

xargs

给其他命令传递参数的一个过滤器

补充说明

xargs 命令是给其他命令传递参数的一个过滤器，也是组合多个命令的一个工具。它擅长将标准输入数据转换成命令行参数。xargs 能够处理管道或者 stdin 并将其转换成特定命令的命令参数。xargs 也可以将单行或多行文本输入转换为其他格式，例如多行变单行，单行变多行。xargs 的默认命令是 echo。空格是默认定界符。这意味着通过管道传递给 xargs 的输入将会包含换行和空白，不过通过 xargs 的处理，换行和空白将被空格取代。xargs 是构建单行命令的重要组成部分之一。

xargs 命令用法

xargs 用作替换工具，读取输入数据重新格式化后输出。

定义一个测试文件，内有多行文本数据：

```
cat test.txt

a b c d e f g
h i j k l m n
o p q
r s t
u v w x y z
```

多行输入单行输出：

```
cat test.txt | xargs

a b c d e f g h i j k l m n o p q r s t u v w x y z
```

使用 -n 进行多行输出

-n 选项 多行输出：

```
cat test.txt | xargs -n3

a b c
d e f
g h i
j k l
m n o
p q r
s t u
v w x
y z
```

使用 -d 分割输入

-d 选项 可以自定义一个定界符：

```
echo "nameXnameXnameXname" | xargs -dX  
  
name name name name
```

结合 **-n** 选项使用：

```
echo "nameXnameXnameXname" | xargs -dX -n2  
  
name name  
name name
```

读取 **stdin**

读取 **stdin** 将格式化后的参数传递给命令

假设一个命令为 `sk.sh` 和一个保存参数的文件 `arg.txt`

```
#!/bin/bash  
#sk.sh 命令内容，打印出所有参数。  
  
echo $*
```

`arg.txt` 文件内容：

```
cat arg.txt  
  
aaa  
bbb  
ccc
```

结合 **-I** 选项

`xargs` 的一个选项 **-I**，使用 **-I** 指定一个替换字符串`{}`，这个字符串在 `xargs` 扩展时会被替换掉，当 **-I** 与 `xargs` 结合使用，每一个参数命令都会被执行一次：

```
cat arg.txt | xargs -I {} ./sk.sh -p {} -l  
  
-p aaa -l  
-p bbb -l  
-p ccc -l
```

复制所有图片文件到 `/data/images` 目录下：

```
ls *.jpg | xargs -n1 -I cp {} /data/images
```

结合 **find** 命令使用

xargs 结合 **find** 使用

用 `rm` 删除太多的文件时候，可能得到一个错误信息：`/bin/rm Argument list too long`. 用 `xargs` 去避免这个问题：

```
find . -type f -name "*.log" -print0 | xargs -0 rm -f
```

xargs -0 将 \0 作为定界符。

统计一个源代码目录中所有 php 文件的行数：

```
find . -type f -name "*.php" -print0 | xargs -0 wc -l
```

查找所有的 jpg 文件，并且压缩它们：

```
find . -type f -name "*.jpg" -print | xargs tar -czvf images.tar.gz
```

打印出执行的命令

结合 -t 选项可以打印出 xargs 执行的命令

```
ls | xargs -t -I{} echo {}
```

会输出当前目录下的文件列表和执行的 echo 命令

使用 -p 选项确认执行的命令

-p 选项会在执行每一个命令时弹出确认，当你需要非常准确的确认每一次操作时可以使用 -p 参数，比如，查找当前目录下 .log 文件，每一次删除都需要确认：

```
find . -maxdepth 1 -name "*.log" | xargs -p -I{} rm {}
```

执行多个命令

使用 -I 选项可以让 xargs 执行多个命令

```
cat foo.txt
one
two
three

cat foo.txt | xargs -I % sh -c 'echo %; mkdir %'
one
two
three

ls
one two three
```

其他应用

xargs 其他应用

假如你有一个文件包含了很多你希望下载的 URL，你能够使用 xargs 下载所有链接：

```
cat url-list.txt | xargs wget -c
```

子 Shell 与 Subshells

运行一个 shell 脚本时会启动另一个命令解释器，就好像你的命令是在命令行提示下被解释的一样，类似于批处理文件里的一系列命令。每个 shell 脚本有效地运行在父 shell(parent shell) 的一个子进程里。这个父 shell 是指在一个控制终端或在一个 xterm 窗口中给你命令指示符的进程。

```
cmd1 | ( cmd2; cmd3; cmd4 ) | cmd5
```

如果 cmd2 是 `cd /` 那么就会改变子 Shell 的工作目录，这种改变只是局限于子 shell 内部，cmd5 则完全不知道工作目录发生的变化。子 shell 是嵌在圆括号 () 内部的命令序列，子 Shell 内部定义的变量为局部变量。

子 shell 可用于为一组命令设定临时的环境变量：

```
COMMAND1
COMMAND2
COMMAND3
(
  IFS=:
  PATH=/bin
  unset TERMINFO
  set -C
  shift 5
  COMMAND4
  COMMAND5
  exit 3 # 只是从子 shell 退出。
)
# 父 shell 不受影响，变量值没有更改。
COMMAND6
COMMAND7
```

reference

- <https://shapeshed.com/unix-xargs/>

From:

<https://rd.irust.top/> - 学习笔记

Permanent link:

<https://rd.irust.top/doku.php?id=command:xargs>

Last update: **2021/10/15 14:58**

