Оглавление

Предисловие

Структура курса

В курсе даются основные понятия операционной системы Linux и важнейшие навыки работы в ней. Изложение сопровождается большим количеством практических примеров. Данный курс может рассматриваться как учебник для студентов, начинающих обучение по специальностям в области информатики и ещё не знакомых с ОС Linux. Он состоит из двух основных частей:

1. В первой части вводятся основные понятия и навыки, необходимые пользователю для того, чтобы начать грамотно работать в Linux. Здесь рассмотрены: пользователи с точки зрения системы, понятие терминал и работа с командной строкой, устройства файловой системы и работа с ней, права доступа в Linux, возможности командной оболочки, текстовые редакторы.

2. Вторая часть посвящена тем понятиям и навыкам, которые требуются для администрирования ОС Linux. Сюда входит обсуждение этапов загрузки системы, технологий работы с внешними устройствами, файловыми системами и сетью в Linux, администрирование системы посредством конфигурационных файлов, управление пакетами.

В завершающей лекции курса даётся обзор истории возникновения и развития Linux. Здесь же приведён обзор социального контекста, существенного для понимания ОС Linux и работы в ней: сообщество пользователей, лицензирование свободного программного обеспечения, место свободного ПО на современном рынке, дистрибутивы Linux и решения на базе Linux.

Примеры

Теоретическое изложение материала перемежается практическими примерами: показаны конкретные действия пользователя и их результат. Наиболее эффективный способ освоить материал курса — по ходу чтения лекций выполнять все примеры самостоятельно. Для этого потребуется доступ к установленному дистрибутиву Linux. Примеры подобраны с таким расчётом, чтобы результат был одинаковым в любом современном дистрибутиве.

В примерах действует один условный пользователь, работающий «в одном и том же месте»: все созданные им файлы сохраняются и используются в последующих лекциях. Он совершает типичные ошибки или, наоборот, делает всё правильно.

Примеры набраны моноширинным шрифтом (типа «пишущая машинка») и по возможности точно воспроизводят то, что пользователь видит на экране монитора (иногда с некоторыми купюрами). Команды, которые должен вводить пользователь, в примерах следуют за «приглашением» (оно завершается знаком «$»): все символы от «$» до конца строки — и есть ввод пользователя.

Названия объектов системы (имена файлов, программ и т. п.), встречающиеся в тексте, также набраны моноширинным шрифтом, их можно в неизменном виде вводить в качестве команд и т. п. Однако иногда такие строки для удобства чтения заключены в кавычки — в этом случае вводить кавычки не нужно.

Цель

Курс адресован студентам, начинающим обучение по специальностям в области информационных технологий, а также всем любознательным пользователям, желающим научиться грамотно и эффективно работать в Linux. Изучение курса не требует специальных знаний в области программирования.

Предварительные знания

Рекомендуемый уровень предварительной подготовки:

1. знакомство с устройством компьютера на уровне пользователя;
2. знакомство с устройством и принципом работы TCP/IP-сетей.

Лекция 1

Сеанс работы в Linux

1.1 Пользователи системы

Между включением питания компьютера и моментом, когда система готова к работе с пользователем, происходит процедура **загрузки системы**. В процессе загрузки будет запущена основная управляющая программа (**ядро**), определено и инициализировано имеющееся оборудование, активизированы сетевые соединения, запущены системные службы. В Linux во время загрузки на экран выводятся диагностические сообщения о происходящих событиях, и если всё в порядке и не возникло никаких ошибок, загрузка завершится выводом на экран приглашения ≪`login:`≫. Оно может быть оформлено по-разному, в зависимости от настройки системы оно может отображаться в красиво оформленном окне или в виде простой текстовой строки вверху экрана. Это приглашение к **регистрации в системе**: система ожидает, что в ответ на это приглашение будет введено **входное имя пользователя**, который начинает работу. Естественно, имеет смысл вводить такое имя, которое уже известно системе, чтобы она могла «узнать», с кем предстоит работать, выполнять команды «незнакомого» Linux откажется.

1.1.1 Многопользовательская модель разграничения доступа

Процедура регистрации в системе *обязательна* для Linux, работать в системе, не зарегистрировавшись под тем или иным именем пользователя, просто *невозможно*[1]. Для каждого пользователя определена сфера его полномочий в системе: программы, которые он может запускать, файлы, которые он имеет право просматривать, изменять, удалять. При попытке сделать что-то, выходящее за рамки полномочий, пользователь получит сообщение об ошибке. Такая строгость

[1] Вместо формального «зарегистрироваться в системе» обычно используют выражение «войти в систему». Операционная система представляется чем-то вроде замкнутого помещения, внутри которого можно оказаться, только успешно проникнув через «дверь» — пройдя процедуру регистрации.

может показаться необязательной, если пользователи компьютера доверяют друг другу, и особенно если у компьютера только один пользователь. Такая ситуация очень распространена на сегодняшний день, когда слово «компьютер» означает в первую очередь «персональный компьютер».

Однако **персональный компьютер** — довольно-таки позднее явление в мире вычислительной техники, получившее широкое распространение только в последние два десятилетия. Несколько раньше «компьютер» ассоциировался с огромным и дорогостоящим (занимавшем целые залы) вычислительным центром, предназначенным в первую очередь для решения разного рода научных задач. Машинное время такого центра стоит очень недёшево, и при этом его возможности необходимы одновременно многим сотрудникам, которые могут ничего не знать о работе друг друга. Требуется следить за тем, чтобы не произошло случайного вмешательства пользователей в чужую работу и повреждения чужих данных (файлов), выделять каждому машинное время (по возможности избежав простаивания), пространство на диске и при этом не допустить узурпирования всех ресурсов одним пользователем и его задачей, а равномерно делить ресурсы между всеми. Для такой системы принципиально важно знать, кому принадлежат задачи и файлы, поэтому и возникла необходимость выдавать доступ к ресурсам системы только после того, как пользователь *зарегистрируется в системе* под тем или иным именем.

Такая модель была реализована в **многопользовательской операционной системе** UNIX. Именно от неё Linux — также многопользовательская система — унаследовал принципы работы с пользователями. Но это не просто дань традиции или стремление к универсальности: многопользовательская модель позволяет решить ряд задач, весьма актуальных и для современных **персональных компьютеров**, и для серверов, работающих в локальных и глобальных сетях, и вообще в любых системах, одновременно выполняющих *разные* задачи, отвечают за которые *разные* люди.

Компьютер — это всего лишь инструмент для решения разного рода прикладных задач: от набора и распечатывания текста до вычислений. Сложность состоит в том, что для изменения этого инструмента и для работы с его помощью используются одни и те же операции: изменение файлов, выполнение программ. Получается, что, если не соблюдать осторожности, побочным результатом работы может стать выход из строя самой системы. Поэтому первоочередная задача для систем любого масштаба — разделять повседневную работу и изменение самой системы. В многопользовательской модели эта задача решается очень просто: разделяются **обычные пользователи** и **администратор (ы)**. В полномочия обычного пользователя входит все необходимое для выполнения прикладных задач, попросту говоря, для работы, однако ему запрещено выполнение действий, изменяющих саму систему. Таким образом можно избежать повреждения системы в результате ошибки пользователя (нажал не ту кнопку) или ошибки в программе, или даже по злому умыслу (например, вредительской программой-вирусом). Полномочия администратора обычно не ограничены.

Для персонального компьютера, с которым работают несколько человек, довольно важно обеспечить каждому независимую рабочую среду. Это снижает вероятность случайного повреждения чужих данных, а также позволяет каждому пользователю настроить внешний вид рабочей среды по своему вкусу и, например, сохранить расположение открытых окон между сеансами работы. Эта задача очевидным образом решается в многопользовательской модели: организуется **домашний каталог**, где хранятся данные пользователя, настройки внешнего вида и поведения его системы и т. п., доступ остальных пользователей к этому каталогу ограничивается.

Если компьютер подключён к глобальной или локальной сети, то вполне вероятно, что какую-то часть хранящихся на нем ресурсов имеет смысл сделать публичной и доступной по сети. Напротив, часть данных, скорее всего, делать публичными не следует (например, личную переписку). Ограничив публичный доступ пользователей к персональным данным друг друга, мы решим и эту задачу.

Именно благодаря гибкости многопользовательской модели разграничения доступа она используется сегодня не только на серверах, но и на домашних персональных компьютерах. В самом простом варианте — для персонального компьютера, на котором работает только один человек — эта модель сводится к двум пользователям: обычному пользователю для повседневной работы и администратору — для настройки, обновления, дополнения системы и исправления неполадок. Но даже в таком сокращённом варианте это даёт целый ряд названных выше преимуществ.

1.1.2 Учётные записи

Конечно, система может быть «знакома» с человеком только в переносном смысле: в ней должна храниться запись о пользователе с таким именем и о связанной с ним системной информации — **учётная запись**. Английский эквивалент термина **учётная запись** — **account**, «счёт». Именно с учётными записями, а не с самими пользователями, и работает система. В действительности, соотношение учётных записей и пользователей-людей в Linux обычно не является однозначным: несколько человек могут использовать одну учётную запись — система не может их различить. И в то же время в Linux имеются учётные записи для **системных пользователей**, от имени которых работают некоторые программы и которые не предназначены для работы людей.

учётная запись
Объект системы, при помощи которого Linux ведёт учёт работы пользователя в системе. Учётная запись содержит данные о пользователе, необходимые для регистрации в системе и дальнейшей работы с ней.

Учётные записи могут быть созданы во время установки системы или после установки. Подробно процедура создания учётных записей (добавления пользователей) описана в лекции «Конфигурационные файлы» (12).

Главное для человека в учётной записи — её название, **входное имя пользователя**. Именно о нём спрашивает система, выводя приглашение «login:». Помимо входного имени в учётной записи содержатся некоторые сведения о пользователе, необходимые системе для работы с ним. Ниже приведён список этих сведений.

входное имя
Название учётной записи пользователя, которое нужно вводить при регистрации пользователя в системе.

1.1.2.1 Идентификатор пользователя

Linux связывает **входное имя** с **идентификатором пользователя** в системе — **UID** (User ID). **UID** — это положительное целое число, по которому система и отслеживает пользователей[2]. Обычно это число выбирается автоматически при регистрации учётной записи, однако оно не может быть совершенно произвольным. В Linux есть некоторые соглашения относительно того, каким типам пользователей могут быть выданы идентификаторы из того или иного диапазона. В частности, **UID** от «0» до «100» зарезервированы для **псевдопользователей**[3].

идентификатор пользователя
Уникальное число, однозначно идентифицирующее **учётную запись** пользователя в Linux. Таким числом снабжены все **процессы** Linux и все объекты **файловой системы**. Используется для персонального учёта действий пользователя и определения **прав доступа** к другим объектам системы.

1.1.2.2 Идентификатор группы

Кроме идентификационного номера пользователя с учётной записью связан **идентификатор группы**. **Группы пользователей** применяются для организации доступа нескольких пользователей к некоторым ресурсам. У группы, так же, как и у пользователя, есть имя и идентификационный номер — **GID** (Group ID). В Linux пользователь должен принадлежать как минимум к одной группе — **группе по умолчанию**. При создании учётной записи пользователя обычно создаётся и группа, имя которой совпадает с **входным именем**[4], именно эта группа будет использоваться как группа по умолчанию для этого пользователя. Пользователь может входить более чем в одну группу, но в учётной записи указывается только номер группы по умолчанию.

[2]Это может оказаться важным, например, в такой ситуации: учётную запись пользователя с именем test удалили из системы, а потом добавили снова. Однако с точки зрения системы это уже другой пользователь, потому что у него другой UID.

[3]Обычно Linux выдаёт нормальным пользователям UID, начиная с «500» или «1000».

[4]Как правило, численное значение GID в этом случае совпадает со значением UID.

1.1.2.3 Полное имя

Помимо **входного имени** в учётной записи содержится и **полное имя** (имя и фамилия) использующего данную учётную запись человека. Конечно, пользователь может указать что угодно в качестве своего имени и фамилии. Полное имя необходимо не столько системе, сколько людям — чтобы иметь возможность определить, кому принадлежит учётная запись.

1.1.2.4 Домашний каталог

Файлы всех пользователей в Linux хранятся раздельно, у каждого пользователя есть собственный **домашний каталог**, в котором он может хранить свои данные. Доступ других пользователей к домашнему каталогу пользователя может быть ограничен. Информация о домашнем каталоге обязательно должна присутствовать в учётной записи, потому что именно с него начинает работу пользователь, зарегистрировавшийся в системе.

1.1.2.5 Командная оболочка

Каждому пользователю нужно предоставить способ взаимодействовать с системой: передавать ей команды и получать её ответы. Для этой цели служит специальная программа — **командная оболочка** (или **интерпретатор командной строки**), она должна быть запущена для каждого пользователя, зарегистрировавшегося в системе. Поскольку в Linux доступно несколько разных командных оболочек, в учётной записи указано, какую из командных оболочек нужно запустить для данного пользователя. Если специально не указывать командную оболочку при создании учётной записи, она будет назначена по умолчанию, вероятнее всего это будет **bash**.

> **интерпретатор командной строки**
> Программа, используемая в Linux для организации диалога человека и системы. Командный интерпретатор имеет три основных ипостаси: (1) редактор и анализатор команд в **командной строке**, (2) высокоуровневый системно-ориентированный язык программирования, (3) средство организации взаимодействия команд друг с другом и с системой.

1.1.3 Понятие «администратор»

В Linux есть ровно один пользователь, полномочия которого в системе принципиально отличаются от полномочий остальных пользователей — это пользователь с идентификатором «0». Обычно учётная запись пользователя с UID=0 называется **root** (англ. «корень»). Пользователь root — это «администратор» системы Linux, учётная запись для root обязательно присутствует в любой системе Linux, даже если в ней нет никаких других учётных записей. Пользователю с таким UID разрешено выполнять *любые* действия в системе, а значит, любая

ошибка или неправильное действие может повредить систему, уничтожить данные и привести к другим печальным последствиям. Поэтому *категорически* не рекомендуется регистрироваться в системе под именем root для повседневной работы. Работать в root следует только тогда, когда это действительно необходимо: при настройке и обновлении системы, восстановлении после сбоев.

Именно root обладает достаточными полномочиями для создания новых учётных записей.

1.2 Регистрация в системе

Вернёмся теперь к нашей загруженной операционной системе Linux, которая по-прежнему ожидает ответа на своё приглашение «`login:`». Если Ваша система настроена таким образом, что это приглашение оформлено в виде графического окна в центре экрана, нажмите комбинацию клавиш *Ctrl+Alt+F1* — произойдёт переключение видеорежима и Вы увидите перед собой чёрный экран с примерно следующим текстом:

Пример 1.1. Начальное приглашение к регистрации

```
Welcome to Some Linux / tty1

localhost login:
```

Мы переключились в так называемый **текстовый режим**, в котором нам недоступны возможности графических интерфейсов: рисование окон произвольной формы и размера, поддержка миллионов цветов, отрисовка изображений. Все возможности текстового режима ограничены набором текстовых и псевдографических символов и несколькими десятками базовых цветов. Однако в Linux в текстовом режиме можно выполнять практически любые действия в системе (кроме тех, которые требуют непосредственного *просмотра* изображений). Текстовый режим в Linux — это полнофункциональный способ управления системой. В различных реализациях Linux работа в графическом режиме может выглядеть очень по-разному[5], более того, графический режим может быть даже недоступен после установки системы без специальной настройки. Текстовый режим, напротив, доступен в любой реализации Linux и всегда выглядит практически одинаково. Именно поэтому все дальнейшие примеры и упражнения мы будем проделывать в текстовом режиме, возможностей которого будет достаточно для освоения всего излагаемого в курсе материала.

Первая строка в примере — это просто приглашение, она носит информационный характер. Существует очень много различных реализаций Linux (существующие реализации будут обсуждаться в лекции «Политика свободного лицензирования. История Linux: от ядра к дистрибутивам» (18)), и в каждом из

[5]Разнообразие графических интерфейсов Linux гораздо выше, чем, например, в **win**, поэтому составить учебный курс, не ориентируясь специально на ту или иную версию, просто невозможно.

них принят свой формат первой пригласительной строки, хотя почти наверняка там будет указано, с какой именно версией Linux Вы имеете дело, и, возможно, будут присутствовать ещё некоторые параметры. В наших примерах мы будем использовать условную реализацию Linux — «Some Linux».

Вторая строка начинается с **имени хоста** — собственного имени системы, установленной на данном компьютере. Это имя существенно в том случае, если компьютер подключён к локальной или глобальной сети, если же он ни с кем более не связан, это имя может быть любым. Обычно имя хоста определяется уже при установке системы, однако в нашем случае этого сделано не было, и используется вариант по умолчанию — «`localhost`». Заканчивается эта строка собственно приглашением к регистрации в системе — словом «`login:`».

Теперь понятно, что в ответ на это приглашение мы должны ввести **входное имя**, для которого есть соответствующая **учётная запись** в системе. В правильно установленной операционной системе Linux должна существовать как минимум одна учётная запись для **обычного пользователя**. Во всех дальнейших примерах у нас будет участвовать Мефодий Кашин, владелец учётной записи «`methody`» в системе «Some Linux». Вы можете пользоваться для выполнения примеров любой учётной записью, которая создана в Вашей системе (естественно, кроме root).

Итак, Мефодий вводит своё входное имя в ответ на приглашение системы:

Пример 1.2. Регистрация в системе

```
Welcome to Some Linux / tty1
localhost login: Methody

Password:
Login incorrect

login:
```

В ответ на это система запрашивает **пароль**. Пароль Мефодия нам неизвестен, поскольку он его никому не говорит. Когда Мефодий вводил свой пароль, на экране монитора он не отображался (это сделано, чтобы пароль нельзя было подсмотреть), однако Мефодий точно знает, что не сделал опечатки. Тем не менее система отказала ему в регистрации, выдав сообщение об ошибке («`Login incorrect`»). Если же внимательно посмотреть на введённое им имя пользователя, можно заметить, что оно начинается с заглавной буквы, в то время как учётная запись называется «`methody`». Linux всегда делает различие между заглавными и строчными буквами, поэтому «`Methody`» для него — уже другое имя. Теперь Мефодий повторит попытку:

Пример 1.3. Успешная регистрация в системе

```
login: methody
```

```
Password:
[methody@localhost methody]$
```

В этот раз регистрация прошла успешно, о чём свидетельствует последняя строка примера — **приглашение командной строки**. Приглашение — это подсказка, выводимая **командной оболочкой** и свидетельствующая о том, что система готова принимать команды пользователя. Приглашение может быть оформлено по-разному, более того, пользователь может сам управлять видом приглашения (подробнее это будет рассмотрено в лекции «Возможности командной оболочки» (8)), но почти наверняка в приглашении содержатся **входное имя** и **имя хоста** — в нашем примере это ≪`methody`≫ и ≪`localhost`≫ соответственно. Заканчивается приглашение чаще всего символом ≪`$`≫. Это **командная строка**, в которой будут отображаться все введённые пользователем с клавиатуры команды, а при нажатии на клавишу *Enter* содержимое командной строки будет передано для исполнения системе.

1.2.1 Идентификация (authentication)

Когда система выводит на экран приглашение командной строки после того, как правильно введены имя пользователя и пароль, это означает, что произошла **идентификация пользователя** (authentication, «проверка подлинности»). Пароль может показаться излишней сложностью, но у системы нет другого способа удостовериться, что за монитором находится именно тот человек, который имеет право на использование данной учётной записи.

Конечно, процедура идентификации имеет очевидное значение для систем, к которым имеют непосредственный или сетевой доступ многие не связанные друг с другом пользователи. Процедура идентификации даёт уверенность, что к такой системе не получит доступ случайный человек, не имеющий права использовать её ресурсы и хранящуюся там информацию. Одновременно она даёт определённую гарантию безопасности от злонамеренного вмешательства: даже если навредить попытается пользователь, имеющий учётную запись, его действия будут зарегистрированы в системе (поскольку системе всегда известно, от имени какой учётной записи выполняются те или иные действия), и злоумышленника можно будет найти и остановить.

Для тех пользователей, кому процедура идентификации кажется утомительной и необязательной (например, единственным пользователям персональных компьютеров), существует возможность получить доступ к системе, минуя процедуру идентификации. Для этой цели служит программа `autologin`. Она предоставляет доступ к работе с графическим интерфейсом сразу после загрузки системы, не запрашивая имя пользователя и пароль. В действительности, `autologin` запускает все программы от имени одного пользователя, зарегистрированного в системе. Например, Мефодий мог бы использовать свою учётную запись `methody` для автоматического входа в систему. Однако у этого подхода есть свои минусы:

- Теряется возможность определить, кто, что и когда делал в системе, потому что все реальные пользователи работают с одной учётной записью, с точки зрения системы все они — один и тот же пользователь.
- Вся личная информация этого пользователя становится «общественной».
- Пароль легко забывается (пароль всё равно есть у любого пользователя), потому что его не нужно вводить каждый день. При этом `autologin` даёт доступ только человеку, сидящему перед монитором и только к работе с графическим интерфейсом. Если же потребуется, например, скопировать файлы с Вашего компьютера по сети, пароль всё равно нужно будет вводить.

Учитывая все перечисленные минусы, можно заключить, что использовать `autologin` разумно только в тех системах, которые не подключены к локальной или глобальной сети, и к которым при этом открыт публичный доступ (например, в библиотеке).

1.2.2 Смена пароля

Если учётная запись была создана не самим пользователем, а администратором многопользовательской системы (скажем, администратором компьютерного класса), скорее всего был выбран тривиальный пароль с тем расчётом, что пользователь его изменит при первом же входе в систему. Распространены тривиальные пароли «`123456`», «`empty`» и т. п. Поскольку пароль — это единственная гарантия, что Вашей учётной записью не воспользуется никто, кроме Вас, есть смысл выбирать в качестве пароля неочевидные последовательности символов. В Linux нет серьёзных ограничений на длину пароля или входящие в него символы (в частности, использовать пробел *можно*), но нет смысла делать пароль слишком длинным — сразу возрастает опасность его забыть. Надёжность паролю придаёт его *непредсказуемость*, а не длина. Например, пароль, представляющий собой Ваше имя или повторяющий название учётной записи, *очень предсказуем*. Наименее предсказуемы пароли, представляющие собой случайную комбинацию прописных и строчных букв, цифр, знаков препинания, но их и труднее всего запомнить.

Пользователь может в любой момент поменять свой пароль. Единственное, что требуется для смены пароля — знать текущий пароль. Допустим, Мефодий придумал более удачный пароль и решил его поменять. Он уже зарегистрирован в системе, поэтому ему нужно только набрать в командной строке команду `passwd` и нажать *Enter*.

Пример 1.4. Смена пароля

```
[methody@localhost methody]$ passwd
Changing password for methody.
```

```
Enter current password:

You can now choose the new password or passphrase.

A valid password should be a mix of upper and lower case letters,
digits, and other characters.  You can use an 8 character long
password with characters from at least 3 of these 4 classes, or
a 7 character long password containing characters from all the
classes.  An upper case letter that begins the password and a
digit that ends it do not count towards the number of character
classes used.

A passphrase should be of at least 3 words, 12 to 40 characters
long and contain enough different characters.

Alternatively, if noone else can see your terminal now, you can
pick this as your password: "spinal&state:buy".

Enter new password:
```

Набрав в командной строке «`passwd`», Мефодий запустил программу `passwd`, которая предназначена именно для замены информации о пароле в учётной записи пользователя. Она вывела приглашение ввести текущий пароль («`Enter current password`»), а затем, в ответ на правильно введённый пароль, предложила подсказку про грамотное составление пароля и даже вариант надёжного пароля, который Мефодий вполне может использовать, если никто в данный момент не видит его монитора. При каждом запуске `passwd` генерирует новый случайный пароль и предлагает его пользователю. Однако Мефодий не воспользовался подсказкой и придумал пароль сам.

Пример 1.5. Смена пароля (продолжение)

```
Enter new password:
Weak password: not enough different characters or classes for this length.
Try again.

. . .

Enter new password:
```

В данном случае операция не удалась, поскольку с точки зрения `passwd` пароль, придуманный Мефодием, оказался слишком простым[6]. В следующий раз ему придётся ввести более сложный пароль. `passwd` запрашивает новый пароль

[6] В разных дистрибутивах Linux используется разные версии программы `passwd`, поэтому не всегда она будет столь придирчива, как в дистрибутиве Мефодия.

дважды, чтобы удостовериться, что в первый раз не было сделано опечатки, если же всё в порядке, она выведет сообщение о том, что операция смены пароля прошла успешно, и завершит работу, вернув Мефодию приглашение командной строки.

Пример 1.6. Пароль изменён

```
Enter new password:
Re-type new password:
passwd: All authentication tokens updated successfully
[methody@localhost methody]$
```

Придирчивость, с которой `passwd` относится к паролю пользователя, не случайна. Пароль пользователя — одно из самых важных и зачастую одно из самых слабых мест безопасности системы. Отгадавший Ваш пароль (причём не имеет значение, человек это сделал или злонамеренная программа) получит доступ к ресурсам системы ровно в том объёме, в котором он предоставляется Вам, сможет читать и удалять Ваши файлы и т. п. Особенно это важно в случае пароля администратора, потому что его полномочия в системе гораздо шире, а действия от его имени могут повредить и саму систему. Обычному пользователю в некоторых обстоятельствах также могут быть переданы полномочия администратора (этот вопрос будет подробно обсуждаться в лекции «Права доступа» (6)), в таком случае не менее важно, чтобы и его пароль был надёжным.

Пароль пользователя root изначально назначается при установке системы, однако он может быть изменён в любой момент впоследствии точно так же, как и пароль обычного пользователя.

1.3 Одновременный доступ к системе

То, что Linux — многопользовательская и многозадачная система, проявляется не только в разграничении прав доступа, но и в организации рабочего места. Каждый компьютер, на котором работает Linux, предоставляет возможность зарегистрироваться и получить доступ к системе одновременно нескольким пользователям. Даже если в распоряжении всех пользователей есть только один монитор и одна системная клавиатура, эта возможность небесполезна: одновременная регистрация в системе нескольких пользователей позволяет работать по очереди без необходимости каждый раз завершать все начатые задачи (закрывать все окна, прерывать исполнение всех программ) и затем возобновлять их. Более того, ничто не препятствует зарегистрироваться в системе несколько раз под одним и тем же **входным именем**. Таким образом, можно получить доступ к одним и тем же ресурсам (своим файлам) и организовать параллельную работу над несколькими задачами.

1.3.1 Виртуальные консоли

Характерный для Linux способ организации параллельной работы пользователей — **виртуальные консоли**.

Допустим, что Мефодий хочет зарегистрироваться в системе ещё раз, чтобы иметь возможность следить за выполнением двух программ одновременно. Он может сделать это, не покидая текстового режима: достаточно нажать комбинацию клавиш *Alt+F2*, и на экране появится новое приглашение к регистрации в системе.

Пример 1.7. Вторая виртуальная консоль

```
Welcome to Some Linux / tty2
localhost login: methody
Password:
[methody@localhost methody]$
```

Мефодий ввёл свой новый пароль и получил приглашение командной строки, аналогичное тому, которое мы уже видели в предыдущих примерах. Нажав комбинацию клавиш *Alt+F1*, Мефодий вернётся к только что покинутой им командной строке, в которой он выполнял команду `passwd` для смены пароля. Приглашение в обоих случаях выглядит одинаково, и это не случайно — обе командные строки предоставляют совершенно эквивалентный доступ к системе, в любой из них можно выполнять любые доступные команды.

Наблюдательный Мефодий обратил внимание, что в последнем примереперва строка приглашения оканчивается словом «`tty2`». «`tty2`» — это обозначение *второй* **виртуальной консоли**. Можно переключаться между виртуальными консолями так, как если бы Вы переходили от одного монитора с клавиатурой к другому, подавая время от времени команды и следя за выполняющимися там программами. По умолчанию в Linux доступно не менее 6-ти виртуальных консолей, переключаться между которыми можно при помощи сочетания клавиши *Alt* с одной из функциональных клавиш (*F1*—*F6*), с каждым сочетанием связана соответствующая по номеру виртуальная консоль. Виртуальные консоли обозначаются «`ttyN`», где «*N*» — номер виртуальной консоли.

виртуальная консоль
Виртуальные консоли — это несколько параллельно выполняемых операционной системой программ, предоставляющих пользователю возможность зарегистрироваться в системе в текстовом режиме и получить доступ к командной строке.

Во многих дистрибутивах Linux одна из виртуальных консолей по умолчанию не может быть использована для регистрации пользователя, однако она не менее, если не более полезна. Если Мефодий нажмёт *Alt+F12*, он увидит консоль, заполненную множеством сообщений системы о происходящих событиях. В частности, там он может обнаружить две записи о том, что в системе зарегистрирован

пользователь «`methody`». На эту консоль выводятся сообщения обо всех важных событиях в системе: регистрации пользователей, выполнении действий от имени администратора (root), подключении устройств и подгрузке драйверов к ним и многое другое.

Пример двенадцатой виртуальной консоли показывает, что виртуальные консоли — довольно гибкий механизм, поддерживаемый Linux, при помощи которого можно решать разные задачи, а не только организацию одновременного доступа к системе. Для того, чтобы на виртуальной консоли появилось приглашение `login:` после загрузки системы, для каждой такой консоли должна быть запущена программа `getty`. Попробуйте нажать *Alt+F10* — с большой вероятностью Вы увидите просто чёрный экран. Десятая виртуальная консоль поддерживается системой, однако чёрный экран означает, что для этой консоли не запущена никакая программа, поэтому воспользоваться её существованием не получится. Для каких именно консолей будет запущена программа `getty` — определяется настройкой конкретной системы. Впоследствии эта настройка может быть изменена пользователем. О том, как это может быть сделано, речь пойдёт в лекции «Этапы загрузки системы» (10).

1.3.2 Графические консоли

Впрочем, как ни широки возможности текстового режима, Linux ими не ограничен. Подробно работа в графическом режиме будет разбираться в последующих лекциях (см. лекцию «Графический интерфейс (X11)» (16)). Сейчас важно заметить, что если при загрузке системы приглашение «`login:`» было представлено в виде графического окна, можно вернуться к этому приглашению, нажав комбинацию клавиш *Ctrl+Alt+F7*. Процедура регистрации здесь будет совершенно аналогична регистрации в текстовом режиме. С той разницей, что после **идентификации** пользователя (правильно введённого имени пользователя и пароля) Вы увидите не приглашение командной строки, а графическую рабочую среду. Как именно она будет выглядеть — зависит от того, какую систему Вы используете, и как она настроена.

Кроме того, что несколько пользователей (или несколько «копий» одного и того же пользователя) могут работать параллельно на разных виртуальных консолях, они могут параллельно зарегистрироваться и работать параллельно в разных графических средах. Обычно в стандартно настроенной Linux-системе можно организовать не менее трёх графических консолей, работающих одновременно. Переключаться между ними можно при помощи сочетаний клавиш *Ctrl+Alt+F7* — *Ctrl+Alt+F9*.

Чтобы переключиться из графического режима в одну из текстовых виртуальных консолей, достаточно нажать комбинацию клавиш *Ctrl+Alt+FN*, где «*N*» — номер необходимой виртуальной консоли.

1.4 Простейшие команды

Работа в Linux при помощи командной строки напоминает *диалог* с системой: пользователь вводит команды (реплики), получая от системы ответные реплики, содержащие сведения о произведённых операциях, дополнительные вопросы к пользователю, сообщения об ошибках или просто молчаливое согласие выполнить следующую команду[7].

Простейшая команда в Linux состоит из одного «слова» — названия программы, которую необходимо выполнить. Одну такую команду (`passwd`) Мефодий уже использовал для того, чтобы изменить свой пароль. Теперь Мефодий решил вернуться на одну из виртуальных консолей, на которой он зарегистрировался, и попробовать выполнить несколько простых команд.

Пример 1.8. Команда whoami

```
[methody@localhost methody]$ whoami
methody
[methody@localhost methody]$
```

Название этой команды происходит от английского выражения «Who am I?» («Кто я?»). В ответ на эту команду система вывела только одно слово: «methody» и завершила свою работу, о чём свидетельствует вновь появившееся **приглашение командной строки**. Программа `whoami` возвращает название учётной записи того пользователя, от имени которого она была выполнена. Эта команда полезна в системах, в которых работает много разных пользователей, чтобы не воспользоваться по ошибке чужой учётной записью. Однако, в приглашении командной строки зачастую указывается имя пользователя (как и в наших примерах), поэтому без команды `whoami` можно обойтись. Следующий пример демонстрирует программу, которая выдаст Мефодию уже больше полезной информации: `who` («Кто»).

Пример 1.9. Команда who

```
[methody@localhost methody]$ who
methody       tty1         Sep 23 16:31 (localhost)
methody       tty2         Sep 23 17:12 (localhost)
[methody@localhost methody]$
[methody@localhost methody]$ who am i
methody       tty2         Sep 23 17:12 (localhost)
[methody@localhost methody]$
```

[7]Реплики в таком диалоге строго чередуются, а собеседники не могут говорить одновременно — в естественном диалоге так никогда не происходит. Скорее это напоминает диалог в учебнике иностранного языка. Однако и в диалоге с Linux у собеседников есть возможность «перебить» друг друга — об этом речь пойдёт в последующих лекциях.

`who` выводит список пользователей, которые в настоящий момент зарегистрированы в системе (вошли в систему). Данная программа выводит по одной строке на каждого зарегистрированного пользователя: в первой колонке указывается **имя пользователя**, во второй — «точка входа» в систему, далее следует дата и время регистрации и **имя хоста**. Из выведенной `who` информации можно заключить, что в системе дважды зарегистрирован пользователь `methody`, который сначала зарегистрировался на первой виртуальной консоли (`tty1`), а примерно через сорок минут — на второй (`tty2`). Конечно, Мефодий и так это знает, однако администратору больших систем, когда пользователи могут зарегистрироваться со многих компьютеров и даже по Сети, программа `who` может быть очень полезна. Могло создаться впечатление, что `who` — очень умная программа, понимающая английский, но это не так. Из всех английских слов она понимает только сочетание «am i» — таким способом Мефодий узнал, за какой консолью он сейчас работает.

Ещё одна программа, выдающая информацию о пользователях, работавших в системе в последнее время — `last`[8]. Выводимые этой программой строки напоминают вывод программы `who`, с той разницей, что здесь перечислены и те пользователи, которые уже завершили работу.

Пример 1.10. Команда last

```
[methody@localhost methody]$ last
methody  tty2         localhost        Thu Sep 23 17:12   still logged in
methody  tty1         localhost        Thu Sep 23 16:31   still logged in
cacheman ???          localhost        Thu Sep 23 16:15 - 16:17 (00:01)
cacheman ???          localhost        Thu Sep 23 16:08 - 16:08 (00:00)
cyrus    ???          localhost        Thu Sep 23 16:08 - 16:08 (00:00)
reboot   system boot  2.4.26-std-up-al Thu Sep 23 16:03         (04:13)
```

В этом примере Мефодий неожиданно обнаружил кроме себя самого неизвестных ему пользователей `cacheman` и `cyrus` — он совершенно точно знает, что не создавал учётных записей с такими именами. Это **псевдопользователи** (или **системные пользователи**) — специальные учётные записи, которые используются для работы некоторыми программами. Поскольку эти «пользователи» регистрируются в системе без помощи монитора и клавиатуры, их «точка входа» в систему не определена (во второй колонке записано «`???`»). В выводе программы `last` появляется даже пользователь `reboot` (перезагрузка). В действительности такой учётной записи нет, программа `last` таким способом выводит информацию о том, когда была загружена система.

[8] В некоторых Linux-системах эта программа может называться `lastlog`.