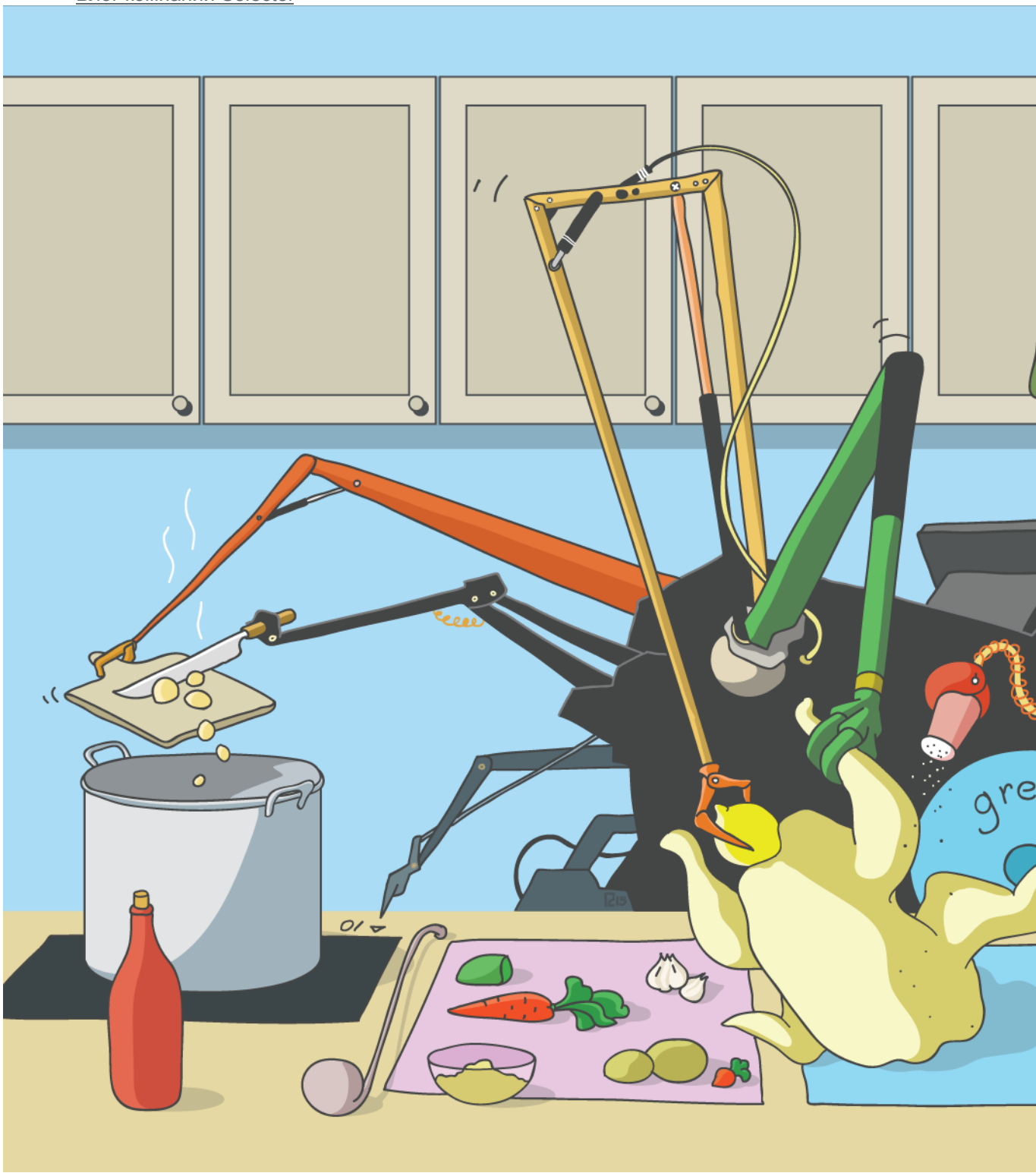
Xargs: многообразие вариантов использования

- Блог компании Selectel



Об утилите xargs написано очень много — что можно написать еще? Но если, что называется, копнуть поглубже, то выясняется, что во многих публикациях излагаются лишь самые основы, но нет главного: не объясняется, как можно

применять xargs в реальной практике. Статей с разбором сложных и нетривиальных вариантов применения этого весьма полезного для системного администратора инструмента, к сожалению, очень мало. Именно поэтому мы написали свою статью и постарались включить в нее как можно больше примеров использования xargs для решения различных проблем.

Сначала мы рассмотрим принцип работы xargs и разберем примеры попроще, а затем перейдем к разбору сложных и интересных кейсов.

Вспоминаем основы

Принцип работы xargs можно описать следующим образом: программа берет данные из стандартного ввода или из файла, разбивает их в соответствии с указанными параметрами, а затем передает другой программе в качестве аргумента.

В общем виде синтаксис команды xargs можно представить так:

```
[команда_генератор_списка] | xargs [опции_xargs] [команда]
```

Рассмотрим, как все это работает, на материале простых и хрестоматийных примеров.

Удаление файлов

Одна из самых частых ситуаций, в которых используется xargs — удаление файлов, найденных при помощи команды find.

Представим себе следующую ситуацию: имеется директория, в которой хранится большое количество файлов. Из нее нужно удалить файлы определенного типа (в нашем примере — файлы с расширением *.sh). Чтобы осуществить эту операцию, нужно передать xargs вывод команды find, и к файлам с указанным расширением будет применена команда -rm:

```
$ ls
one.sh one.py two.sh two.py

$ find . -name "*.sh"| xargs rm -rf

$ ls
one.py two.py
```

Отметим, что операцию удаления файлов можно осуществить и без xargs,

а с помощью команды

```
$ find . -name "*.sh" -exec rm -rf '{}' \
```

Описанный способ не сработает, если в имени одного из удаляемых файлов содержится пробел. Имя, состоящее из двух слов, разделенных пробелом, не будет воспринято как единое целое.

Проиллюстрируем это следующим примером:

```
$ ls
new file.sh one.sh one.py two.sh two.py

$ find . -name "*.sh"| xargs rm -rf

$ ls
new file.sh one.py two.py
```

Как видим, файл, в имени которого имеется пробел, не был удалён.

Чтобы решить эту проблему, используется опция print0 для команды find и опция -0 для команды xargs. Она заменяет стандартный разделитель (перенос строки на нуль-символ (\x0), который и означает конец хранимой строки:

```
$ find . -name "*.sh" -print0 | xargs -0 rm -rf
```

Xargs может также помочь, например, быстро удалить все временные файлы, имеющие расширение tmp:

```
$ find /tmp -name "*.tmp"| xargs rm
```

Сжатие файлов

Сжать все файлы в текущей директории с помощью gzip можно, введя следующую команду:

```
$ ls | xargs -p -l gzip
```

Рассмотрим еще один пример: сжатие с помощью tar всех файлов с расширением *.pl:

```
$ find . -name "*.pl" | xargs tar -zcf pl.tar.gz
```

Переименование файлов

С помощью xargs можно осуществлять массовое переименование файлов. Представим себе, что у нас есть группа файлов с расширением *.txt, и нам нужно заменить это расширение на *.sql. Это можно сделать при помощи xargs и потокового текстового редактора sed:

```
$ ls | sed -e "p;s/.txt$/.sql/" | xargs -n2 fmv
```

В результате ее выполнения на консоль будет выведен список переименованных файлов.

С помощью xargs можно также добавлять к дополнительные элементы к именам файлов (например, дату):

```
$ ls | xargs -I FILE mv {} <...>-{}
```

Вместо <..> можно подставить всё, что угодно.
Фигурные скобки {} в этом примере означают «текущий аргумент» (т.е. текущее имя файла).

Изменение прав для папок и файлов

С помощью xargs можно также ускорить процесс смены прав на файлы и папки для определенного пользователя или группы. Предположим, нам нужно найти все папки пользователя root и заменить их владельца на temp. Эта операция осуществляется при помощи команды:

```
$ find . -group root -print | xargs chown temp
```

Чтобы найти все папки группы root и заменить группу на temp, используется команда:

```
$ find . -group root -print | xargs chgrp temp
```

Xargs и find: сложные операции

С помощью команд find и xargs можно выполнять и более сложные операции. Вот так, например, можно удалить временные файлы, созданные более 7 дней назад:

```
$ find /tmp -type f -name '*' -mtime +7 -print0 | xargs -0 rm -f
```

А вот так — принудительно остановить процессы, которые уже работают больше 7 дней:

```
$ find /proc -user myuser -maxdepth 1 -type d -mtime +7 -exec
basename {} \; | xargs kill -9
```

Xargs и cut

Xargs довольно часто используется в сочетании с командой cut, позволяющей вырезать строки из текстовых файлов. Рассмотрим некоторые практические примеры. С помощью приведённой ниже команды на консоль будет выведен список всех пользователей системы:

```
$ cut -d: -f1 < /etc/passwd | sort | xargs echo
```

А команда вида

```
file * | grep ASCII | cut -d":" -f1 | xargs -p vim
```

будет последовательно открывать файлы для редактирования в vim.
Обратим внимание на опцию -p. Благодаря ей команда будет выполняться в интерактивном режиме: перед открытием каждого файла будет запрашиваться подтверждение (y/n).

В заключение приведём ещё один сложный и интересный пример — рекурсивный поиск файлов самого большого размера в некоторой директории:

```
$ find . -type f -printf '%20s %p\n' | sort -n | cut -b22- | tr
'\n' '\000' | xargs -0 ls -laSr
```

Параллельный запуск процессов

Xargs часто используется для параллельного запуска нескольких процессов. Вот так, например, можно одновременно сжать несколько директорий в tar.gz:

```
$ echo dir1 dir2 dir3 | xargs -P 3 -I NAME tar czf NAME.tar.gz
NAME
```

В приведенном примере используется ключ -P. Он указывает максимальное количество процессов, которые будут выполняться одновременно. Предположим, что у нас на входе имеется 10 аргументов. Если мы введём команду xargs с ключом -P 3, то будет запущено 3 экземпляра команды, следующей после xargs, с каждым из этих аргументов.

С помощью xargs можно также параллельно загружать из Интернета множество файлов:

```
$ wget -nv <ссылка> | egrep -o "http://[^[:space:]]*.jpg" |
xargs -P 10 -n 1 wget -nv
```

В приведенном примере с указанного адреса будут скачаны все графические файлы с расширением jpg; ключ -P указывает, что требуется скачивать по 10 файлов одновременно.

Предварительные итоги

Подведём предварительные итоги и сформулируем несколько правил работы с xargs.

1. Xargs не работает с файлами, в имени которых присутствует пробел. Для решения этой проблемы с командой xargs используется опция −0. Пробел в имени файла можно обойти также следующим образом:

2.
```
$ xargs -I FILE my_command “FILE”
```

3. Команда xargs принимает команды из со стандартного ввода, разделенные пробелом или переводом строки. Чтобы группировать эти команды, можно использовать двойные или одинарные кавычки. Можно также указать разделитель с помощью опции -d;

4. Если команде xargs не передать вообще никаких аргументов, то по умолчанию будет выполнена команда /bin/echo;
5. Во многих случаях команду xargs можно заменить циклом for. Например, команда

6.
```
$ find . -type f -and -iname "*.deb" | xargs -n 1 dpkg -I
```

полностью эквивалента циклу

```
$ for file in `find . -type f -and -iname "*.deb"`; do dpkg -
I "$file"; done
```

Нетривиальные примеры

Основы мы вспомнили, типичные варианты использования рассмотрели… Перейдем теперь к более сложным и нетривиальным примерам. До некоторых из них мы додумались самостоятельно, работая над повседневными задачами, а некоторые — почерпнули с сайта http://www.commandlinefu.com (всем желающим научиться тонкостям работы с командной строкой очень рекомендуем время от времени его посещать — там порой можно найти очень полезные советы).

Баним IP-адреса из списка

Чтобы забанить IP-адреса из списка, нужно их добавить в IP tables с правилом DROP. Эта операция осуществляется при помощи команды:

```
$ cat bad_ip_list | xargs -I IP iptables -A INPUT -s IP -j DROP
```

Можно проделать и более сложную операцию и забанить все адреса по AS:

```
$ /usr/bin/whois -H -h whois.ripe.net -T route -i origin
AS<номер>|egrep "^route"|awk '{print $2}' |xargs -I NET iptables
-A INPUT -s NET -j DROP
```

Изменяем формат URL

Преобразовать URL вида «http%3A%2F%2Fwww.google.com» в «www,google.com» можно при помощи команды:

```
echo "http%3A%2F%2Fwww.google.com" | sed -e's/%\([0-9A-F][0-9A-F]\)/\\\\\x\1/g' | xargs echo -e
```

Генерируем пароль из 10 символов

Сгенерировать надежный пароль можно при помощи команды вида:

```
$ tr -dc A-Za-z0-9_ < /dev/urandom | head -c 10 | xargs
```

Генерировать пароли можно и без помощи xargs: для этого существует специализированная утилита pwgen. Некоторые другие способы генерации паролей описаны также здесь.

Ищем бинарные файлы, установленные без использования dpkg

Такая операция может потребоваться в случае, если, например, машина стала жертвой хакерской атаки и на ней было установлено вредоносное программное обеспечение. Выявить, что за программы поставили злоумышленники, поможет следующая команда (она ищет запущенные «бинарники», установленные без использования менеджера пакетов dpkg):

```
$ cat /var/lib/dpkg/info/*.list > /tmp/listin ; ls /proc/*/exe |xargs -l readlink | grep -xvFf /tmp/listin; rm /tmp/listin
```

Удаляем устаревшие пакеты ядра

```
$ dpkg -l linux-* | awk '/^ii/{ print $2}' | grep -v -e `uname -r | cut -f1,2 -d"-"` | grep -e [0-9] | xargs sudo apt-get -y purge
```

Проблема удаления старых ядер уже обсуждалась на Хабре — см. здесь (по этой же ссылке можно найти любопытные примеры команд).

Преобразуем скрипт в строку

Иногда возникает необходимость преобразовать большой скрипт в одну строку. Сделать это можно так:

```
$ (sed 's/#.*//g'|sed '/^ *$/d'|tr '\n' ';'|xargs echo) < script.sh
```

Заключение

Как видно из проделанного обзора, возможности xargs гораздо шире, чем это может показаться на первый взгляд. Надеемся, что приведённые в статье примеры окажутся для вас полезными. Если вам известны другие интересные варианты использования xargs — добро пожаловать в комментарии. Читателей, которые по тем или иным причинам не могут оставлять комментарии здесь, приглашаем в наш блог.

Теги:

- xargs
- linux
- системное администрирование
- селектел
- selectel

- **+62**
  - **478**
  - **81,1k**
  - **44**