

# Глава 5. Системное администрирование: управление пользователями

Никакая сложная операционная система, будь то Windows, OS X или Unix, не может функционировать бесконечно долго без вмешательства человека. Если вы работаете в многопользовательской системе Linux, значит, кто-то выполняет задачи системного администрирования. Вы можете игнорировать пресловутого «человека за ширмой», управляющего всем и вся, или сами быть великим и могучим волшебником из страны Оз — тем, кто двигает рычаги и нажимает кнопки, чтобы обеспечить нормальную работу системы. Если вы единственный пользователь системы, вам придется регулярно решать задачи системного администрирования самостоятельно.

К счастью, сценарии командной оболочки не в последнюю очередь существуют для того, чтобы упростить жизнь администраторам систем Linux (о чем и пойдет речь в этой главе). Довольно многие команды Linux в действительности являются сценариями, и многие из самых основных задач, такие как добавление пользователей, анализ использования дискового пространства и управление файлами гостевой учетной записи, можно достаточно эффективно решать с помощью коротких сценариев.

Что интересно, многие сценарии, предназначенные для системного администрирования, включают не более 20–30 строк. С помощью команд Linux можно выявить, какие команды являются сценариями, а добавив конвейер — узнать, сколько строк содержит каждый из них. Ниже перечисляется 15 самых коротких сценариев в `/usr/bin/`:

```
$ file /usr/bin/* | grep "shell script" | cut -d: -f1 | xargs wc -l \  
| sort -n | head -15  
 3 zcmp  
 3 zegrep  
 3 zfgrep  
 4 mkfontdir  
 5 pydoc  
 7 sgm1which  
 8 batch  
 8 ps2pdf12  
 8 ps2pdf13
```

```
8 ps2pdf14
8 timed-read
9 timed-run
10 c89
10 c99
10 neqn
```

Ни один из 15 самых коротких сценариев в каталоге `/usr/bin/` не содержит больше 10 строк. И десятистрочный сценарий форматирования формул `neqn` наглядно демонстрирует, как короткий сценарий командной оболочки может упрощать жизнь пользователям:

```
#!/bin/bash
# Присутствие этого сценария не должно расцениваться как наличие поддержки
# GNU eqn и groff -Tascii|-Tlatin1|-Tutf8|-Tcpl047.

: ${GROFF_BIN_PATH=/usr/bin}
PATH=${GROFF_BIN_PATH}:$PATH
export PATH
exec eqn -Tascii ${1+"$@"}

# eof
```

Сценарии, которые будут представлены в этой главе, такие же короткие и полезные, как `neqn`, и помогают решить множество административных задач, включая резервное копирование системы, добавление и удаление учетных записей и пользовательских данных, управление учетными записями. Также вы получите простой и удобный интерфейс к команде `date`, изменяющий текущие дату и время, и инструмент для проверки файлов `crontab`.

## № 35. Анализ использования дискового пространства

Даже с появлением очень емких жестких дисков и постоянным уменьшением цен на них системным администраторам постоянно приходится следить за использованием дискового пространства, чтобы общедоступные диски не переполнились.

Наиболее типичным приемом мониторинга является исследование каталога `/usr` или `/home` с использованием команды `du`, чтобы определить объем всех подкаталогов, с последующим выводом списка 5 или 10 пользователей, занявших больше всего дискового пространства. Однако этот подход не позволяет контролировать потребление дискового пространства в других местах на жестких дисках. Если у отдельных пользователей есть дополнительное

архивное пространство на втором диске или у вас завелись хитрецы, которые хранят огромные видеофайлы в каталоге с именем, начинающимся с точки и находящемся в каталоге */tmp* или в неиспользуемом каталоге в области *ftp*, такие факты расходования дискового пространства не будут обнаружены. Кроме того, если домашние каталоги пользователей разбросаны по нескольким дискам, поиск каждого каталога */home* может оказаться неоптимальным.

Лучшее решение — получить имена всех учетных записей непосредственно из файла */etc/passwd* и затем отыскать в файловой системе все файлы, принадлежащие каждой учетной записи, как показано в листинге 5.1.

## Код

### Листинг 5.1. Сценарий `fquota`

```
#!/bin/bash
# fquota -- инструмент анализа расходования дискового пространства для Unix;
# предполагается, что все учетные записи рядовых пользователей
# имеют числовые идентификаторы UID >= 100

MAXDISKUSAGE=20000 # В мегабайтах

for name in $(cut -d: -f1,3 /etc/passwd | awk -F: '$2 > 99 {print $1}')
do
    /bin/echo -n "User $name exceeds disk quota. Disk usage is: "
    # Вам может потребоваться изменить следующий список каталогов, чтобы
    # он лучше соответствовал структуре каталогов на вашем диске.
    # Наиболее вероятно, что вам придется заменить имя /Users на /home.
    ❶ find /usr /var /Users -xdev -user $name -type f -ls | \
        awk '{ sum += $7 } END { print sum / (1024*1024) " Mbytes" }'
    ❷ done | awk "\$9 > $MAXDISKUSAGE { print \$0 }"

exit 0
```

## Как это работает

В соответствии с соглашениями, идентификаторы пользователей (User ID, UID) от 1 до 99 отводятся для системных демонов и административных задач, а идентификаторы со значениями 100 и выше можно выбирать для учетных записей обычных пользователей. Поскольку администраторы Linux обычно весьма организованные люди, этот сценарий пропускает все учетные записи со значениями UID меньше 100.

Аргумент `-xdev` в вызове команды `find` ❶ гарантирует, что `find` не будет выполнять поиск во всех файловых системах. Иными словами, этот аргумент

предотвращает обход командой системных областей, каталогов, доступных только для чтения, извлекаемых устройств, каталога */proc* действующих процессов (в Linux) и других подобных областей. Вот почему в список явно включены такие каталоги, как */usr*, */var* и */home*. Эти каталоги часто размещаются в отдельных файловых системах для упрощения их резервного копирования и организации. Добавление их в список, когда они действительно находятся в корневой файловой системе, не означает, что они будут просмотрены дважды.

На первый взгляд кажется, что сценарий выведет сообщение `exceeds disk quota` (превысил дисковую квоту) для любой учетной записи, но это не так: команда `awk`, следующая за концом цикла **2**, позволит вывести такое сообщение только для учетных записей, файлы которых занимают больше чем `MAXDISKUSAGE`.

## Запуск сценария

Сценарий не имеет аргументов и должен запускаться с привилегиями `root`, чтобы гарантировать доступность всех каталогов и файловых систем. Запускать сценарии с такими привилегиями предпочтительнее с помощью команды `sudo` (выполните команду `man sudo` в окне терминала, чтобы получить дополнительную информацию). Почему именно с помощью `sudo`? Потому что такой прием позволяет выполнить с привилегиями `root` только одну команду, после чего привилегии командной оболочки будут вновь понижены до уровня обычного пользователя. Каждый раз, когда вам потребуется выполнить административную команду, используйте для этого `sudo`. Использование команды `su - root`, напротив, позволит выполнить все последующие команды с привилегиями `root`, пока подоболочка не будет закрыта явно, а отвлекшись на что-то срочное, легко забыть, что вы получили привилегии `root`, и есть риск по ошибке сделать что-то, что приведет к разрушительным последствиям.

### ПРИМЕЧАНИЕ

---

Измените список каталогов в команде `find` **1**, чтобы он точно соответствовал структуре каталогов на вашем диске.

---

## Результаты

Сценарий выполняет поиск по целым файловым системам, поэтому не надо удивляться, что ему для работы требуется немало времени. В больших файловых системах процесс легко может занять промежуток между утренней чашкой чая и обедом. В листинге 5.2 приводится пример результатов работы сценария.