

# Занятие 9. Разное

## Занятие 9. Разное

Монтирование и размонтирование устройств хранения  
Определение имени устройства  
Контроль использования места на носителе информации  
Архивирование данных  
Процессы  
ssh  
Литература

## Монтирование и размонтирование устройств хранения

В современных user-friendly версиях Linux обычно достаточно подключить запоминающее устройство, например, флешку, к компьютеру, и оно уже готово к работе. Но если используемый дистрибутив не обеспечивает автоматическое подключение, эту операцию приходится выполнять вручную. Подключение запоминающего устройства к дереву файловой системы мы будем называть *монтированием*.

Для монтирования мы будем использовать команду `mount`. Будучи вызванной без аргументов, `mount` отображает список устройств, смонтированных в настоящее время:

```
1 user@linux-pc:~$ mount
2 sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
3 proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
4 udev on /dev type devtmpfs
  (rw,nosuid,noexec,relatime,size=4027732k,nr_inodes=1006933,mode=755)
5 devpts on /dev/pts type devpts
  (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
6 tmpfs on /run type tmpfs (rw,nosuid,nodev,noexec,relatime,size=815276k,mode=755)
7 /dev/sda1 on / type ext4 (rw,relatime)
8 securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
9 tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
```

Команда `mount` предоставляет следующую информацию в каждой строке:

1. Имя файла устройства с носителем информации.
2. Точка монтирования.
3. Тип файловой системы.
4. Параметры.

Для монтирования устройства необходимо обладать правами суперпользователя. Обычно для монтирования используется команда монтирования со следующим набором параметров и аргументов:

```
1 | mount -t тип_файловой_системы имя_устройства каталог
```

Linux поддерживает большое количество файловых систем. Если устройства хранения информации применяется для работы и в ОС Windows, и в ОС Linux, то, скорее всего, оно использует одну из следующих файловых систем:

vfat (файловая система Windows с длинными именами файлов);

ntfs (файловая система Windows);

iso9660 (стандартная файловая система компакт-дисков).

Флешки и дискеты чаще всего бывают отформатированными под vfat. Если требуется смонтировать CD с данными, то необходимо будет использовать файловую систему iso9660.

Иногда бывает необходимо установить драйвер, например, для возможности использовать устройства с ФС exfat. Запоминающие устройства с проприетарными ФС, например, жёсткие диски или соседние разделы с ОС Windows, рекомендуется монтировать только на чтение.

Чтобы вручную смонтировать флешку (устройство с именем /dev/sdb1) в каталог /media/disk, необходимо воспользоваться следующей командой:

```
1 | mount -t vfat /dev/sdb1 /media/disk
```

После того, как устройство будет смонтировано, пользователь root получает полный доступ к этому устройству, однако доступ других может быть ограничен. Чтобы избежать таких проблем, нужно проверить права у каталога, в который выполняется монтирование.

После окончания работы с устройством его необходимо размонтировать. Для размонтирования устройств используется команда umount. Команда umount предоставляет возможность указывать либо имя смонтированного каталога, либо имя смонтированного устройства:

```
1 | umount [каталог | имя_устройства]
```

Если на устройстве хотя бы один файл открыт какой-либо программой, то система не позволит размонтировать устройство. Однако, если с файлом работает программа обычного пользователя, а команда размонтировать устройство исходить от суперпользователя, то устройство может быть размонтировано с потерей данных.

## Определение имени устройства

Иногда сложно определить имя устройства.

Для запоминающих устройств, которые даже не были монтированы, можно вызвать команду

```
1 | fdisk -l
```

По объёму носителя можно попытаться определить, под каким именно именем система видит флешку.

Для всех остальных устройств можно использовать следующую уловку: запустите мониторинг содержимого файла `/var/log/messages` (RHEL) или `/var/log/syslog` (Debian/Ubuntu-like) в режиме реального времени (для чего понадобятся права суперпользователя):

```
1 root@linux-pc:# tail -f /var/log/syslog
2 Nov  4 15:08:45 linux-pc systemd[1]: run-user-119.mount: Succeeded.
3 Nov  4 15:08:45 linux-pc systemd[1]: user-runtime-dir@119.service: Succeeded.
4 Nov  4 15:08:45 linux-pc systemd[1]: Stopped User Runtime Directory /run/user/119.
5 ...
```

Эта команда выведет последние строки из файла и приостановится.

Подключите устройство. Ядро сразу же его обнаружит и проверит:

```
1 root@linux-pc:# tail -f /var/log/syslog
2 Nov  4 15:08:45 linux-pc systemd[1]: run-user-119.mount: Succeeded.
3 Nov  4 15:08:45 linux-pc systemd[1]: user-runtime-dir@119.service: Succeeded.
4 Nov  4 15:08:45 linux-pc systemd[1]: Stopped User Runtime Directory /run/user/119.
5 Nov  4 15:17:18 linux kernel: [ 589.576282] usb 1-1: new high-speed USB device
number 2 using ehci-pci
6 Nov  4 15:17:19 linux-pc kernel: [ 589.941430] usb 1-1: New USB device found,
idVendor=13fe, idProduct=5500, bcdDevice= 1.10
7 Nov  4 15:17:19 linux-pc kernel: [ 589.941435] usb 1-1: New USB device strings:
Mfr=1, Product=2, SerialNumber=0
8 Nov  4 15:17:19 linux-pc kernel: [ 589.941437] usb 1-1: Product: 2307 Boot ROM
9 Nov  4 15:17:19 linux-pc kernel: [ 589.941439] usb 1-1: Manufacturer: Phison
10 Nov  4 15:17:19 linux-pc mtp-probe: checking bus 1, device 2:
"/sys/devices/pci0000:00/0000:00:0b.0/
11 Nov  4 15:17:20 linux-pc kernel: [ 591.112234] sd 3:0:0:0: Attached scsi generic
sg2 type 0
12 Nov  4 15:17:20 linux-pc kernel: [ 591.162182] sd 3:0:0:0: [sdb] Attached SCSI
removable disk
```

Когда вывод опять приостановится, нажмите CTRL+C, чтобы вернуться в приглашение командной строки. Наибольший интерес для нас представляют строки с упоминанием имени устройства - в данном случае это `/dev/sdb`. Следующие две строки являются особенно показательными

```
1 Nov  4 15:17:20 linux-pc kernel: [ 591.112234] sd 3:0:0:0: Attached scsi generic sg2
type 0
2 Nov  4 15:17:20 linux-pc kernel: [ 591.162182] sd 3:0:0:0: [sdb] Attached SCSI
removable disk
```

Зная имя устройства, можно смонтировать флешку, но нужно помнить, что обычно устройства носят имена формата `\dev\sdX`, а первый раздел на устройстве `\dev\sdX1`, например в нашем случае это `\dev\sdb1`.

## Контроль использования места на носителе информации

Иногда возникает необходимость узнать, каков объем свободного пространства на каждом отдельном устройстве. С помощью команды `df` можно легко определить характеристики всех смонтированных дисков:

```

1 user@linux-pc:~$ df
2 Файл.система 1K-блоков Использовано Доступно Использовано% Смонтировано в
3 udev          4027732          0 4027732          0% /dev
4 tmpfs         815276          1084 814192          1% /run
5 /dev/sda1     41016304       14120340 24782748        37% /
6 tmpfs         4076368          0 4076368          0% /dev/shm
7 tmpfs         5120             4    5116            1% /run/lock
8 tmpfs         4076368          0 4076368          0% /sys/fs/cgroup
9 tmpfs         815272           8   815264          1% /run/user/1001
10 /dev/sdb1     60550144        66560 60483584        1% /media/user/INTERNAL

```

В команде `df` предусмотрено большое количество параметров командной строки, большая часть которых обычно не используется. Одним из широко применяемых параметров является параметр `-h`, который показывает место на диске в формате, удобном для восприятия человеком:

```

1 user@linux-pc:~$ df -h
2 Файл.система  Размер Использовано  Дост  Использовано% Смонтировано в
3 udev          3,9G          0 3,9G          0% /dev
4 tmpfs         797M          1,1M 796M          1% /run
5 /dev/sda1     40G           14G 24G           37% /
6 tmpfs         3,9G          0 3,9G          0% /dev/shm
7 tmpfs         5,0M          4,0K 5,0M          1% /run/lock
8 tmpfs         3,9G          0 3,9G          0% /sys/fs/cgroup
9 tmpfs         797M          8,0K 797M          1% /run/user/1001
10 /dev/sdb1     58G           65M 58G           1% /media/user/INTERNAL
11

```

Команда `du` показывает использование дискового пространства применительно к конкретному каталогу. С помощью этой команды можно быстро определить, нет ли таких файлов, которые бесполезно расходуют дисковое пространство в системе

```

1 user@linux-pc:~$ du
2 4      ./Музыка
3 4      ../gnupg/private-keys-v1.d
4 8      ../gnupg
5 8      ../cache/mesa_shader_cache/f5
6 8      ../cache/mesa_shader_cache/da
7 8      ../cache/mesa_shader_cache/9b
8 8      ../cache/mesa_shader_cache/5f
9 8      ../cache/mesa_shader_cache/f6
10 ...

```

Команду `du` обычно используют со следующими параметрами

- `-c` (сформировать общий итог по всем перечисленным файлам);
- `-h` (выводить значения размеров в форме, удобной для восприятия человеком);
- `-s` (суммировать данные по каждому параметру).

```

1 user@linux-pc:~$ du -h
2 8,0K  ./linux-pc-maintenance/.git/objects/c0
3 8,0K  ./linux-pc-maintenance/.git/objects/90
4 8,0K  ./linux-pc-maintenance/.git/objects/3c
5 8,0K  ./linux-pc-maintenance/.git/objects/cf

```

```

6 8,0K ./linux-pc-maintenance/.git/objects/18
7 884K ./linux-pc-maintenance/.git/objects
8 1,1M ./linux-pc-maintenance/.git
9 1,1M ./linux-pc-maintenance
10 4,0K ./Музыка
11 4,0K ./gnupg/private-keys-v1.d
12 8,0K ./gnupg
13 4,0K ./cache/mesa_shader_cache
14 176K ./cache/pcmanfm-qt/lxqt
15 180K ./cache/pcmanfm-qt
16 4,0K ./cache/thumbnails/normal
17 8,0K ./cache/thumbnails
18 4,0K ./cache/openbox/sessions
19 8,0K ./cache/openbox
20 11M ./cache
21 4,0K ./local/share
22 8,0K ./local
23 4,0K ./Загрузки
24 4,0K ./Видео
25 4,0K ./Изображения
26 12K ./dbus/session-bus
27 16K ./dbus
28 4,0K ./Общедоступные
29 8,0K ./config/pcmanfm-qt/lxqt
30 12K ./config/pcmanfm-qt
31 8,0K ./config/Clipper
32 28K ./config/openbox
33 8,0K ./config/featherpad
34 48K ./config/lxqt
35 120K ./config
36 4,0K ./Desktop
37 4,0K ./Шаблоны
38 4,0K ./Документы
39 12M .
40 ...

```

## Архивирование данных

Под *архивированием* или *архивацией* обычно понимают процесс агрегации множества файлов в один большой файл. Если Вы до этого работали в системах семейства Windows, то наверняка упомянете сжатие этого большого файла, но на самом деле сжатие обязательным не является. Архивирование широко применяется как один из этапов создания резервных копий системы. Оно также используется при перемещении старых данных из системы в некоторое долговременное хранилище.

В качестве инструмента архивирования в Linux широко применяется команда `tar`. Команда `tar` (type archive) первоначально использовалась для записи файлов на ленточное устройство в целях архивирования. Но эта команда позволяет также записывать вывод в файл, и этот способ стал широко применяться для архивирования данных в Linux.

Формат команды `tar` приведен ниже:

```
1 | tar режим [параметр...] путь...
```

Режимы команды tar

Режим	Описание
c	Создать архив из списка каталогов/файлов
x	Извлечь файла из архива
r	Добавить каталог и/или файл в архив
t	Вывести список содержимого архива

Режим указывается вместе с параметрами.

Параметры команды tar

Параметр	Описание
C каталог	Перейти в указанный каталог
f файл	Вывести результаты в файл
j	Перенаправить вывод в команду bzip2 для сжатия
p	Сохранить все права доступа к файлу
v	Выводить имена файлов в процессе обработки
z	Перенаправить вывод в команду gzip для сжатия

Рассмотрим типичные сценарии использования команды tar. Прежде всего можно создать файл архива с использованием следующей команды:

```
1 | tar -cvf tests.tar test_1/ test_2/
```

Команда, приведенная выше, создает файл архива с именем tests.tar, который включает содержимое двух каталогов test\_1 и test\_2.

Команда

```
1 | tar -tf tests.tar
```

позволяет просмотреть содержимое архива.

Наконец, команда

```
1 | tar -xvf tests.tar
```

извлекает содержимое tar-файла. Обратите внимание, что если tar-файл был создан исходя из определенной структуры каталогов, то воссоздается вся эта структура каталогов, начиная с текущего каталога.

А теперь заархивируем со сжатием через утилиту gzip

```
1 | tar -czvf tests.tar.gz test_1/ test_2/
```

что эквивалентно вызову двух команд

```
1 | tar -cvf tests.tar test_1/ test_2/
2 | gzip tests.tar >tests.tar.gz
```

обратите внимание, что будет удалён оригинальный файл tests.tar. Чтобы создать копию, вызываем с ключом -c:

```
1 | tar -cvf tests.tar test_1/ test_2/
2 | gzip tests.tar -c >tests.tar.gz
```

## Процессы

Программу, исполняемую в системе, принято называть процессом. Мы не будем делать уточнений или давать точных определений, чтобы не залезать в дебри спецдисциплин старших курсов, нам достаточно на данном этапе знать, что каждой программе соответствует процесс.

Чаще всего для просмотра списка процессов используется команда ps. Команда ps имеет множество параметров, но в самом простейшем случае она используется следующим образом:

```
1 | user@linux-pc:~$ ps
2 |    PID TTY          TIME CMD
3 |     9 tty1        00:00:00 bash
4 |    71 tty1        00:00:00 ps
5 | user@linux-pc:~$
```

По умолчанию команда ps показывает только процессы, которые принадлежат текущему пользователю и выполняются на текущем терминале. В данном случае приведены сведения только о командной оболочке bash, а также о самой команде ps. Выходные данные команды в основной форме показывают идентификатор процесса (process ID — PID), терминал (TTY), на котором он выполняется, и процессорное время, использованное процессом.

aux — одна из популярных комбинаций параметров команды ps. Эта комбинация параметров выводит процессы, принадлежащие всем пользователям:

```
1 | user@linux-pc:~$ ps -aux
2 | USER  PID TTY          TIME CMD
3 | root    772 0.0  0.1 136752 14100 ?        Ssl  12:51   0:00 /usr/bin/sddm
4 | root    780 0.0  0.0 295432 2944 ?        S1   12:51   0:00
   /usr/sbin/VBoxService --pidfile /var/run/vboxadd-service.sh
5 | root    792 1.7  0.9 851272 81368 tty1     Ssl+ 12:51   0:01
   /usr/lib/xorg/Xorg -nolisten tcp -auth /var/run/sddm/{242038b2-39c1-4f2a-90b2-
   4ee4cde7d55f} -background none -noreset -dis
6 | rtkit   820 0.0  0.0 152940 3068 ?        SNsl 12:51   0:00
   /usr/libexec/rtkit-daemon
7 | root    836 0.0  0.0  8268 4576 ?        Ss   12:51   0:00
   /usr/lib/bluetooth/bluetoothd
```

```

8 root      843  0.0  0.1  59620 14480 ?          S   12:52  0:00 /usr/lib/x86_64-
linux-gnu/sddm/sddm-helper --socket /tmp/sddm-autha37e3d01-4bc0-4ea5-a421-
4afc4e27644d --id 1 --start env
9 user      845  0.4  0.1  18400  9612 ?          Ss  12:52  0:00 /lib/systemd/systemd
--user
10 user      846  0.0  0.0 103080  3252 ?          S   12:52  0:00 (sd-pam)
11 user      851  0.4  0.2 622444 19052 ?          S<s1 12:52  0:00 /usr/bin/pulseaudio
--daemonize=no --log-target=journal
12 user      853  0.3  0.0   7552  4768 ?          Ss  12:52  0:00 /usr/bin/dbus-daemon
--session --address=systemd: --nofork --nopidfile --systemd-activation --syslog-
only
13 user      857  0.0  0.0 160624  4760 ?          Sl  12:52  0:00 /usr/bin/gnome-
keyring-daemon --daemonize --login
14 user      860  1.2  0.4 265160 39320 ?          Sl  12:52  0:00 lxqt-session
15 user      906  0.0  0.0  22620   364 ?          S   12:52  0:00 /usr/bin/VBoxClient
--clipboard
16 user      907  0.0  0.0 154756  4188 ?          Sl  12:52  0:00 /usr/bin/VBoxClient
--clipboard
17 user      918  0.0  0.0  22620   364 ?          S   12:52  0:00 /usr/bin/VBoxClient
--seamless
18 user      920  0.0  0.0 154856  2756 ?          Sl  12:52  0:00 /usr/bin/VBoxClient
--seamless
19 user      926  0.0  0.0  22620   360 ?          S   12:52  0:00 /usr/bin/VBoxClient
--draganddrop
20 user      927  0.1  0.0 155372  2680 ?          Sl  12:52  0:00 /usr/bin/VBoxClient
--draganddrop
21 user      933  0.0  0.0  22620   360 ?          S   12:52  0:00 /usr/bin/VBoxClient
--vmsvga
22 user      934  0.0  0.0 157136  3500 ?          Sl  12:52  0:00 /usr/bin/VBoxClient
--vmsvga
23 user      938  0.0  0.0   6040   460 ?          Ss  12:52  0:00 /usr/bin/ssh-agent
/usr/bin/im-launch env LXQT_DEFAULT_OPENBOX_CONFIG=/etc/xdg/xdg-
Lubuntu/openbox/lxqt-rc.xml /usr/bin/st
24 user      956  0.9  0.3 231920 29524 ?          Sl  12:52  0:00 /usr/bin/openbox --
config-file /home/user/.config/openbox/lxqt-rc.xml
25 user      959  0.0  0.0 305256  6420 ?          Sl  12:52  0:00 /usr/libexec/at-spi-
bus-launcher --launch-immediately
26 user      961  0.0  0.0 236936  4744 ?          Sl  12:52  0:00
/usr/libexec/geoclue-2.0/demos/agent
27 user      966  0.0  0.0   7248  3912 ?          S   12:52  0:00 /usr/bin/dbus-daemon
--config-file=/usr/share/defaults/at-spi2/accessibility.conf --nofork --print-
address 3
28 user      972  1.7  1.1 1357868 93972 ?          Sl  12:52  0:00 /usr/bin/pcmanfm-qt
--desktop --profile=lxqt
29 user      973  0.7  0.4 338344 37512 ?          Sl  12:52  0:00 /usr/bin/lxqt-
globalkeysd
30 user      974  1.0  0.9 982104 76872 ?          Sl  12:52  0:00 /usr/bin/lxqt-
notificationd
31 user      975  3.8  1.2 1339876 98188 ?          Sl  12:52  0:01 /usr/bin/lxqt-panel
32 user      976  0.3  0.4 412504 38396 ?          Sl  12:52  0:00 /usr/bin/lxqt-
policykit-agent
33 user      977  1.6  0.6 284552 50124 ?          Sl  12:52  0:00 /usr/bin/lxqt-
runner
34 user      979  0.2  0.0  15940   5044 ?          S   12:52  0:00
/usr/bin/xscreensaver -no-splash
35 user      982  1.2  0.4  59692 34720 ?          S   12:52  0:00 /usr/bin/python3
/usr/share/system-config-printer/applet.py

```



```

36 user      998  0.0  0.0  2616  540 ?          S   12:52  0:00 /bin/sh
   /usr/lib/lubuntu-update-notifier/lubuntu-upg-notifier.sh
37 user     1043  0.0  0.0 239720  7700 ?          Ss1 12:52  0:00 /usr/libexec/gvfsd
38 user     1050  0.0  0.0 312808  6360 ?          S1  12:52  0:00 /usr/libexec/gvfsd-
fuse /run/user/1001/gvfs -f -o big_writes
39 user     1051  0.0  0.0 313896  7856 ?          S1  12:52  0:00 /usr/libexec/gvfsd-
trash --spawner :1.14 /org/gtk/gvfs/exec_spaw/0
40 user     1064  0.0  0.1 313872  8740 ?          Ss1 12:52  0:00 /usr/libexec/gvfs-
udisks2-volume-monitor
41 user     1069  0.0  0.0 235708  6396 ?          Ss1 12:52  0:00 /usr/libexec/gvfs-
mtp-volume-monitor
42 ...
43 user@linux-pc:~$

```

Описание некоторых заголовков столбцов приведено в таблице ниже.

Заголовок	Значение
USER	Идентификатор пользователя (владельца процесса)
%CPU	Использование процессора в процентах
%MEM	Использование памяти в процентах
START	Время запуска процесса. Для значений свыше 24 часов выводится дата

Программа ps предоставляет массу информации о том, что происходит в системе, но она делает лишь мгновенный снимок. Чтобы увидеть работу компьютера в динамике воспользуемся командой top.

```

1 user@linux-pc:~$ top
2 root      772  0.0  0.1 136752 14100 ?          Ss1 12:51  0:00 /usr/bin/sddm
3 root      780  0.0  0.0 295432  2944 ?          S1  12:51  0:00
   /usr/sbin/VBoxService --pidfile /var/run/vboxadd-service.sh
4 root      792  1.7  0.9 851272 81368 tty1       Ss1+ 12:51  0:01
   /usr/lib/xorg/Xorg -nolisten tcp -auth /var/run/sddm/{242038b2-39c1-4f2a-90b2-
4ee4cde7d55f} -background none -noreset -dis
5 rtkit     820  0.0  0.0 152940  3068 ?          SNs1 12:51  0:00
   /usr/libexec/rtkit-daemon
6 root      836  0.0  0.0  8268  4576 ?          Ss   12:51  0:00
   /usr/lib/bluetooth/bluetoothd
7 root      843  0.0  0.1 59620 14480 ?          S   12:52  0:00 /usr/lib/x86_64-
linux-gnu/sddm/sddm-helper --socket /tmp/sddm-autha37e3d01-4bc0-4ea5-a421-
4afc4e27644d --id 1 --start env
8 user      845  0.4  0.1 18400  9612 ?          Ss   12:52  0:00 /lib/systemd/systemd
--user
9 user      846  0.0  0.0 103080  3252 ?          S   12:52  0:00 (sd-pam)
10 user      851  0.4  0.2 622444 19052 ?         S<s1 12:52  0:00 /usr/bin/pulseaudio
--daemonize=no --log-target=journal
11 user      853  0.3  0.0  7552  4768 ?          Ss   12:52  0:00 /usr/bin/dbus-daemon
--session --address=systemd: --nofork --nopidfile --systemd-activation --syslog-
only
12 user      857  0.0  0.0 160624  4760 ?          S1  12:52  0:00 /usr/bin/gnome-
keyring-daemon --daemonize --login
13 user      860  1.2  0.4 265160 39320 ?          S1  12:52  0:00 lxqt-session

```

```

14 user      906  0.0  0.0  22620  364 ?      S   12:52  0:00 /usr/bin/VBoxClient
    --clipboard
15 user      907  0.0  0.0 154756  4188 ?     S1  12:52  0:00 /usr/bin/VBoxClient
    --clipboard
16 user      918  0.0  0.0  22620  364 ?      S   12:52  0:00 /usr/bin/VBoxClient
    --seamless
17 user      920  0.0  0.0 154856  2756 ?     S1  12:52  0:00 /usr/bin/VBoxClient
    --seamless
18 user      926  0.0  0.0  22620  360 ?      S   12:52  0:00 /usr/bin/VBoxClient
    --draganddrop
19 user      927  0.1  0.0 155372  2680 ?     S1  12:52  0:00 /usr/bin/VBoxClient
    --draganddrop
20 user      933  0.0  0.0  22620  360 ?      S   12:52  0:00 /usr/bin/VBoxClient
    --vmsvga
21 user      934  0.0  0.0 157136  3500 ?     S1  12:52  0:00 /usr/bin/VBoxClient
    --vmsvga
22 user      938  0.0  0.0   6040   460 ?     Ss  12:52  0:00 /usr/bin/ssh-agent
    /usr/bin/im-launch env LXQT_DEFAULT_OPENBOX_CONFIG=/etc/xdg/xdg-
    Lubuntu/openbox/lxqt-rc.xml /usr/bin/st
23 user      956  0.9  0.3 231920 29524 ?     S1  12:52  0:00 /usr/bin/openbox --
    config-file /home/user/.config/openbox/lxqt-rc.xml
24 user      959  0.0  0.0 305256  6420 ?     S1  12:52  0:00 /usr/libexec/at-spi-
    bus-launcher --launch-immediately
25 user      961  0.0  0.0 236936  4744 ?     S1  12:52  0:00
    /usr/libexec/geoclue-2.0/demos/agent
26 user      966  0.0  0.0   7248  3912 ?     S   12:52  0:00 /usr/bin/dbus-daemon
    --config-file=/usr/share/defaults/at-spi2/accessibility.conf --nofork --print-
    address 3
27 user      972  1.7  1.1 1357868 93972 ?     S1  12:52  0:00 /usr/bin/pcmanfm-qt
    --desktop --profile=lxqt
28 user      973  0.7  0.4 338344  37512 ?     S1  12:52  0:00 /usr/bin/lxqt-
    globalkeysd
29 user      974  1.0  0.9 982104  76872 ?     S1  12:52  0:00 /usr/bin/lxqt-
    notificationd
30 user      975  3.8  1.2 1339876 98188 ?     S1  12:52  0:01 /usr/bin/lxqt-panel
31 user      976  0.3  0.4 412504  38396 ?     S1  12:52  0:00 /usr/bin/lxqt-
    policykit-agent
32 user      977  1.6  0.6 284552  50124 ?     S1  12:52  0:00 /usr/bin/lxqt-
    runner
33 user      979  0.2  0.0  15940  5044 ?     S   12:52  0:00
    /usr/bin/xscreensaver -no-splash
34 user      982  1.2  0.4  59692 34720 ?     S   12:52  0:00 /usr/bin/python3
    /usr/share/system-config-printer/applet.py
35 user      998  0.0  0.0   2616   540 ?     S   12:52  0:00 /bin/sh
    /usr/lib/lubuntu-update-notifier/lubuntu-upg-notifier.sh
36 user     1043  0.0  0.0 239720  7700 ?     Ss1 12:52  0:00 /usr/libexec/gvfsd
37 user     1050  0.0  0.0 312808  6360 ?     S1  12:52  0:00 /usr/libexec/gvfsd-
    fuse /run/user/1001/gvfs -f -o big_writes
38 user     1051  0.0  0.0 313896  7856 ?     S1  12:52  0:00 /usr/libexec/gvfsd-
    trash --spawner :1.14 /org/gtk/gvfs/exec_spaw/0
39 user     1064  0.0  0.1 313872  8740 ?     Ss1 12:52  0:00 /usr/libexec/gvfs-
    udisks2-volume-monitor
40 user     1069  0.0  0.0 235708  6396 ?     Ss1 12:52  0:00 /usr/libexec/gvfs-
    mtp-volume-monitor
41

```

По умолчанию после запуска команды `top` сортировка процессов осуществляется с учетом значения `%CPU`.

Программа `top` принимает ряд команд с клавиатуры. Наибольший интерес представляет команда `h`, которая выводит экран со справочной информацией, и `q`, которая завершает `top`.

Команда `kill` используется для «убийства» (`kill`), то есть для завершения процессов. Она позволяет принудительно завершить выполнение вышедшей из-под контроля программы, отвергающей любые другие попытки закрыть ее.

Хотя все выглядит достаточно просто, в действительности команда `kill` не просто «убивает» (`kill`) процессы — она посылает им сигналы. Сигналы — один из нескольких способов, которыми операционная система общается с программами.

Наиболее типичный синтаксис команды `kill` имеет следующий вид:

```
1 | kill [-сигнал] PID...
```

Если сигнал явно не указан в команде, по умолчанию посылается сигнал 15 (`TERM`). Для прекращения выполнения программы ей обычно посылается сигнал 9 (`KILL`).

## ssh

SSH (Secure Shell) - это сетевой протокол, который обеспечивает безопасное удаленное соединение между двумя системами. Системные администраторы используют утилиты SSH для управления компьютерами, копирования или перемещения файлов между системами.

В своей работе протокол SSH опирается на два компонента. На удалённом компьютере работает сервер SSH, принимающий соединения по определённому порту (по умолчанию порт 22), а в локальной системе работает клиент SSH, осуществляющий обмен информацией с удаленным сервером. Большинство дистрибутивов Linux включают реализацию SSH с названием OpenSSH из проекта BSD.

Для подключения к удаленному компьютеру вам потребуется его IP-адрес или имя. Давайте попытаемся присоединиться к Raspberry Pi в домашней сети, зная, что IP-адрес нашего устройства 192.168.1.34, а работать мы будем под учётной записью `pi`, которая на удалённом сервере существует:

```
1 | user@linux-pc:~$ ssh pi@192.168.1.34:22
```

При первой попытке подключения на экран выводится предупреждение, сообщающее, что аутентичность удаленного компьютера не может быть установлена. Это объясняется тем, что программа-клиент прежде никогда не подключалась к данному удаленному компьютеру. Чтобы принять идентификационные данные удаленного узла, введите `yes` в ответ на приглашение. После установки соединения пользователю будет предложено ввести пароль:

```
1 | user@linux-pc:~$ ssh pi@192.168.1.34:22
2 | password:
```

После ввода действительного пароля в терминале появится приглашение командной оболочки из удаленной системы:

```
1 user@linux-pc:~$ ssh pi@192.168.1.34:22
2 password:
3 pi@192.168.1.34:~$
```

Сеанс с удаленной командной оболочкой продолжается, пока пользователь не введет команду `exit` в приглашении удаленной командной оболочки, после чего соединение закроется. В этот момент возобновится сеанс локальной командной оболочки и появится ее приглашение к вводу.

```
1 user@linux-pc:~$ ssh pi@192.168.1.34:22
2 password:
3 pi@192.168.1.34:~$ exit
4 user@linux-pc:~$
```

По SSH можно безопасно копировать файлы с помощью программы SCP.

Например, чтобы скопировать файл `file1.txt` в каталог `/home/pi/` на удаленном сервере с проверкой имени пользователя, введите:

```
1 scp file1.txt pi@192.168.1.34:22:/home/pi/
```

Помимо открытия сеанса командной оболочки в удаленной системе `ssh` позволяет также выполнить единственную команду. Чтобы удаленно выполнить команду с локального компьютера, добавьте эту команду к команде SSH. Например, чтобы удалить файл, введите:

```
1 user@linux-pc:~$ ssh pi@192.168.1.34:22 'rm /home/pi/file1.txt'
2 password:
3 user@linux-pc:~$
```

Введите пароль, и файл на удаленном сервере будет удален без создания новой оболочки.

## Литература

1. Шотс У. “Командная строка Linux.”, глава 10, глава 15 (Монтирование и размонтирование устройств хранения), глава 16 (Безопасные взаимодействия с удаленными узлами), глава 18 (Сжатие файлов, Архивирование файлов)
2. Блум Р., Бреснахэн К. “Командная строка Linux и сценарии оболочки.”, глава 4