

secureCRT终端常用命令

1.1 ls

[语法]: `ls -RadCxmlnogrtucpFbqisf1`

[说明]: `ls` 命令列出指定目录下的文件，缺省目录为当前目录 `./`，缺省输出顺序为纵向按字符顺序排列。

- `-R` 递归地列出每个子目录的内容
- `-a` 列出所有文件，包括第一个字符为“.”的隐藏文件
- `-d` 若后面参数是目录，则只列出目录名而不列出目录内容，常与`-l`选项连

用以显示目录状态。

- `-C` 输出时多列显示
- `-x` 横向按字符顺序排列
- `-m` 输出按流式格式横向排列，文件名之间用逗号(,)分隔
- `-l` 长列表输出，显示文件详细信息，每行一个文件，从左至右依次是：

文件存取模式 链接数 文件主 文件组 文件字节数 上次修改时间

其中文件存取模式用10个字母表示，从左至右的意义如下：

第一个字母表示文件种类，可以是以下几种情况：

`d` 为目录文件

`l` 为链接

`b` 为块文件

`c` 为字符型文件

`p` 为命名管道（FIFO）

- 为普通文件

后面9个字母分别表示文件主、同组用户、其他用户对文件的权力，用`r`表示可读，`w`表示可写，`x`表示可执行。如果是设备文件，则在文件字节数处显示：主设备 从设备。

- `-n` 与`-l`选项相同，只是文件主用数字(即UID)显示，文件组用数字

(即GID)表示

- `-o` 与`-l`选项相同，只是不显示文件组
- `-g` 与`-l`选项相同，只是不显示文件主
- `-r` 逆序排列
- `-t` 按时间顺序排列而非按名字
- `-u` 显示时间时使用上次访问时间而非上次修改时间
- `-c` 显示时间时使用上次修改i节点时间而非上次修改时间
- `-p` 若所列文件是目录文件，则在其后显示斜杠(/)
- `-F` 在目录文件后加"/"，在可执行文件后加"*"
- `-b` 文件名中若有非打印字符，则用八进制显示该字符

- -q 文件名中的打印字符用?表示
- -i 显示节点号
- -s 显示文件长度时使用块长度而非字节长度
- -f 将后面的参数解释为目录并列出的每一项
- -1 每行仅列一项

[例子]:

ls 列出当前目录下的文件

ls -al /bin 以长列表的形式列出目录 /bin 下的所有文件，包括隐藏文件

1.2 cd

[语法]: **cd** [目录]

[说明]: 本命令用于改变当前的工作目录，无参数时使用环境变量HOME 作为其参数，HOME 一般为注册时进入的路径。

[例子]:

cd 回到注册进入时的目录

cd /tmp 进入 /tmp 目录

cd ../ 进入上级目录

1.3 mkdir

[语法]: **mkdir -m** 模式 目录名

- -m 按指定存取模式建立目录
- -p 建立目录时建立其所有不存在的父目录

[例子]:

mkdir tmp 在当前目录下建立子目录 tmp

mkdir -m 777 /tmp/abc 用所有用户可读可写可执行的存取模式

建立目录 /tmp/aaa，存取模式参看命令 chmod

mkdir -p /tmp/a/b/c 建立目录 /tmp/a/b/c，若不存在目录 /tmp/a

及/tmp/a/b 则建立之

1.4 rmdir

[语法]: **rmdir -p** 目录名

- -p 删除所有已经为空的父目录
- -s 当使用-p 选项时，出现错误不提示

[例子]:

`rmdir /tmp/abc` 删除目录 `/tmp/abc`

`rmdir -p /tmp/a/b/c` 删除目录 `/tmp/a/b/c`，若目录 `/tmp/a/b`

及 `/tmp/a` 空，则删除

1.5 cp

[语法]: `cp -p` 文件 1 [文件 2 ...] 目标

[说明]: 将文件1(文件2 ...)拷贝到目标上，目标不能与文件同名，若目标是文件名，则拷贝的文件只能有一个，若目标是目录，则拷贝的文件可以有多个，若目标文件不存在，则建立这个文件，若存在，则覆盖其以前的内容，若目标是目录，则将文件拷贝到这个目录下。

- - `i` 在覆盖已存在文件时作提示，若回答 `y` 则覆盖，其他则中止
- - `p` 不仅拷贝文件内容，还有修改时间，存取模式，存取控制表，但不拷贝

UID 及 GID

- - `r` 若文件名为目录，则拷贝目录下所有文件及子目录和它们的文件，此时

目标必须为目录

[例子]:

`cp file1 file2` 将文件 `file1` 拷贝到文件 `file2`

`cp file1 file2 /tmp` 将文件 `file1` 和文件 `file2` 拷贝到目录 `/tmp` 下

`cp -r /tmp /mytmp` 将目录 `/tmp` 下所有文件及其子目录拷贝至目录 `/mytmp`

1.6 rm

[语法]: `rm -f` 文件...

或 `rm -r -f` 目录名... [文件]&

- `-f` 删除文件时不作提示
- `-r` 递归地删除目录及其所有子目录
- `-i` 删除文件之前先作提示

[例子]:

`rm file1` 删除文件 `file1`

`rm -i /tmp/*` 删除目录 `/tmp` 下的所有文件

`rm -r /mytmp` 递归地删除目录 `/mytmp`

1.7 chmod

[语法]: `chmod [-R] 模式 文件...`

或 `chmod [ugoa] {+|-|=} [rwxst] 文件...`

- `chmod nnnn file` , `n`为0-7的数字, 意义如下:

4000 运行时可改变UID

2000 运行时可改变GID

1000 置粘着位

0400 文件主可读

0200 文件主可写

0100 文件主可执行

0040 同组用户可读

0020 同组用户可写

0010 同组用户可执行

0004 其他用户可读

0002 其他用户可写

0001 其他用户可执行

`nnnn` 就是上列数字相加得到的, 例如 `chmod 0777 file` 是指将文件 `file` 存取权限置为所有用户可读可写可执行。

- `-R` 递归地改变所有子目录下所有文件的存取模式

`u` 文件主

`g` 同组用户

`o` 其他用户

`a` 所有用户

`+` 增加后列权限

`-` 取消后列权限

`=` 置成后列权限

`r` 可读

`w` 可写

`x` 可执行

`s` 运行时可置UID

`t` 运行时可置GID

[例子]:

chmod 0666 file1 file2 将文件 file1 及 file2 置为所有用户可读可写

chmod u+x file 对文件 file 增加文件主可执行权限

chmod o-rwx 对文件file 取消其他用户的所有权限

1.8 find

[语法]: **find** 路径名... 表达式

[说明]: **find** 命令递归地遍历指定路径下的每个文件和子目录，看该文件是否能使表达式值为真，以下 **n** 代表一个十进制整数，**+n** 代表打印 **n**，**-n** 代表小于 **n**，下面是合法表达式说明：

- **-name** 模式 文件名与模式匹配则为真，(\ 为转意符)
- **-perm [-]**八进制数 文件存取模式与八进制数相同则为真若有 **-** 选项，则文件存

取模式含有八进制数规定模式即为真

- **-size n[c]** 文件块长度为 **n** 则真(一块为512字节)，若

有 **c** 选项，则文件字节长度为 **n** 则真

- **-atime n** 若文件的最近访问时间为 **n** 天前则为真，

find 命令将改变其访问的目录的访问时间

- **-mtime n** 若文件的最近修改时间为 **n** 天前则为真
- **-ctime n** 若文件状态为 **n** 天前改变则为真
- **-exec** 命令 { } \; 若命令返回值为0则真，{ }内为命令参数，

此命令必须以 \; 为结束

- **-ok** 命令 { } \; 与 **exec** 相同，只是在命令执行前先提示，若

回答 **y** 则执行命令

- **-print** 显示输出使表达式为真的文件名
- **-newer** 文件 若文件的访问时间比 **newer** 指定的文件新则真
- **-depth** 先下降到搜索目录的子目录，然后才至其自身
- **-mount** 仅查找包含指定目录的文件系统
- **-local** 文件在当前文件系统时为真
- **-type c** 文件类型为 **c** 则真，**c** 取值可为 **b**(块文件) **c**(字符文件)

d(目录) **l**(符号链接) **p**(命名管道) **f**(普通文件)

(表达式) 表达式为真则真

- **-links n** 文件链接数为 **n** 时为真
- **-user** 用户 当文件属于用户时为真，用户可用数字表示UID
- **-nouser** 当文件不属于 **/etc/passwd** 中的一个用户时为真
- **-group** 文件组 当文件属于文件组时为真，文件组可用数字表示GID
- **-nogroup** 当文件不属于 **/etc/group** 中的一个组时为真
- **-fstype** 类型 当文件所属文件系统类型为指定类型时真
- **-inum n** 当文件 **i** 节点号为 **n** 时为真
- **-prune** 当目录名与模式匹配时，不再搜索其子目录

可以用逻辑操作符将简单表达式连接成复杂表达式

逻辑操作符有 ! 表示非操作， -o 表示或操作，两个表达式并列则表示与操作

[例子]:

```
find / -name find* -print
```

从根目录开始搜索文件名如 find* 的文件并显示之

```
find ./ -exec sleep{1}\; -print
```

每秒显示一个当前目录下的文件

```
find $HOME (-name a.out -o -name '*.o') -atime +7 -exec rm {} \;
```

从\$HOME目录开始搜索，删除所有文件名为a.out 或 *.o 且访问时间在7天前的文件

1.9 grep

[语法]: **grep** [选项] 模式 [文件...]

[说明]: 在指定的文件中搜索模式，并显示所有包含模式的行，模式是一个正则表达式，在使用正则表达式时，最好将其引在单引号(') 中，若指定文件为缺省，则使用标准输入，正则表达式可以是：

- . 匹配任意一个字符
- ○ 匹配0个或多个*前的字符
- ^ 匹配行开头
- \$ 匹配行结尾
- [] 匹配[]中的任意一个字符，[]中可用 - 表示范围，

例如[a-z]表示字母a 至z 中的任意一个

\ 转意字符

命令中的选项为：

- -b 显示块号
- -c 仅显示各指定文件中包含模式的总行数
- -i 模式中字母不区分大小写
- -h 不将包含模式的文件名显示在该行上
- -l 仅显示包含模式的文件名
- -n 显示模式所在行的行号
- -s 指定文件若不存在或不可读，不提示错误信息
- -v 显示所有不包含模式的行

[例子]:

grep 'good' * 在所有文件中搜索含有字符串 good 的行

grep '^myline' mytext 在文件mytext中搜索行首出现myline字符串的行

1.10 vi

[语法]: **vi -wn** 文件...

[说明]: **vi** 是一个基于行编辑器 **ex** 上的全屏幕编辑器，可以在 **vi** 中使用 **ex**, **ed** 的全部命令，**vi** 选项中 **-wn** 指将编辑窗口大小置为 **n** 行，**-R** 为将编辑的文件置为只读模式，**vi** 工作模式分为命令模式和输入模式，一般情况下在命令模式下，可敲入 **vi** 命令，进入输入模式下时可以编辑要编辑的文本，命令 **a A i l o O c C s S R** 可进入输入模式，在输入模式下按 **ESC** 键可推出输入模式，回到命令模式，在命令模式中敲入: 命令，则可进入 **ex** 方式，在屏幕底部出现提示符: ，此时可使用任意 **ex** 命令，屏幕底行也用来作 **/ ? !** 命令的提示行，大多数命令可以在其前面加数字，表示命令执行的重复次数，下面简单介绍一下 **vi** 的命令集，**^** 表示 **(CTRL)** 键

- **^B** 退回前一页，前面加数字表示重复次数，每次换页时保留上一页的两行
- **^D** 在命令模式下，表示下滚屏幕的一半，在输入模式下，表示回退至左边的自动缩进处
- **^E** 显示屏幕底线之下的一行
- **^F** 前进一页，前面加数字表示重复次数，每次换页时保留上一页的两行
- **^G** 显示当前文件名，当前行号和文件总行数，并用百分号当前行在整个文件中的位置
- **^H**(退格) 在命令模式下，光标左移一格；在输入模式下，删去前面的字符
- **^I**(**TAB**) 在输入模式下，产生一串空格
- **^J**(**LF**) 光标下移一行
- **^L** 刷新屏幕，即将屏幕重新显示
- **^M**(回车) 在命令模式下，光标移动至下行开头，在输入模式下，开辟一新行
- **^N** 光标下移一行
- **^P** 光标上移一行
- **^Q** 在输入模式下，将其后的非打印字符作为正文插入
- **^R** 刷新屏幕
- **^U** 屏幕上滚一半，前面加数字时表示上滚的行数，此数字对以后的 **^D ^U** 命令有效
- **^V** 在输入模式下，将其后的非打印字符作为正文插入
- **^W** 在输入模式下，使光标回退一个字
- **^Y** 显示屏幕底线之上的一行
- **^Z** 暂停编辑，退回上层 **Shell**
- **^[**(**ESC**) 退出输入模式，回到命令模式
- **!** 暂时退出编辑，执行 **Shell** 命令

(双引号) 用于标志有名缓冲区，编号缓冲区 1-9 用于保存被删去的正文，字

母名缓冲区 **a-z** 供用户存放自定义的正文，将光标移动到当前行尾，前加数字则表示前移行数，如 2 表示移动到下一行行尾

- **%** 将光标移动到配对的小括号 **()** 或大括号 **{}** 上去
- **(** 退回句子开头
- **)** 前移到句子开头
- **o** 退到上一行第一个非空格字符
- **.** 重复上一次改变缓冲区内容的命令
- **/** 模式 向前搜索模式，将光标移动到模式出现处，模式是一个正规表达式，(参看 **grep**)
- **:** 在屏幕底部提示: ，其后可使用 **ex** 命令
- **?** 功能同 **/**，但方向是向前查找

- [[光标回退至前一节分界处
- \ 转意符
-]] 光标前移至节分界处
- ^ (不是CTRL) 光标移至当前行第一个非空字符上
- ‘连续两个”表示将光标移至其移动前的位置，’后跟字母表示光标字母标记的行首(参看 m 命令)
- A 在行尾插入正文，进入输入模式
- B 光标回退一个字
- C 替换光标后的内容
- D 删除光标后的内容
- E 光标前移到字尾
- F 字符 在当前行向左查找指定字符
- G 光标移动到其前面数字指定的行，若未指定则移动到最后一行
- H 光标移动到屏幕顶行，若前面有数字，则移动到屏幕上该数字指定的行
- I 在行开头插入正文
- J 连接两行，若前面有数字则连接数字指定的行
- L 光标移动到屏幕底行，若前面有数字，则移动到屏幕底线往上数该数字指定的行
- M 光标移动到屏幕中线
- N 使用模式查找/或?时，重复找下一个匹配的模式，但方向与上次相反，其功能同 n，但方向相反
- O 在当前行上开辟一新行
- P 将上次被删除的正文插入光标前面，可在其前面加缓冲区编号，编号1-9用于保存被删去的正文，字母名缓冲区a-z供用户存放自定义的正文

Q 从vi 推出进入ex命令状态

- R 替换字符串
- S 替换整行
- T 字符 向左查找字符
- U 将当前行恢复至第一次修改前的状态
- W 光标移至下一个字首
- X 删除光标前的字符
- Y 将当前行存入无名缓冲区，前面加数字表示存入的行数，也可用有名缓冲区来保存，以后可用命令p或P将其取出
- ZZ 存盘退出vi
- a 光标后插入正文
- b 光标回退至上一个字首
- cw 替换当前字
- c) 替换当前句子
- dw 删除一个字
- dd 删除一行
- e 光标移到下一个字末
- f 字符 在当前行向前查找字符
- h 光标左移一格
- i 在光标前插入正文

- j 光标下移一行
- k 光标上移一行
- l 光标右移一格
- m 字母 用字母标记当前行，以后可用 '字母使光标移动到当前行，(参看'命令)
- n 重复上次 / 或 ? 命令
- o 在当前行下开辟一新行
- p 将用户缓冲区内容放到光标位置(参看P命令)
- r 替换当前字符
- s 用一串字符替换当前字符
- t 字符 光标移动至字符前
- u 取消上次操作
- w 光标移至下一字首
- x 删除当前字符
- yw 将当前字存入无名缓冲区，前面可加x，表示存入名字为x的有名缓冲区(x为a-z)，也可加数字表示存入的字数，以后可用P或p命令取出
- yy 将当前行存入无名缓冲区，用法参看yw
- { 光标移动至前一段开头
- | 光标移至行首，若前面加数字，则移到数字指定行的行首
- } 光标移至下一段开头

在：提示符下，常用命令如下：

- : w 当前文件存盘
- : w! 强制存盘
- : w 文件 将内容写入指定文件
- : w! 文件 强制写入指定文件
- : x, yw 文件 将 x至 y 行写入指定文件中
- : r 文件 将文件读到光标位置
- : r! 命令 将系统命令的输出读到光标位置
- : q 退出编辑
- : q! 强制退出
- : x 与命令ZZ相同
- : e 文件名 编辑另一文件
- : e! 重新编辑文件，放弃任何改变
- : sh 执行sh，结束后回到编辑
- : ! 命令 执行命令后回到编辑
- : n 编辑下一文件
- : n 文件表 重新定义待编辑文件表
- : set 设置 vi 的选项，例如 set nu 表示每行前显示行号，在选项前加no则表示清除该选项，例如 set nonu 表示每行前不显示行号，下面是一些常用的选项:
- ai 自动缩进
- aw 编辑下一文件前自动存盘
- ic 查找字符串时不区分大小写
- nu 每行前显示行号
- sm 输入)及}时显示与之配对的(或 {
- slow 插入时延迟屏幕刷新
- ws 使查找能绕过文件尾从头进行
- wa 写文件之前不作对文件的检查

