Linux 常用工具手册

作者：kenshinl

时间：Last Update: October 6, 2022

版本：0.1

封面图片作者：Elina Bernpaintner

Linux Command Handbook。——kenshinl

目录

第 1 章 序章

这本工具书主要用来帮助新手快速熟悉 Linux 下的常用工具。本书的 Linux 程序选取的原则有如下几条。

1. 选取的程序需要是大部分 Linux 发行版已经预装的，不需要用户再自行安装。
2. 选取的程序只介绍常用的参数，对于不常用的参数通过提供手册链接让用户自行查阅。
3. 介绍的程序使用发行已经较久的版本，不介绍近 2 年才发行的版本和特性。

第 2 章 文本操作程序

本章节介绍常用的 Linux 文本操作程序。

2.1 cut

cut 是一个文本处理程序, 用来取出每个文件中的每一行的某个部分。

版本：cut (GNU coreutils) 8.30

2.1.1 用法和参数

```
cut OPTION [FILE]
        OPTION 参数是必选的，FILE 参数是可选的

  -b, --bytes=LIST
        选择特定的字节序列

  -c, --characters=LIST
        选择特定的字符序列

  -d, --delimiter=DELIM
        使用 DELIM 而不是 TAB 作为字段分隔符，默认使用 TAB 作为字段分隔符

  -f, --fields=LIST
        选择特定的字段序列；除非指定了 -s 参数，否则会打印所有
        不包含分隔符的行

  -n    (ignored)

  --complement
        补充所选的字节、字符或字段集

  -s, --only-delimited
        不打印不包含分隔符的行

  --output-delimiter=STRING
        使用 STRING 作为输出的字段分隔符，默认使用和输入相同的字段分隔符

  -z, --zero-terminated
        使用 NUL 而不是换行符作为行分隔符，不能单独使用
```

2.1.2 使用示例

使用的测试文件 test.txt 内容如下。

```
abc cde
bdf fgh
abc hhh
```

```
# 输出每行的第 1 个字节
$ cut -b 1 test.txt
```

```
a
b
a

# 输出每行的第1到3字节
$ cut -b 1-3 test.txt
abc
bdf
abc

# 输出每行的第5个字节及之后的内容
$ cut -b 5- test.txt
cde
fgh
hhh

# 输出每行的第1到3个字符，-b与-c参数的用法类似
$ cut -c 1-3 test.txt
abc
bdf
abc

# 以空格为分隔符将每一行切割为多列，输出第二列
$ cut -d " " -f 2 test.txt
cde
fgh
hhh

# 任务：假设服务器所有日志存在一个文件
# 假设
$ find . -name "*.log" | xargs file | grep "text" | cut -d: -f1 | xargs cat | grep "
    on_bpass_season_end"
```

2.1.3 练习

假设服务器所有日志存储在一个目录/server_log/下，文件名格式为 module_name.log，日志文件大小达到 10M 之后会将旧日志重命名为 module_name.log.1, module_name.log.2 等，当旧日志数量达到 10 个之后，会将旧日志进行压缩，压缩文件命名为 module_name.log.tar.gz，现在要求查找 http 模块中所有包含”UserName” 的日志，压缩日志除外。

```
find . -name "http.log*" | xargs file | grep "text" | cut -d: -f1 | xargs cat | grep "UserName"
```

2.1.4 文档

- 在线手册

2.2 rev

rev 程序的功能是按字符反转文本行。rev 将指定的文件复制到标准输出并且颠倒每一行中字符的顺序。如果没有指定文件，则读取标准输入。rev 使用内为宽字符行分配的内存缓冲区，如果输入文件非常大并且没有换行符的话，rev 程序可能运行失败。

2.2.1 用法和参数

```
uniq [OPTION]... [INPUT [OUTPUT]]
rev [option] [file...]
```

2.2.2 使用示例

用来测试的文件 test.txt 内容

```
abc def
这是一行测试 rev 的文本
```

```
$ rev test.txt
fed cba
本文的 ver 试测行一是这
```

2.2.3 文档

- 在线手册

2.3 uniq

uniq 从输入文件或者标准输入中过滤**相邻**的匹配行，写入输入文件或者标准输入，如果没有添加任何参数的话，匹配的行都会合并到首次出现的行中。uniq 在日常中主要用来做文本重复行的过滤、统计、去重三个操作。

2.3.1 用法和参数

```
uniq [OPTION]... [INPUT [OUTPUT]]

  -c, --count
        统计每一行出现的次数，出现次数并作为前缀添加到首次出现的行前

  -d, --repeated
        输出所有重复的行，每组重复的行只输出一次

  -D    输出所有重复的行

  --all-repeated[=METHOD]
        和 —D 类似，但是允许使用空行来分隔不同的重复组
        METHOD={none(default),prepend,separate}

  -f, --skip-fields=N
        不比较每行的前 N 个字段

  --group[=METHOD]
        展示所有项，并且使用空行进行分割
        METHOD={separate(default),prepend,append,both}

  -i, --ignore-case
        在比较的时候忽略大小写

  -s, --skip-chars=N
        跳过比较每行的前 N 个字符

  -u, --unique
        只打印出现过一次的行

  -z, --zero-terminated
        使用 NUL 而不是换行符作为行分割符

  -w, --check-chars=N
        只比较每行的最多前 N 个字符
```

2.3.2 使用示例

下面测试中使用的文件。

uniq.txt

```
aaaa
bbbb
cccc
aaaa
bbbb
aaaa
```

uniq2.txt

```
1 aaaa
2 bbbb
3 cccc
4 aaaa
5 bbbb
6 aaaa
```

uniq3.txt

```
aaa12
aaa23
bbb34
```

uniq 各参数使用实例。

```
$ cat uniq.txt | sort | uniq
aaaa
bbbb
cccc

$ cat uniq.txt | sort | uniq -c
     3 aaaa
     2 bbbb
     1 cccc

$ cat uniq.txt | sort | uniq -d
aaaa
bbbb

$ cat uniq.txt | sort | uniq -D
aaaa
aaaa
aaaa
bbbb
bbbb

$ cat uniq.txt | sort | uniq --all-repeated=separate
aaaa
aaaa
aaaa

bbbb
bbbb

$ cat uniq2.txt | rev | sort | rev | uniq --skip-fields=1 -D
1 aaaa
4 aaaa
6 aaaa
2 bbbb
5 bbbb

$ cat uniq.txt | sort | uniq --group=append
aaaa
aaaa
aaaa

bbbb
bbbb

cccc
```

```
$ cat uniq2.txt | rev | sort | rev | uniq -s 2 -D
1 aaaa
4 aaaa
6 aaaa
2 bbbb
5 bbbb

$ cat uniq.txt | sort | uniq -u
cccc

$ cat uniq3.txt | uniq -w 3 -c
      2 aaa12
      1 bbb34
```

2.3.3 文档

第 3 章 性能监控

本章介绍 Linux 下的性能监控工具

3.1 使用 tcmalloc 编写内存接口

3.2 参考资料

本节列一下参考资料。

第 4 章 系统属性查看

本章节主要介绍一些查看系统属性、文件熟悉的 Linux 程序。

4.1 file

file 程序会测试每个参数来尝试对文件进行分类。按如下顺序执行三组测试：文件系统测试、魔术测试、语言测试。按这个顺序，哪个测试成功了，就打印对应的文件类型。

打印的文件类型通常为下面几类:

- text 文本类型，只包含可打印字符和一些控制字符。
- executable 可执行文件类型。
- data 数据类型，表示除上面两类外的其他类型，文件内容通常是二进制或不可打印的格式。

file 命令至少从 Research Version 4 版本就被添加到了 UNIX 系统中，System V 系统版本引入了魔法类型的外部列表，稍微减慢了 file 程序的速度但是使得程序更加灵活。

版本：file-5.38

4.1.1 用法和参数

```
file [-bcdEhiklLNnprsSvzZ0] [--apple] [--exclude-quiet]
     [--extension] [--mime-encoding] [--mime-type] [-e testname]
     [-F separator] [-f namefile] [-m magicfiles] [-P name=value]
     file ...

    -b, --brief
      不要将文件名添加到输出行（简要模式）

    -i, --mime
           输出文件的 mime type 字符串。

    --mime-type, --mime-encoding
           和 -i 类似，但是只打印指定类型的元素

    -p, --preserve-date
           在支持 utime(3) 或者 utimes(2) 的系统上，让 file 命令在执行的
           时候不修改文件的访问时间

    -r, --raw
           不将不可大打印的字符转换成八进制，一般而言，file 指令会将不可打印
           的字符转换成八进制表示

    -0, --print0
           在文件名之后输出一个 '\0' 字符，便于使用 cut 指令进行分割，这个
           参数不会影响分隔符的正常输出

           如果这个参数被重复多次，file 会在输出的每个字段之后都加上一个 NUL('\0')
```

4.1.2 使用示例

使用到的测试文件如下。

file.txt

```
file cmd test.
```

各参数使用例子。

```
$ file file.txt
file.txt: ASCII text

$ file -b file.txt
ASCII text

$ file -i file.txt
file.txt: text/plain; charset=us-ascii

$ file --mime-encoding file.txt
file.txt: us-ascii

$ file file.txt | cut -d '' -f 1
file.txt: ASCII text

$ file file.txt -0 | cut -d '' -f 1
file.txt
```

4.1.3 文档

- 在线手册