

选项	含义
-i —interactive	删除一个已经存在文件前，提醒用户确认
-r —recursive	递归删除目录，如果有子目录，也将其删除
-f -force	忽略不存在文件并无需提示
-v -verbose	删除文件时显示消息性文件

创建链接

```
ln file link 用来创建硬连接
ln -s item link 用来创建符号链接
```

识别命令

type —— 显示命令的类型

```
type command
```

which ——显示可执行程序的位置

```
有时候，程序可能安装一个可执行程序多个版本，which可以确定一个给定可执行文件的准确位置
```

help——获得shell内置命令的帮助文档

```
bash为每一个shell内置命令提供了一个内置的帮助工具。
例如： help cd
```

help——显示命令的使用信息

```
很多可执行程序都支持 --help 选项， --help选项描述了命令支持的语法和选项
```

man——显示程序的手册页

```
文档的组织结构
```

部分	内容
1	用户命令
2	内核系统调用的程序接口
3	c库函数程序接口
4	特殊文件，如设备节点和驱动保护程序
5	文件格式
6	游戏和娱乐，例如屏保
7	其他杂项

部分	内容
8	系统管理命令

whatis——显示命令的简要描述

```
whatis 程序显示匹配具体关键字的手册页的名字和一行描述
```

info——显示程序的info条目 *****

```
暂时掠过
```

README和其他程序文档文件

```
系统中安装的很多软件包由自己的文档文件，他们存放在/user/share/doc目录中。大部分是以纯文本格式存储的
```

使用别名创造自己的命令

```
我们可以使用alias命令来创建自己的命令
```

- 1. 使用 不带参数的alias 查看所有已使用的别名
- 2. 使用 alias创造自己的命令
 - 2.1 使用 type 命令查看是否是已经用过的名字
 - 2.2 使用 alias name='string'eg: （alias foo='cd /user; ls; cd -'）创建新命令的别名
- 3. 使用unalias 命令删除别名

```
alias 的效力仅限于本次操作，若要每次登陆就自动设置好别名，可以在 /etc/profile 末尾设置指令的别名（用于 mac 电脑），或者在 ~/.bashrc 中设置（虚拟机中的Linux可以使用）
```

```
如果需要在每个用户都有效的别名，则把alias命令 加在 /etc/bashrc 最后面
```

命令行最炫酷的功能！ I/O重定向

1. 标准输入，标准输出和标准错误

```
1 | 与 UNIX “一切都是文件” 的思想一致，类似 ls 的程序实际上吧他们的运行结果发送到一个称为标准输出 （standard output）
2 | 的设备得到输出，默认情况下，标准输出连接到键盘。
3 | 许多程序从一个称为标准输入（standard input，表示为stdin）的设备得到输入，默认情况下，标准输入连接到键盘。
```

2. 标准输出重定向（主要用于把输出内容报春到文件中，而不是输出到屏幕）

```
1 | I/O重定向功能可以重新定义标准输出内容发送到哪里。使用重定向操作 “>”，后面接文件名，就可以把标准输出重定向到！
```

```
1 | chenlindaMacBook-Pro:~ chenlin$ ls -l /usr/bin/ > ls-output.txt
2 | chenlindaMacBook-Pro:~ chenlin$ ls -l ls-output.txt
3 | -rw-r--r--  1 chenlin  staff  70562  8  2 14:57 ls-output.txt
4 | //可以使用 less 查看这个文件的内容
```

```
将创建/usr/bin 目录的一个长列表信息，并将这个结果输入到ls-output.txt文件中。
/user/bin 目录存放大部分软件使用的指令存放的地方。 /bin存放系统所需的指令的地方
```

现在，让我们做重复的重定向测试，这次把目录名称换成一个不复存在的目录

```
1 | chenlindaMacBook-Pro:~ chenlin$ ls -l /bin/usr > ls-output.txt
2 | ls: /bin/usr: No such file or directory
```

我们会收到一条错误信息。因为我们指定的目录不存在，所以这个错误信息是正确的。 **但是为什么错误信息显示在屏幕上，而不是重定向到ls-output.txt文件中呢？** ==原因是ls程序并不会把它的运行的错误信息发送到标准输出文件中。而是与绝大多数写的很好的UNIX程序一样，把错误信息发送到标准错误文件中。==因为我们只重定向到标准输出，并没有重定向到标准错误所以错误信息仍然输出到屏幕上。

首先查看这个输出文件发生了什么变化！

```
1 | chenlindaMacBook-Pro:~ chenlin$ ls -l ls-output.txt
2 | -rw-r--r--  1 chenlin  staff    0  8  2 14:41 ls-output.txt
```

当前文件大小为零～！ **这是因为使用重定向符 “>” 来重定向标准输出时，目的文件通常会从文件开头部分重写，由于ls命令执行后没有输出任何内容，只输出一条错误信息，所以重新改写这个文件。事实上，如果需要删除一个文件的内容，或创建一个新的文件，可以采用这样的方式**

```
1 | chenlindaMacBook-Pro:~ chenlin$ > ls-output.txt
```

那么，如何不从文件的首位置开始覆盖文件，而是从文件的尾部添加内容～， 可以使用重定向符“ >> ”来实现 eg:

```
1 | chenlindaMacBook-Pro:~ chenlin$ ls -ls /usr/bin/ >> ls-output.txt
2 | chenlindaMacBook-Pro:~ chenlin$ ls -ls /usr/bin/ >> ls-output.txt
3 | chenlindaMacBook-Pro:~ chenlin$ ls -ls /usr/bin/ >> ls-output.txt
4 | chenlindaMacBook-Pro:~ chenlin$ ls -l ls-output.txt
5 | -rw-r--r--  1 chenlin  staff 211686  8  2 14:56 ls-output.txt
```

3.标准错误重定向

标准错误重定向不能简单的使用一个专用的重定向符来实现
要实现标准错误的重定向，不得不提到它的文件描述符（file descriptor）。
一个文件可以把生成的输出内容发送到任意文件中。如果把这些文件流中前三个分别对应标准输入文件、标准输出文件、标准错误文件，那么shell将在内部使用**文件描述符**分别索引它们为0、1、2。shell提供了文件描述输入编号来重定向文件的表示法。
[外链图片转存失败,源站可能有防盗链机制,建议将图片保存下来直接上传(img-H0oN2MDs-1599556799116)(media/15960824450559/15963528337931.jpg)]
（图片百度而来）

```
1 | chenlindeMacBook-Pro:~ chenlin$ ls -l /bin/usr 2> ls-error.txt
2 | //文件描述符 “2” 紧跟在重定向符之前，将标准错误重定向到 ls-error.txt 文件中。
```

4.将标准输出和标准错误重定向到同一个文件

```
1 | chenlindeMacBook-Pro:~ chenlin$ ls -l /bin/usr > ls-output.txt 2>&1
```

使用这个办法，将执行两个重定向操作。首先将标准输出重定向ls-output.txt文件中，再使用标记符 2&>1 把文件描述符2 重定向到文件描述符1中。

注意⚠️

重定向的操作顺序非常重要，标准错误的重定向通常发生在标准输出重定向操作之后，否则它不起作用。

2. bash版本提供了效率更高的第二种方法

```
1 | chenlindeMacBook-Pro:~ chenlin$ ls -l /bin/usr &> ls-output.txt
```

只使用一个标记符 "&>" 就把标准输出和标准错误都重定向到了ls-output.txt 文件中。

5.处理不想要的输出

有时候，命令执行后我们并不希望获得输出，而是把这个输出丢弃，尤其是在输出错误和状态信息的情况下更为需要。系统提供了一种方法，即通过把输出重定向到一个称为/dev/null的特殊文件中.这个文件是一个称为位桶（bit bucket）的系统设备，它接受输入但是不对输入进行任何处理。以下命令可以抑制（隐藏）一个命令的错误信息

```
1 | chenlindeMacBook-Pro:dev chenlin$ ls -l /bin/usr 2> /dev/null
```

6.标准输入重定向

学习之前，先了解一个命令

cat —— 合并文件

cat 命令读取一个或多个文件，并把它们复制到标准输出文件中 cat *[file]*
cat 经常用来显示短的文本文件。 由于可以接受多个文件作为输出参数。所以它也可以用来把文件连接在一起。

eg：例如多个文件movie.mpeg.001 movie.mpeg.002 ... movie.mpeg.099

可以使用这个命令将他们重新连接在一起

```
1 | chenlindeMacBook-Pro:~ chenlin$ cat movie.mpeg.0* > movie.mpeg
```

如果输入cat 不带任何参数，会出现什么结果~

```
1 | chenlindeMacBook-Pro:~ chenlin$ cat
2 |
```

没有任何结果—— 它只是停留在那边不动，好像被挂了起来，但是它实际上正在执行我们所期望它做的事情
如果cat 命令没有给定任何参数，它将从标准输入读取内容。由于标准输入在默认情况下连接到键盘，所以它实际上在等待键盘输入内容！

管道

命令从 标准输入 到 读取数据， 并将 数据发送到 标准输出 的能力，是使用了名为管道的shell特性。使用管道操作符“ | ” 可以把一个命令的标准输出传送到另一个命令的 标准输入中。

```
1 | command1 | command2
```

eg:

```
1 | chenlindeMacBook-Pro:~ chenlin$ ls -l /usr/bin/ | less
```

1.过滤器

把多条命令合在一起构成一个管道。这种方式用到的命令通常被称为过滤器（filter）

eg: 把/bin 和 /usr/bin 目录下可执行程序合成一个列表，并且按照顺序排序，最后在查看这个列表

```
1 | chenlindeMacBook-Pro:~ chenlin$ ls /bin/ /usr/bin/ | sort | less
```

uniq —— 报告或忽略文件中重复的行

忽略文件中重复的行（uniq 可以接收标准输入或者一个单一文件名参数对应的**已排好序的数据列表**）

```
1 | chenlindeMacBook-Pro:~ chenlin$ ls /bin/ /usr/bin/ | sort | uniq | less
```

查看重复行命令

```
1 | chenlindeMacBook-Pro:~ chenlin$ ls /bin/ /usr/bin/ | sort | uniq -d | less
```

wc —— 打印行数、字数和字节数（-l 参数，只报告行数）

例如要查看排好序列表中的条目数


```
1 | chenlindeMacBook-Pro:~ chenlin$ ls /bin/ /usr/bin/ | sort | uniq | wc -l
```

grep —— 打印匹配行

-i 忽略大小写。 -v。显示与模式不匹配的行

head/tail —— 打印文件的开头部分/结尾部分（tail 可以用来实时监控）

可以通过 -n 来调整输出的行数（一般基本上是10行）
使用 tail -f 可以用来实时监控文件，一旦添加了最新行，新行将会立即显示在屏幕上。

tee —— 从stdin 读取数据，并同时输出到stdout和文件

权限

传统的UNIX操作系统与传统的MS-DOS操作系统不同，区别在于它们不仅是多重任务处理系统，而且还是多用户系统。
例如：如果计算机连接到一个网络或者互联网中，远程用户可以通过ssh（安全shell）登录并操作这台计算机。

id : 显示用户身份标示

```
1 | root@chenlin:~/userfile# id
2 | uid=0(root) gid=0(root) groups=0(root)
```

查看id命令的输出结果。在创建用户账号的时候，用户将被分配一个称为用户ID（userID）或者uid的号码。用户名与用户名一一映射。
同时用户将被分配一个有效组ID或者称为gid，而且该用户也可以归属于其他的群组。

用户账户定义在文件 /etc/passwd 中
用户组定义在文件 /etc/group中
在创建用户账户和群组时，这些文件随着文件/etc/shadow的变动而修改
/etc/shadow保存了用户的密码信息
==对于每一个用户账户，文件/etc/passwd中都定义了对应用户的用户名，uid， gid，账户的真实姓名，主目录，登录shell信息。

chmod：更改文件的模式(权限)

```
1 | root@chenlin:~/userfile# ls -l foo.txt
2 | -rw-r--r-- 1 root root 0 Sep  8 15:45 foo.txt
```

文件类型

属性	文件类型
-	普通文件
d	目录文件
l	符号链接，文件属性始终是rwxrwxrwx，它是一个伪属性值，指向的文件的属性才是真正的文件属性
c	字符设备文件，以字节流形式处理数据的设备
b	块设备文件，以数据块方式处理数据的设备

更改文件的权限

```
1 | root@chenlin:~/userfile# ls -l
2 | -rwxrwxrwx 1 root root 0 Sep  8 15:45 foo.txt
3 | root@chenlin:~/userfile# chmod 600 foo.txt
4 | root@chenlin:~/userfile# ls -l
5 | -rw----- 1 root root 0 Sep  8 15:45 foo.txt
```

umask：设置文件的默认权限

umask 命令控制着创造文件时指定给文件的默认权限，它使用八进制表示法来表示从文件模式属性中删除一个位掩码。

```
1 | root@chenlin:~/userfile# umask
2 | 0022
3 | root@chenlin:~/userfile# > foo.txt
4 | root@chenlin:~/userfile# ls -l foo.txt
5 | -rw-r--r-- 1 root root 0 Sep  8 16:04 foo.txt
```

更改umask值

```
1 | root@chenlin:~/userfile# rm -f foo.txt
2 | root@chenlin:~/userfile# umask 0002
3 | root@chenlin:~/userfile# > foo.txt
4 | root@chenlin:~/userfile# ls -l foo.txt
5 | -rw-rw-r-- 1 root root 0 Sep  8 16:05 foo.txt
```

将umask值与原始文件模式进行比对				
原始文件模式	—	rw-	rw-	rw-
掩码	000	000	000	010
结果	—	rw-	rw-	r—

有 1 出现的位置 对应的属性被删除

一些特殊的权限

虽然通常看到的八进制权限掩码都是用三位数字表示的，但是，从技术层面上来说，它使用 **四位数字** 表示的 1.setuid位（八进制表示为4000），把它应用到一个可执行文件时，有效用户ID将从实际用户ID（实际运行该程序的用户）设置成该程序所有者的ID。大多数情况下，该权限设置通常应用于一些超级用户所有的程序。当普通用户运行一个具有“setuid root”（以设置setuid位，由root用户所有）属性的程序时，该程序将以超级用户的权限来执行。使得程序可以访问一些普通用户禁止访问的文件和目录。

```
1 root@chenlin:~/userfile# chmod u+s foo.txt
2 root@chenlin:~/userfile# ls -l
3 -r-Sr--r-- 1 root root 0 Sep  8 16:05 foo.txt
```

2.setgid（八进制表示为2000），它会把有效组ID从该用户的实际组ID更改为该文件所有者的组ID。如果对一个目录设置setgid，那么该文件下新创建的文件将由该目录所在组所有，而不是文件创建者所在组所有。

```
1 root@chenlin:~/userfile# chmod g+s foo.txt
2 root@chenlin:~/userfile# ls -l
3 -r-Sr-Sr-- 1 root root 0 Sep  8 16:05 foo.txt
```

3.sticky位（八进制表示为1000）。它是从传统的UNIX中继承下来的，可以标记一个可执行文件位“不可交换的”。在Linux中，会忽略文件的sticky位，如果对一个目录设置sticky位，能阻止用户删除或者重命名文件，除非用户时这个目录的所有者、文件所有者、超级用户。

```
1 root@chenlin:~# chmod +t userfile/
2 root@chenlin:~# ls -l | grep userfile
3 drwxr-xr-t 2 root root      4096 Sep  8 16:05 userfile
```

su：以另一个用户的身份运行shell

会启动一个新的shell环境

sudo：以另一个用户的身份来执行命令

不会启动新的shell环境

/etc/sudoer配置文件

chown：更改文件的所有者和所属组群

语法格式 `chown [owner][:[group]] file`

chown命令行参数实例

参数	结果
bob	文件所有者改成bob
bob: user	文件所有者改成bob，所属组改成user
: admins	所属组改成admins
bob:	所有者改成bob，所属组改成bob登录时所属的组

chgrp：更改文件所属群组

passwd：更改用户密码

高级键盘技巧

1.编辑命令行

bash 使用了一个名为Readline 的库 （供不同的应用程序共享使用的线程集合）来显示命令行的编辑。

⚠注意

下面的有些组合键（尤其是那些使用了Alt 的组合键）可能会被GUI（图形用户界面）识别为其他功能。当使用虚拟控制台时，所有组合键应用能够正常工作。

光标移动

--

组合键	作用
ctrl a	移动到行首
ctrl e	移动到行尾
ctrl f	向前一个字符
ctrl b	向后一个字符
alt f	向前一个字
alt b	向后一个字
ctrl l	清屏并把光标移动到左上角

修改文本

--

组合键	作用
ctrl d	删除光标处字符
ctrl t	对调光标和前面的字符
alt t	字与字对调
alt l	从光标到尾部 转成小写字母
alt u	从光标到尾部 转成大写字母

剪切和粘贴文本

--

组合键	作用
ctrl k	剪切从光标到行尾
ctrl u	剪切从光标到行首
alt d	剪切从光标到词尾

组合键	作用
alt backspace	剪切从光标到词头，如果在单词开头，剪切前一个单词
ctrl y	吧kill-ring缓冲区短的文本粘贴到光标位置

🔗 文章知识点与官方知识档案匹配，可进一步学习相关知识

CS入门技能树 > Linux进阶 > 新增用户 32027 人正在系统学习中

Linux命令行大全总结pdf 11-13
Linux命令行大全总结

Linux常用命令系统相关 FY_07170424的博客 8738
基本常用命令系统相关 1.查看网络ip地址 ip a 或者 ip addr 2.测试网络联通情况 ping 目标机器ip 3.查看当前系统时间 date 4.关机 shutdo...

Linux命令大全(超详细版)_第二范式的博客 6-10
less命令 1# 以分页的方式浏览文件信息(适合查看大文档),进入浏览模式 less 文件名 # 浏览模式快捷键 ↑#上一行 ↓#下一行 G #最后一...

最常用的linux命令大全(建议收藏)_linux命令行大全_海鸥爱上鱼的博客... 6-8
说明:service命令用于运行System V init脚本,这些脚本一般位于/etc/init.d文件下,这个命令可以直接运行这个文件夹里面的脚本,而不用加...

Linux常用命令大全（非常全面） 热门推荐 liuwei0376的专栏 17万+
目录 一、进程管理 二、系统信息 三、关机 (系统的关机、重启以及登出) 四、文件和目录 五、文件搜索 六、挂载一个文件系统 七、磁...

Linux常见命令 xiaochen6s的博客 2566
一、基础知识 1.1 Linux系统的文件结构 /bin 二进制文件，系统常规命令 /boot 系统启动分区，系统启动时读取的文件 /dev 设备文件 /etc...

Linux 命令大全(看这一篇就足够)_linux命令大全_张大炮的菜鸟码头的博... 6-9
2.9.3 测试网络连通性:ping 命令2.9.4 查看和设置网卡信息.ifconfig2.9.5 查看发送电子邮件:mail 命令2.9.6 查看登录用户历史信息:last 命...

Linux命令大全(最新)_《linux命令大全》_IT筱筱的博客 6-8
--进入后,操作界面有三种模式:命令模式(command mode)、插入模式(Insert mode)和底行模式(last line mode) 命令模式:-刚进入文件就...

Linux常用60个命令用法 qq_68724529的博客 9362
Linux命令，图片显示不出来，请看原文链接。 https://www.cnblogs.com/accwiz/p/15793166.html

Linux常用操作命令大全 huanglm_OneWholeLife的博客 6万+
Linux常用操作命令大全

Linux常用命令大全(非常全面)_江畔独步的博客 6-8
Linux和windows相比,很多管控都需要用命令来操作。 windows以直观的可视化的方式操作,特别适合在桌面端PC上操作执行相应的软件...

...总结大全,包含所有linux命令_linux命令行大全_chm_避凉闲庭的博客... 6-7
共计123个结果,如果要准确搜索您需要在命令前面加三个*号,注意后面不要加空格 这下只有6个结果了 文件传输 ***bye 说明: 中断FTP连...

linux命令行大全.pdf 03-15
linux命令行大全.pdf

Linux命令行大全.epub 09-07
Linux命令行大全

Linux命令大全完整版_熟悉的新风景的博客 5-25
1. linux系统管理命令adduser功能说明:新增用户帐号。 语法:adduser补充说明:在Slackware中,adduser指令是个script程序,利用交谈的方...

Linux面试必备20个常用命令_linux常用命令_袁袁袁满的博客-CSDN博 ... 6-10
第一章 什么是linux 第二章 linux的基础命令 1.pwd 命令 2.ls 命令 3.cd 命令 4.man 命令 5.grep 命令 6.find 命令 7.chmod 命令 8.ps 命令 ...

Linux命令行大全.pdf 04-21
学好Linux从命令行开始，只需要静下心来学习几天就可以学好了。

linux命令行大全(分模块、很清晰全面).pdf 12-15
linux命令行大全(分模块、很清晰全面)

linux 命令行 大全(常用)_linux命令_向着曙光前进的博客 6-5
linux 命令行 大全(常用) Linux常用命令大全(非常全!!!) 最近都在和Linux打交道,感觉还不错。我觉得Linux相比windows比较麻烦的就是很...

Linux常用命令行——表格形式，一目了然 01-20
Linux学习笔记（2） ——常用命令行 点个小星星，方便工作的时候使用吧，隔三行做分割线，命令行的 竖直线|，该文章打了好久...

Linux命令行技术大全 06-29
非常详细通俗易懂的一本linux命令行入门书籍，很适合初学者！

Linux命令行大全 07-08
Linux命令行大全 shell awk grep sed

linux命令行大全 08-15
linux 命令行 shell

Linux命令行大全 中文PDF详细版.pdf 最新发布 05-09
linux命令行 为什么使用命令行？ 你是否注意到，在电影中一个“超级黑客”坐在电脑前，从不摸一下鼠标， 就能够在30秒内侵入到超安全...

常见的linux命令行大全汇总 11-21
资源里有将近一百多个平时使用Linux常见的命令行，有命令行的使用方法和相应命令行执行后的结果

linux 命令行大全 思维导图 10-09
linux 命令行大全 思维导图

Ubuntu.Linux命令行大全.pdf 12-20
Ubuntu.Linux命令行大全.pdf

Linux命令大全完整版.pdf 07-27
主要包含linux 系统管理和设置命令、压缩备份命令、文档管理命令、文件传输命令、磁盘管理命令、磁盘维护命令、网络通讯命令、电...

“相关推荐”对你有帮助么？
🙄 非常没帮助 🙄 没帮助 😐 一般 😊 有帮助 😄 非常有帮助

关于我们 招贤纳士 商务合作 寻求报道 400-660-0108 kefu@csdn.net 在线客服 工作时间 8:30-22:00
公安备案号11010502030143 京ICP备19004658号 京网文〔2020〕 1039-165号 经营性网站备案信息 北京互联网违法和不良信息举报中心 家长监护 网络110报警服务 中国互联网举报中心 Chrome商店下载 账号管理规范 版权与免责声明 版权申诉 出版物许可证 营业执照
©1999-2023北京创新乐知网络技术有限公司