

Командная строка Linux

2-е издание

Санкт-Петербург
«БХВ-Петербург»

2025

УДК 004.4
ББК 32.973.26-018.2
К63

К63 Командная строка Linux. — 2-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2025. — 208 с.: ил. — (Системный администратор)
ISBN 978-5-9775-2045-4

Рассмотрены задачи, которые выполняются из командной строки операционной системы Linux. Объясняется, как попасть в командную строку, работать в консоли, настраивать систему с помощью программ, обладающих только текстовым интерфейсом. Описаны особенности файловой системы Linux, наиболее полезные команды для работы с текстом, сетью и Интернетом, команды для работы с файловой системой. Рассмотрены способы перенаправления ввода-вывода, маски и псевдонимы, различные варианты запуска программ, эффективные приемы использования клавиатуры, примеры сложных команд и другие вопросы.

Во втором издании особое внимание уделено задачам системного администратора — разметке диска, восстановлению загрузчика, в том числе при использовании жестких дисков на основе GPT/UEFI, установке пакетов, мониторингу системы и автоматизации на языках оболочек `bash`.

*Для системных администраторов, программистов
и квалифицированных пользователей Linux*

УДК 004.4
ББК 32.973.26-018.2

Группа подготовки издания:

Руководитель проекта	<i>Евгений Рыбаков</i>
Зав. редакцией	<i>Людмила Гауль</i>
Редактор	<i>Григорий Добин</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Дизайн обложки	<i>Зои Канторович</i>

Подписано в печать 05.03.25.
Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 16,77.
Тираж 1000 экз. Заказ №
"БХВ-Петербург", 191036, Санкт-Петербург, Гончарная ул., 20.
Отпечатано с готового оригинал-макета
ООО "Принт-М", 142300, М.О., г. Чехов, ул. Полиграфистов, д. 1

ISBN 978-5-9775-2045-4

© ООО "БХВ", 2025
© Оформление. ООО "БХВ-Петербург", 2025

Оглавление

Глава 1. Командная строка: основы основ	9
1.1. Переключаемся из графического режима в консоль	9
1.2. Графические (псевдо)-терминалы	12
1.3. Удаленная консоль	14
1.4. Завершение работы из консоли. Команды <i>poweroff</i> , <i>halt</i> , <i>reboot</i> , <i>shutdown</i>	15
Глава 2. Работаем в консоли	17
2.1. Ввод команд. Команда <i>man</i>	17
2.2. Ввод/вывод и каналы	18
2.3. Шесть полезных команд	20
2.3.1. Команда <i>wc</i>	20
2.3.2. Команда <i>head</i>	21
2.3.3. Команда <i>cut</i>	21
2.3.4. Команда <i>grep</i>	22
2.3.5. Команда <i>sort</i>	23
2.3.6. Команда <i>uniq</i>	24
Глава 3. Оболочка bash	26
3.1. Маски имен файлов	26
3.2. Переменные окружения	27
3.3. Псевдонимы	28
3.4. Изменение приглашения командной строки	28
3.5. Еще раз о перенаправлении ввода/вывода	30
3.6. Пробелы в именах файлов	31
3.7. Инициализация переменных	32
3.8. Просмотр истории команд	32
3.9. Вызов предыдущих команд	33
3.10. Защита от случайного удаления файла	34
3.11. Использование последнего вывода команды	34
3.12. Редактирование командной строки.....	34
Глава 4. Файловая система. Команды для работы с файловой системой	36
4.1. Особенности файловой системы Linux	36
4.1.1. Имена файлов в Linux	36

4.1.2. Файлы и устройства.....	36
4.1.3. Корневая файловая система и монтирование.....	37
4.1.4. Стандартные каталоги Linux.....	38
4.2. Команды для работы с файлами и каталогами.....	39
4.2.1. Работа с файлами.....	39
4.2.2. Работа с каталогами.....	42
4.2.3. Удобная навигация по файловой системе.....	44
4.3. Команда <i>ln</i> : создание ссылок.....	47
4.4. Команды <i>chmod</i> , <i>chown</i> и <i>chattr</i>	48
4.4.1. Команда <i>chmod</i> : права доступа к файлам и каталогам.....	48
4.4.2. Команда <i>chown</i> : смена владельца файла.....	50
4.4.3. Специальные права доступа: SUID и SGID.....	50
4.4.4. Команда <i>chattr</i> : атрибуты файла, запрет изменения файла.....	51
4.5. Монтирование файловых систем.....	52
4.5.1. Команды <i>mount</i> и <i>umount</i>	52
4.5.2. Файлы устройств и монтирование.....	54
Жесткие диски.....	54
Приводы оптических дисков.....	56
Флешки и внешние жесткие диски.....	56
4.5.3. Опции монтирования файловых систем.....	56
4.5.4. Монтирование разделов при загрузке.....	58
4.5.5. Подробно о UUID и файле <i>/etc/fstab</i>	60
4.5.6. Монтирование флешек.....	62
4.6. Настройка журнала файловой системы ext3/ext4.....	63
4.7. Файловая система ext4.....	64
4.7.1. Сравнение ext3 и ext4.....	64
4.7.2. Совместимость с ext3.....	65
4.8. Особые команды.....	66
4.8.1. Команда <i>mkfs</i> : создание файловой системы.....	66
4.8.2. Команда <i>fsck</i> : проверка и восстановление файловой системы.....	66
4.8.3. Команда <i>chroot</i> : смена корневой файловой системы.....	67
4.8.4. Установка скорости CD/DVD.....	67
4.8.5. Монтирование каталога к каталогу.....	67
4.8.6. Команды поиска файлов.....	67
4.9. Многофункциональная команда <i>dd</i>	69
4.9.1. Копирование файлов с помощью команды <i>dd</i>	69
4.9.2. Разделение файла на несколько частей.....	70
4.9.3. Создание резервной копии жесткого диска.....	71
4.9.4. Создание архива с резервной копией всего жесткого диска.....	71
4.9.5. Уничтожение всех данных раздела жесткого диска.....	71
4.10. Команда <i>du</i>	72
Глава 5. Процессы.....	73
5.1. Оболочки, команды и программы.....	73
5.2. Родительский и дочерний процессы.....	74
5.3. Команды <i>kill</i> , <i>killall</i> , <i>xkill</i> и <i>ps</i>	75
5.4. Программы <i>top</i> и <i>htop</i> : кто больше всех расходует процессорное время?.....	77
5.5. Команды <i>nice</i> и <i>renice</i> : изменение приоритета процесса.....	80
5.6. Команда <i>fuser</i> : кто открыл ресурс?.....	80

5.7. Команда <i>lsof</i>	80
5.8. Команды управления сервисами	81
5.9. Команда <i>journalctl</i>	82
Глава 6. Выполнение команд и установка программ.....	85
6.1. Списки	85
6.1.1. Условные списки	85
6.1.2. Безусловные списки.....	86
6.2. Подстановка	87
6.2.1. Подстановка команды	87
6.2.2. Подстановка процесса.....	88
6.3. Команда как строка	89
6.3.1. Передача команды в виде аргумента	89
6.3.2. Перенаправление команды на стандартный ввод <i>bash</i>	89
6.4. Удаленное выполнение команды по SSH	90
6.5. Фоновое выполнение команд	90
6.6. Команда <i>nohup</i>	92
6.7. Планировщик <i>cron</i>	92
6.8. Установка программ командой <i>apt</i>	94
6.9. Установка программ командой <i>dnf</i>	96
Глава 7. Некоторые полезные команды.....	97
7.1. Команда <i>seq</i>	97
7.2. Фигурные скобки.....	98
7.3. Команда <i>find</i>	98
7.4. Команды <i>zip</i> и <i>unzip</i>	99
7.5. Команда <i>tar</i>	100
Глава 8. Команды для работы с текстом.....	102
8.1. Команда <i>sort</i> : сортировка файлов.....	102
8.2. Команда <i>diff</i> : сравнение файлов	102
8.3. Команда <i>diff3</i> : сравнение трех файлов	103
8.4. Команда <i>cmp</i> : сравнение двух файлов	104
8.5. Команда <i>comm</i> : еще одна команда для сравнения файлов.....	104
8.6. Команда <i>column</i> : разбивка текста на столбцы.....	105
8.7. Команда <i>egrep</i> : расширенный текстовый фильтр	105
8.8. Команда <i>expand</i> : замена символов табуляции пробелами	106
8.9. Команда <i>fmt</i>	107
8.10. Команда <i>fold</i>	107
8.11. Команда <i>grep</i> : текстовый фильтр	107
8.12. Команды <i>more</i> и <i>less</i> : страничный вывод	108
8.13. Команды <i>head</i> и <i>tail</i> : вывод начала и хвоста файла	108
8.14. Команда <i>look</i>	108
8.15. Команда <i>split</i> : разбиение файлов на несколько частей.....	108
8.16. Команда <i>unexpand</i> : замена пробелов символами табуляции	109
8.17. Команды <i>vi</i> , <i>nano</i> , <i>ee</i> , <i>mcedit</i> , <i>pico</i> : текстовые редакторы	109
8.18. Команда <i>sed</i> : потоковый текстовый редактор.....	113
8.19. Команда <i>wc</i> : подсчет слов в файле	114
8.20. Команда <i>gpg</i> : шифруем файлы паролем	114
8.21. Некоторые команды преобразования символов и форматов	116

Глава 9. Эффективное использование клавиатуры	117
9.1. Работа с окнами	117
9.2. Доступ к веб-браузерам из командной строки	119
9.3. Работаем с буфером обмена	121
Глава 10. Команды для работы с сетью и Интернетом	123
10.1. Команда <i>ifconfig</i> : управление сетевыми интерфейсами	123
10.2. Команда <i>ip</i>	125
10.3. Маршрутизация.....	126
10.3.1. Команда <i>netstat</i> : просмотр таблицы маршрутизации и другой сетевой информации.....	126
10.3.2. Команда <i>route</i> : изменение таблицы маршрутизации	130
10.4. Команды получения информации об узле	133
10.4.1. Получение информации о доменном имени.....	133
10.4.2. Команды <i>host</i> и <i>dig</i>	133
10.4.3. Утилита <i>DMitry</i>	135
10.5. Текстовые браузеры	135
10.6. Команда <i>ftp</i> : FTP-клиент	136
10.7. Команда <i>wget</i> : загрузка файлов	137
10.8. Команды для диагностики сети.....	139
10.9. Команда <i>ssh</i>	143
10.10. Сетевой сканер <i>nmap</i>	145
10.10.1. Что такое <i>nmap</i> ?.....	145
10.10.2. Где мне взять <i>nmap</i> ?	146
10.10.3. Примеры использования <i>nmap</i>	146
10.11. Команда <i>systemd-resolve</i>	148
10.12. Команда <i>netplan</i>	149
10.13. Команда <i>ufw</i>	150
Глава 11. Команды системного администратора	152
11.1. Программы разметки диска	152
11.1.1. Программа <i>fdisk</i>	152
11.1.2. Программа <i>Parted</i>	155
11.1.3. Таблица разделов GPT	159
11.1.4. Разметка диска на физическом сервере	160
11.1.5. Добавление диска на виртуальном сервере	161
11.1.6. Расширение существующего диска.....	164
11.1.7. Команды восстановления загрузчика GRUB2	166
Восстановление загрузчика GRUB2 при использовании UEFI.....	166
Команда <i>boot-repair</i>	167
11.2. Информация о системе и пользователях	167
11.2.1. Команда <i>uptime</i> : информация о работе системы.....	167
11.2.2. Команда <i>users</i> : информация о пользователях	168
11.2.3. Команды <i>w</i> , <i>who</i> , <i>ftpwho</i> и <i>whoami</i> : информация о пользователях	168
11.2.4. Команды <i>last</i> и <i>id</i>	169
11.2.5. Мониторинг работы системы	169
11.2.6. Команды <i>uname</i> и <i>lsb_release</i>	172
11.2.7. Команда <i>sensors</i>	172
11.3. Команды <i>date</i> и <i>timedatectl</i>	173

11.4. Планировщик <i>at</i>	174
11.4.1. Команда <i>at</i> : добавление задания.....	174
11.4.2. Команды <i>atq</i> и <i>atrm</i> : очередь заданий и удаление задания	175
11.5. Планировщик <i>cron</i>	175
11.6. Команды <i>free</i> и <i>df</i> : информация о системных ресурсах.....	175
11.7. Команды управления пользователями.....	176
11.8. Получение информации об оборудовании	176
11.8.1. Команда <i>dmesg</i>	176
11.8.2. Команда <i>lscpu</i> : получение информации о процессоре	177
11.8.3. Команда <i>lsusb</i> : список USB-устройств	178
11.8.4. Команда <i>lspci</i> : список PCI-устройств.....	178
11.8.5. Команда <i>lshw</i> : получение информации об устройствах	179
11.8.6. Команда <i>dmidecode</i>	179
11.8.7. Команда <i>lsblk</i>	181
11.8.8. Команда <i>badblocks</i>	182
11.9. Команда <i>rsync</i>	182

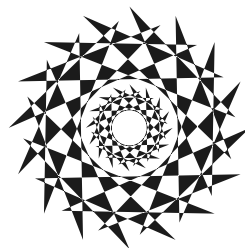
Глава 12. Автоматизация рутинных задач с помощью оболочки *bash*..... 184

12.1. Настройка <i>bash</i>	184
12.2. Автоматизация задач с помощью сценариев <i>bash</i>	185
12.3. Привет, мир!.....	185
12.4. Использование переменных в собственных сценариях.....	186
12.5. Передача параметров сценарию	187
12.6. Массивы и <i>bash</i>	188
12.7. Циклы	188
12.8. Условные операторы	189
12.9. Функции.....	191
12.10. Примеры сценариев.....	191
12.10.1. Сценарий мониторинга журнала	191
12.10.2. Переименование файлов	192
12.10.3. Преобразование систем счисления	192
12.10.4. Проверка прав пользователя.....	193
12.10.5. Генератор имени временного файла	193
12.10.6. Проверка свободного дискового пространства с уведомлением по электронной почте.....	194

Глава 13. Примеры полезных решений 195

13.1. Поиск дубликатов файлов	195
13.2. Сценарий <i>scanner</i>	196
13.3. Изменение прав доступа к файлам и каталогам.....	197
13.4. Аварийный перезапуск сервисов.....	197
13.4.1. Проверка работоспособности веб-сервера	197
13.4.2. Проверка работоспособности MySQL	198
13.4.3. Если «падают» процессы... ..	198
13.5. Поиск битых ссылок с помощью AWK	199
13.6. Считаем количество файлов в папке и подпапках	200
13.7. Резервное копирование базы данных.....	200

Предметный указатель 202



ГЛАВА 1

Командная строка: ОСНОВЫ ОСНОВ

1.1. Переключаемся из графического режима в консоль

Настоящий линуксоид должен уметь работать в консоли и использовать командную строку Linux. Ведь когда система Linux только появилась, существовала лишь консоль, — о графическом интерфейсе не было и речи. Знаете, почему UNIX и Linux отталкивали обычных пользователей, и с ними работали одни лишь профессионалы? Потому что у них не было хорошего графического интерфейса. Сейчас все изменилось — в Linux очень удобный графический интерфейс, который с удовольствием используют как новички, так и профессионалы (дождались наконец-то!), забывая о командной строке. Современные дистрибутивы Linux вообще ориентированы на работу в графическом режиме, а в официальных руководствах, которые можно найти в Интернете, о консоли вовсе не упоминается. А ведь она есть! И в этой главе мы поговорим о том, как добраться до командной строки в Linux. Совсем не обязательно работать полностью в текстовом режиме, вы можете использовать материал этой главы для эффективной работы с *терминалом* — эмулятором консоли, или для удаленного управления сервером, подключившись к нему по SSH.

Обычные пользователи в консоль ни ногой — даже принципиально: мол, зачем возвращаться в DOS? Под DOS они имеют в виду командную строку Linux. Да, вид ее не очень дружелюбный, но это только на первый взгляд. Стоит вам поработать в консоли, и вы поймете все ее преимущества. Начнем с того, что командная строка Linux намного удобнее командной строки DOS, — об этом мы еще поговорим. В консоли можно выполнять те же операции, что и в графическом режиме, причем намного быстрее. Понятно, что, даже несмотря на наличие консольного браузера, вы не будете смотреть в консольном режиме веб-страницы. Сейчас консоль используется в основном для тонкой настройки и администрирования локальных и удаленных машин. Причем для администрирования удаленной машины вам даже не нужен Linux — вы можете работать в привычной для себя Windows, установив какой-либо SSH-клиент (например, PuTTY или Bitvise).

По умолчанию в современных дистрибутивах при входе в систему запускается графический менеджер регистрации (рис. 1.1). Однако из всех правил могут быть ис-

ключения. Пример тому дистрибутив Slackware — в нем сначала нужно выполнить вход в консоли (рис. 1.2), а потом для запуска графического интерфейса ввести команду `startx`. Тем не менее любой дистрибутив можно настроить на запуск исключительно в консольном режиме — чтобы графика не загружалась. Как правило, так настраиваются серверы — там графический интерфейс не нужен.

Для входа в систему вам потребуется указать имя пользователя и пароль. После этого загрузится графическая среда KDE или GNOME (в зависимости от того, какая

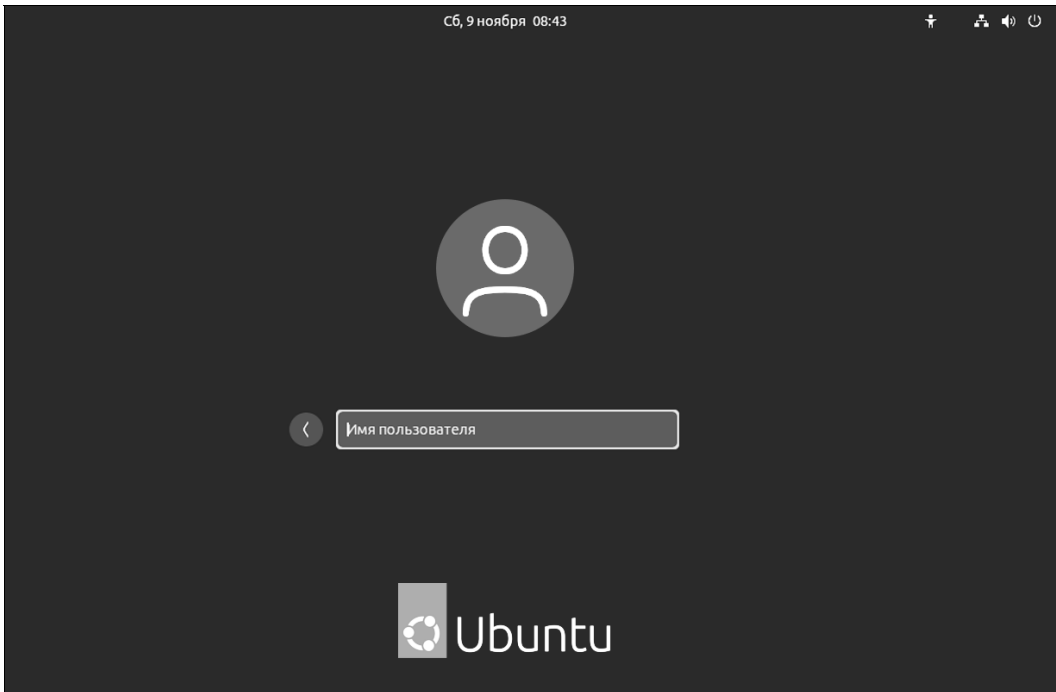



Рис. 1.1. Графический вход в систему (Ubuntu 24.10)

```
[  1.126497] RETbleed: WARNING: Spectre v2 mitigation leaves CPU vulnerable to
RETbleed attacks, data leaks possible!
[  2.223975] cpufreq: cpufreq_online: Failed to initialize policy for cpu: 0 (
-19)
[  2.224342] cpufreq: cpufreq_online: Failed to initialize policy for cpu: 1 (
-19)
[ 10.765174] piix4_smbus 0000:00:07.3: SMBus Host Controller not enabled!

Astra Linux 1.8.1 astra-31625 tty1
astra-31625 login: astra ←————— Имя пользователя (логин)
Password: ←————— Пароль (при вводе не отображается)

Login incorrect ←————— Логин и/или пароль неправильны
Login: astra
Password:
astra@astra-31625:~$ ←————— Вход выполнен (приглашение командной строки)
```

Рис. 1.2. Регистрация в консоли (фрагмент стартового окна Slackware)

графическая среда установлена в вашем дистрибутиве по умолчанию). Конечно, по желанию может быть загружена и какая-либо другая графическая среда, но обычно по умолчанию загружается KDE или GNOME. Для выбора графической среды нужно нажать кнопку **Тип сессии**, **Тип сеанса** или просто **Сеанс** (в Fedora и некоторых других дистрибутивах) — в ряде дистрибутивов эта кнопка также может быть представлена графическим значком. На рис. 1.3 показан экран входа дистрибутива Astra Linux — в нем команда **Консольный вход** почему-то помещена в меню, открывающееся по нажатию кнопки **Особое действие** , а не в меню **Тип сессии**.

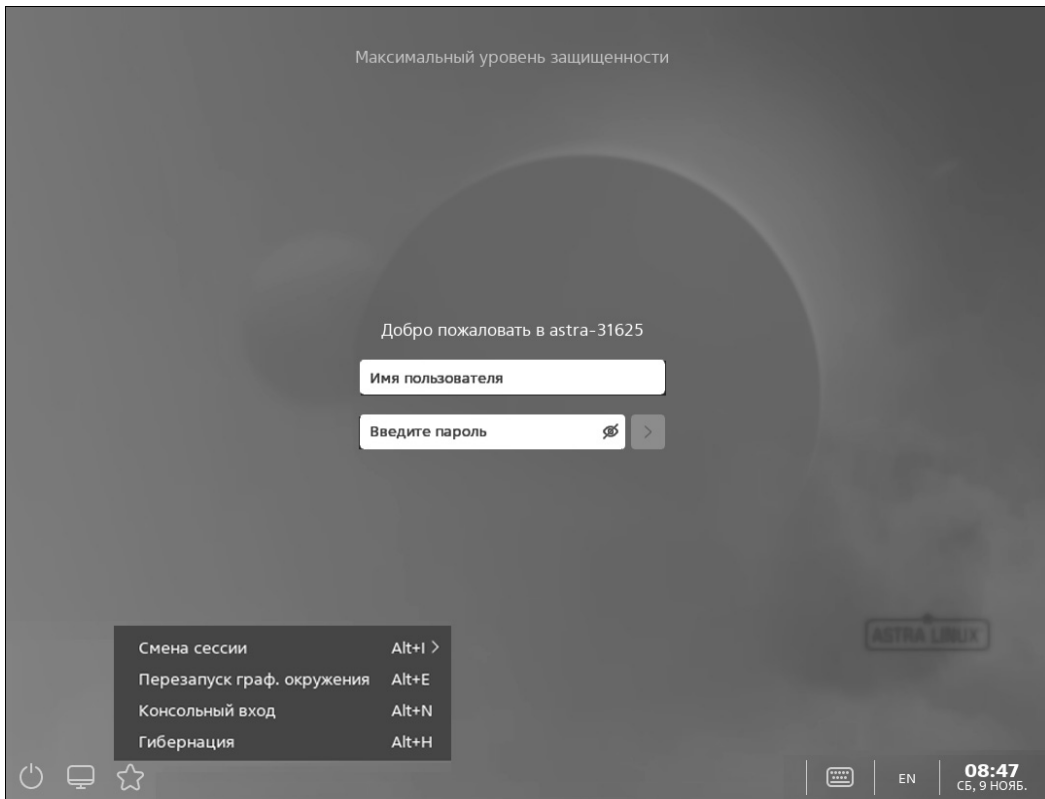


Рис. 1.3. Консольный вход (Astra Linux)

Итак, сейчас вы находитесь в графическом режиме. Чтобы перейти из графического режима в консоль (см. рис. 1.2), нажмите клавиатурную комбинацию $\langle \text{Ctrl} \rangle + \langle \text{Alt} \rangle + \langle F_n \rangle$, где n — номер консоли (от 1 до 6). То есть для перехода на первую консоль нужно нажать комбинацию клавиш $\langle \text{Ctrl} \rangle + \langle \text{Alt} \rangle + \langle F1 \rangle$, на вторую — $\langle \text{Ctrl} \rangle + \langle \text{Alt} \rangle + \langle F2 \rangle$ и т. д. Обратите внимание, что так можно перейти в консоль только из графического режима. Если вы уже находитесь в консоли, то для переключения между консолями служат комбинации клавиш от $\langle \text{Alt} \rangle + \langle F1 \rangle$ до $\langle \text{Alt} \rangle + \langle F6 \rangle$, а также $\langle \text{Alt} \rangle + \langle F7 \rangle$ — для перехода в графический режим. Чтобы вы лучше запомнили все эти комбинации, они сведены в табл. 1.1.

Таблица 1.1. Клавиши переключения между консолями и графическим режимом

Комбинация клавиш	Назначение
<Ctrl>+<Alt>+<Fn> (n — от 1 до 6)	Переключение из графического режима в консоль с номером n
<Alt>+<Fn> (n — от 1 до 6)	Переключение между консолями
<Alt>+<F7>	Переключение из консоли в графический режим

1.2. Графические (псевдо)-терминалы

Большинство современных дистрибутивов Linux оснащены графическим интерфейсом, который к тому же запускается по умолчанию. Поэтому понятно, что основной контингент пользователей не станет жертвовать удобным и привычным интерфейсом ради консоли.

Однако вместо того, чтобы переключаться в консоль, можно использовать терминалы — эмуляторы консоли. *Терминал* — это графическая программа (рис. 1.4), в окне которой вы можете вводить команды и видеть результат их выполнения. Терминал легко запустить из графического интерфейса — например, в Ubuntu для этого надо щелкнуть правой кнопкой мыши на рабочем столе и выбрать команду **Открыть терминал**, а в Astra Linux — выбрать команду **Терминал** из меню **Инструменты** (см. рис. 1.4).

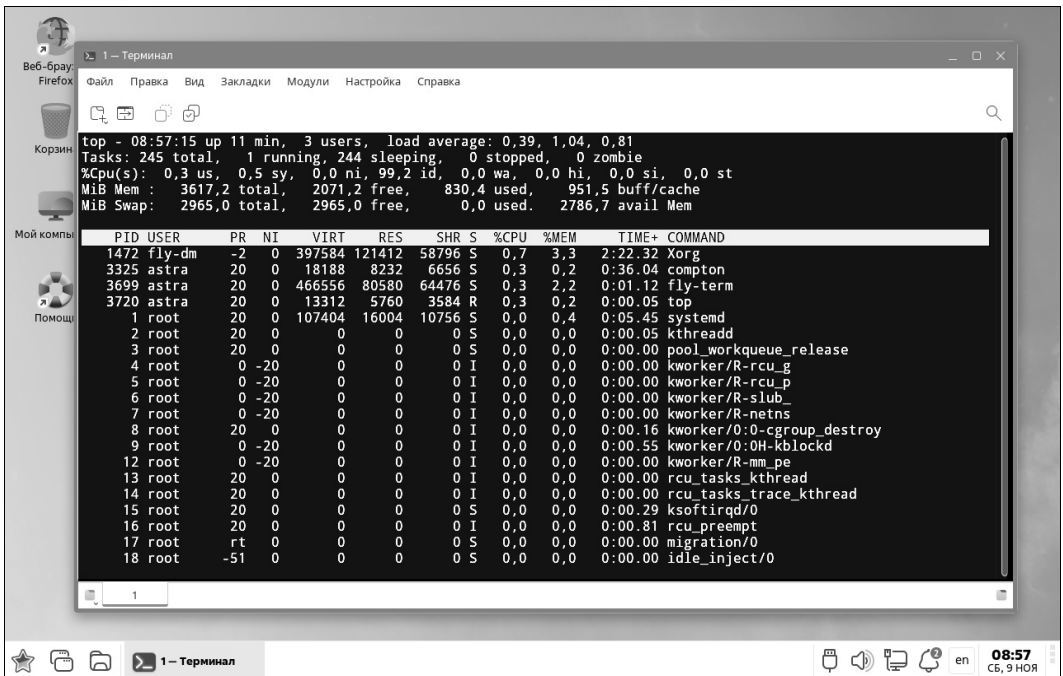


Рис. 1.4. Графический терминал (Astra Linux)

Работая в современных дистрибутивах Linux, как правило, нет необходимости переключаться в консоль, поскольку в них всегда можно запустить терминал, открыть консоль в окне терминала и работать с ней так же, как бы вы работали в консольном режиме. Использовать терминал или переключаться в консоль — дело вкусовых предпочтений.

Прежде чем перейти к следующему разделу — небольшой экскурс в историю. Но сначала войдите в систему, если вы это еще не сделали. Нажмите комбинацию клавиш <Ctrl>+<Alt>+<F1> и войдите в консоль, как было показано ранее. Затем нажмите комбинацию клавиш <Alt>+<F7>, чтобы вернуться в графический режим. Теперь откройте терминал и введите команду `w`, которая показывает всех пользователей, зарегистрированных в системе в текущий момент (не все учетные записи, а только тех пользователей, которые вошли в систему). На рис. 1.5 видно, что один и тот же пользователь **astra** вошел в систему тремя разными способами: через виртуальную консоль **tty1**, через графический менеджер дисплея **:0** и через **pts/0** — это и есть так называемый *псевдотерминал*.

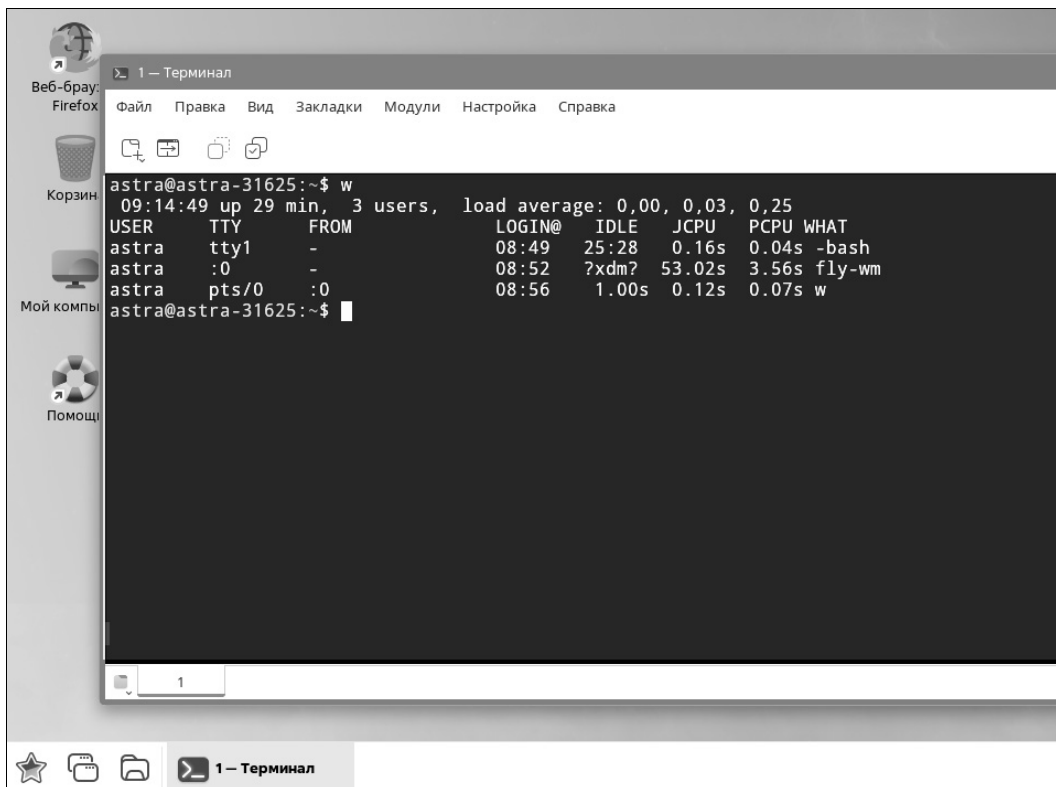


Рис. 1.5. Вывод команды `w`

А СЕЙЧАС ОБЕЩАННЫЙ ЭКСКУРС...

Когда-то очень давно, когда еще не было персональных компьютеров, использовались компьютеры размером с приличный холодильник или даже с целую комнату. Они назывались *мейнфреймами*. К ним подключались *терминалы* — рабочие места поль-

зователей, состоящие из алфавитно-цифрового монитора-дисплея и клавиатуры. Таких мест могло быть — в зависимости от модели компьютера — несколько. Несколько — это не 2 или 3, а сотни и тысячи рабочих мест. Понятно, что вся эта техника была очень дорогой и недоступной широким массам.

В 1981 году появились первые персональные компьютеры — IBM PC. По сравнению с мейнфреймами они были значительно дешевле, что помогло им быстро завоевать популярность у обычных пользователей, которые получили возможность приобретать их для домашнего и офисного использования. Теперь позволить себе собственный компьютер мог простой конечный пользователь, а не только биржа, банк или правительственное учреждение.

Мейнфреймы 1970–1980-х годов работали под управлением операционной системы UNIX. Понятно, что с появлением персональных компьютеров стали разрабатываться различные варианты UNIX, приспособленные для работы с ними (в том числе и Linux — с 1991 года). Но возникла проблема нескольких рабочих мест — к персональному компьютеру можно подключить только один монитор и одну клавиатуру. И даже если вы сумеете подключить к нему несколько клавиатур и несколько мониторов, организовать столько же рабочих мест у вас не получится — на то он и персональный компьютер. Однако в ядре UNIX, переключавшем в операционные системы персональных компьютеров, оставалась возможность работы с терминалами. И из положения вышли путем создания концепции *виртуальных терминалов*, переключение между которыми осуществляется нажатием комбинаций клавиш <Alt>+<F1> ... <Alt>+<F6>. Комбинация клавиш <Alt>+<F7> особая — она возвращает в графический режим, если графический интерфейс запущен.

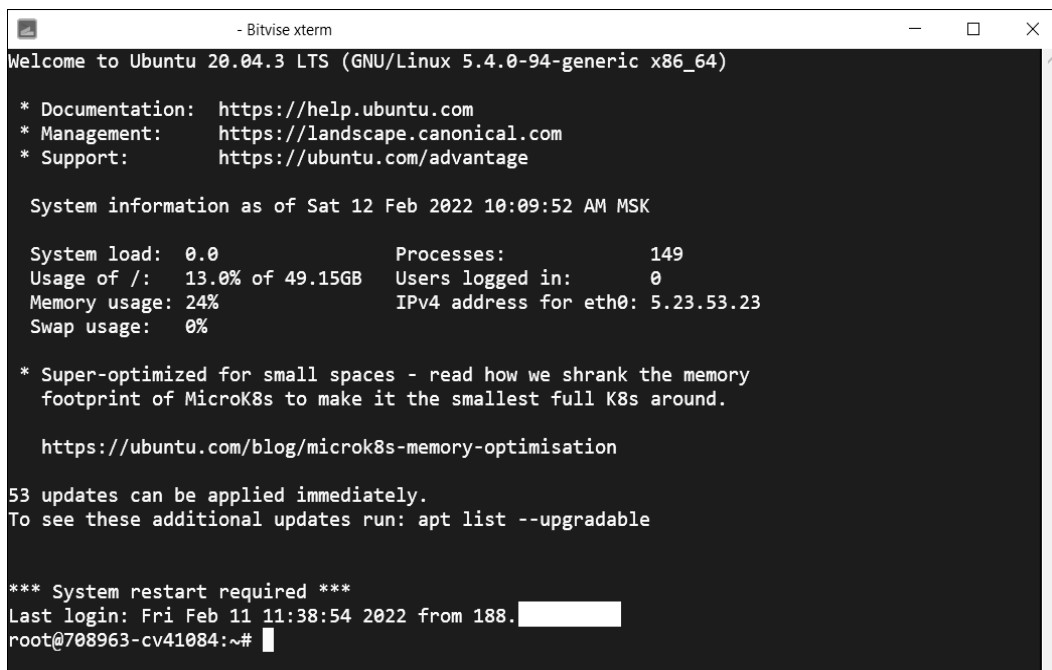
Так вот, вы можете зарегистрироваться на виртуальных консолях (если хотите — под разными логинами/паролями) шесть раз — т. е. как бы организовать на одном персональном компьютере шесть отдельных и не связанных между собой рабочих мест для разных пользователей (разумеется, одновременно они работать на нем не смогут — только попеременно, но эта задача сейчас и не ставится). На каждой из этих виртуальных консолей зарегистрированные пользователи могут запустить разные программы. Например, на одном — программу для мониторинга сети, на втором — `htop` для мониторинга системных ресурсов, на третьем — файловый менеджер, на четвертом — текстовый браузер и т. д. Другими словами, так можно организовать для себя комфортный режим использования нескольких рабочих мест даже без графического интерфейса.

В графическом интерфейсе, как уже было отмечено, для ввода команд можно запустить приложение **Терминал** (или несколько таких приложений). При этом терминалы, запущенные в графическом интерфейсе, стали называть *псевдотерминалами* — чтобы отличить их от обычных виртуальных терминалов.

1.3. Удаленная консоль

Помимо управления своей локальной машиной, вы можете управлять и удаленной Linux-системой. При этом все вводимые вами команды будут выполняться не на локальной машине, а на удаленной. Но к удаленной системе надо сначала подключиться.

Если вы работаете в Linux, то для подключения к удаленной машине достаточно выполнить команду `ssh` (подробно о ней рассказано в *разд. 10.9*), а вот если вы работаете в Windows, и возникла необходимость подключиться к Linux-серверу, придется воспользоваться SSH-клиентом — например, очень удобной и бесплатной программой Bitvise SSH Client (рис. 1.6).



```
- Bitvise xterm
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-94-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Sat 12 Feb 2022 10:09:52 AM MSK

System load:  0.0          Processes:            149
Usage of /:   13.0% of 49.15GB  Users logged in:    0
Memory usage: 24%          IPv4 address for eth0: 5.23.53.23
Swap usage:   0%

* Super-optimized for small spaces - read how we shrank the memory
  footprint of MicroK8s to make it the smallest full K8s around.

https://ubuntu.com/blog/microk8s-memory-optimisation

53 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

*** System restart required ***
Last login: Fri Feb 11 11:38:54 2022 from 188. [REDACTED]
root@708963-cv41084:~#
```

Рис. 1.6. Bitvise SSH Client: консоль удаленного сервера

Преимущество Bitvise в ее функциональности. Она поддерживает различные способы аутентификации, в том числе по ключу (а не только по паролю), а также содержит средства для работы с ключами. Удобно организована в ней и работа с буфером обмена — вы можете выделить текст на консоли мышью, и он будет скопирован в буфер обмена Windows. Для вставки текста в удаленную консоль из буфера обмена Windows достаточно нажать комбинацию клавиш <Shift>+<Insert> — это очень удобно, когда вы читаете какое-то руководство и настраиваете сервер по нему. Вам не надо будет перенабирать вручную файл конфигурации — вы сможете просто вставить его текст в открытую консоль. Экономия времени и страховка от опечаток вам гарантирована.

СОВЕТ

Работая в Linux с удаленной системой, соблюдайте осторожность: очень легко перепутать окна локального и удаленного терминалов (они выглядят одинаково) и вместо того, чтобы вводить команды на удаленной системе, выполнять их на локальной (или наоборот). Поэтому лучше на время работы с удаленной системой закрывать окно локального терминала — так уж точно не перепутаешь.

1.4. Завершение работы из консоли. Команды *poweroff*, *halt*, *reboot*, *shutdown*

Для выхода из консоли (чтобы ею никто не воспользовался во время вашего отсутствия) следует выполнить команду `logout` или ее аналог `exit`.

Перезагрузка компьютера осуществляется командой `reboot`. Кроме нее вы можете использовать еще две команды: `halt` и `poweroff`:

- ❑ команда `halt` завершает работу системы, но не выключает питание. Вы увидите сообщение `System is halted`, свидетельствующее о возможности выключения питания. Эта команда предназначена для старых компьютеров, не поддерживающих расширенное управление питанием;
- ❑ команда `poweroff` завершает работу системы и выключает ее питание.

Самая «продвинутая» команда — `shutdown`. Она позволяет завершить работу и перезагрузить систему в назначенное время. Предположим, вы хотите уйти пораньше, но компьютер нужно выключить ровно в 19:30 (вдруг некоторые пользователи задержались на работе, а вы выключите сервер, — некрасиво получится). Вот тут-то вам и поможет команда `shutdown`:

```
# shutdown -h 19:30 [сообщение]
```

ПОЯСНЕНИЕ

Здесь и далее решетка (#) означает, что команда должна быть выполнена от имени пользователя `root`. Если перед командой ничего не указано или же указан символ доллара (\$), команду можно выполнить от имени обычного пользователя. Если учетная запись `root` в вашем дистрибутиве отключена, используйте команду `sudo` для запуска команд завершения работы — например, `sudo reboot` или `sudo poweroff`.

Сообщение [сообщение] можно и не указывать, все равно Windows-пользователи его не увидят.

Если нужно завершить работу системы прямо сейчас, вместо времени укажите `now`:

```
# shutdown -h now
```

Для перезагрузки системы служит опция `-r`:

```
# shutdown -r now
```