

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ им. Н.Э. БАУМАНА

Факультет «Информатика и системы управления»

Кафедра «Автоматизированные системы обработки информации и  
управления»



**Сёмкин П.С., Сёмкин А.П.**

Методические указания по выполнению лабораторных работ  
по дисциплине  
«Операционные системы»

Лабораторная работа № 10  
«ОС Ubuntu. Управление пользователями»

**Москва**  
**2017 г.**

## ОГЛАВЛЕНИЕ

1	ЦЕЛЬ РАБОТЫ .....	3
2	ТЕОРЕТИЧЕСКАЯ ЧАСТЬ .....	3
2.1	Администратор в Ubuntu .....	3
2.1.1	Утилита sudo.....	3
2.1.2	Запуск графических программ с правами администратора.....	4
2.1.3	Запуск программ с правами администратора в терминале .....	5
2.1.4	Получение прав администратора для выполнения нескольких команд 5	
2.1.5	Использование root аккаунта и команды su .....	5
2.1.6	Настройка sudo и прав доступа на выполнение различных команд 6	
2.1.7	Конфигурационные файлы .....	6
2.1.8	Редактирование конфигурационных файлов .....	7
2.1.9	Критические системные файлы.....	8
2.2	Безопасность в операционных системах Unix.....	8
2.2.1	Концепция безопасности UNIX .....	8
2.2.2	Классическая концепция безопасности. ....	9
2.2.3	Назначение прав доступа .....	10
2.2.4	Структура файла /etc/passwd.....	11
2.2.5	Структура файла /etc/shadow.....	13
2.2.6	Пользовательские файлы конфигурации .....	15
2.2.7	Группы пользователей.....	16
2.3	Создание учётных записей пользователей и групп пользователей.....	17
2.3.1	Создание новой учётной записи пользователя в графической оболочке GUI .....	17
2.3.2	Создание новой учётной записи пользователя с помощью командной строки .....	18
2.3.3	Работа с группами пользователей .....	22
2.3.4	Наблюдение за пользователями .....	23
3	ЗАДАНИЕ НА ВЫПОЛНЕНИЕ РАБОТЫ .....	24
4	КОНТРОЛЬНЫЕ ВОПРОСЫ .....	24
5	ЛИТЕРАТУРА.....	24

## 1 Цель работы

Целью работы является знакомство с концепцией безопасности и политикой учётных записей пользователей и групп пользователей в операционных системах семейства Linux

Продолжительность работы – 2 часа

## 2 Теоретическая часть

### 2.1 Администратор в Ubuntu

В любой Linux системе обязательно есть один привилегированный пользователь - **root**. Этот пользователь имеет права на выполнение любых действий, удаление любых файлов и изменение любых параметров. Как-то ограничить свободу действий **root** практически невозможно. С другой стороны, все остальные пользователи системы обычно не имеют большинства необходимых прав, например, прав на установку программ, поскольку это является административной операцией, права на которую есть только у **root**. Ещё одной распространённой операцией, доступной только суперпользователю, является копирование и изменение файлов в системных папках, куда обычный пользователь доступа не имеет.

Раньше данная проблема решалась достаточно просто: при обладании паролем **root** можно было зайти в систему под его аккаунтом, либо временно получить его права, используя команду **sudo**. **В современных дистрибутивах Linux вместо root-аккаунта для администрирования используется утилита sudo.**

В Ubuntu по умолчанию **root**-аккаунт отключён, поэтому войти под учётной записью **root** , не включив её, нельзя. **root** отключён, т.е. он присутствует в системе, но чтобы его использовать необходимо включить **root** аккаунт.

#### 2.1.1 Утилита **sudo**

**sudo** - это утилита, предоставляющая привилегии **root** для выполнения административных операций в соответствии со своими настройками. Она

позволяет легко контролировать доступ к важным приложениям в системе. По умолчанию, при установке Ubuntu первому пользователю (тому, который создаётся во время установки) предоставляются полные права на использование `sudo`. Т.е. фактически первый пользователь обладает той же свободой действий, что и `root`. Однако такое поведение `sudo` легко изменить, об этом при настройке `sudo`.

`sudo` используется всегда, когда запускается функция администрирования системы.

Например, при запуске **Synaptic** - программы управления установленным ПО, для её запуска нужны права администратора, которые пользователь получает через `sudo`, вводя свой пароль.

### 2.1.2 Запуск графических программ с правами администратора

Для запуска графических программ с правами администратора можно воспользоваться диалогом запуска программ, вызываемым по умолчанию сочетанием клавиш `Alt+F2`.

Допустим, необходимо запустить файловый менеджер **Nautilus** с правами администратора, чтобы через графический интерфейс изменить содержимое системных папок. Для этого необходимо ввести в диалоге запуска приложений команду

**`gksudo nautilus`**

Вместо **`gksudo`** можно подставить **`gksu`**, кроме того, пользователи KDE должны вместо **`gksudo`** писать **`kdesu`**. Необходимо ввести пароль, и, если пользователь обладает нужными правами, **Nautilus** запустится от имени администратора. Запуск любого графического ПО можно производить с правами администратора, просто написав в диалоге запуска

`gksudo <имя_команды>`

**Надо быть предельно внимательным при работе в приложениях, запущенных с правами администратора, т.к. безо всяких предупреждений со стороны системы можно выполнить любую операцию, в частности, удалить системные файлы, сделав при этом систему неработоспособной.**

### 2.1.3 Запуск программ с правами администратора в терминале

Для запуска в терминале команды с правами администратора надо просто набрать перед ней `sudo`:

**`sudo <команда>`**

Необходимо ввести пароль. Пароль при вводе никак не отображается, это сделано в целях безопасности. После ввода пароля указанная команда исполнится от имени `root`.

Система какое-то время помнит введённый пароль (сохраняет открытой `sudo`-сессию). Поэтому при последующих выполнениях `sudo` ввод пароля может не потребоваться. Для гарантированного прекращения сессии `sudo` необходимо набрать в терминале

**`sudo -K`**

### 2.1.4 Получение прав администратора для выполнения нескольких команд

Иногда возникает необходимость выполнить подряд несколько команд с правами администратора. В этом случае можно временно перейти под суперпользователя командой

**`sudo -s`**

После этого произойдёт переход в режим суперпользователя (с ограничениями, наложенными через настройки `sudo`), о чём говорит символ `#` в конце приглашения командной строки. Данная команда по действию похожа на `su`, однако она не меняет домашний каталог на `/root`, что обычно очень удобно. Для того, чтобы при переходе под суперпользователя сменился также и домашний каталог необходимо выполнить

**`sudo -s -H`**

Для выхода обратно в режим обычного пользователя необходимо набрать `exit` или просто нажать **Ctrl+D**.

### 2.1.5 Использование root аккаунта и команды `su`

Отключение `root`-аккаунта в Ubuntu заключается в том, что у `root` просто не задан пароль. Поэтому для получения возможности использовать `root` логин необходимо установить для него пароль:

**sudo passwd root**

Кроме того, по умолчанию вход root в систему через GUI заблокирован, установка пароля для root это никак не исправит, для разблокирования необходимо перейти в меню **Система→Администрирование→Окно входа в систему**, потом на вкладку «Безопасность» и там поставить галочку напротив «Разрешить локальный вход администратора системы».

**Работа в GUI от имени администратора - это прямая дорога к многочисленным проблемам.**

Для отключения учётной записи root, надо просто удалить его пароль:

**sudo passwd -l root**

### 2.1.6 Настройка sudo и прав доступа на выполнение различных команд

sudo позволяет разрешать или запрещать пользователям выполнение конкретного набора программ. Все настройки, связанные с правами доступа, хранятся в файле /etc/sudoers. Это не совсем обычный файл. Для его редактирования необходимо (в целях безопасности) использовать команду

```
sudo visudo
```

По умолчанию, в нём написано, что все члены группы admin имеют полный доступ к sudo, о чём говорит строчка

```
%admin ALL=(ALL) ALL
```

Подробнее о синтаксисе и возможностях настройки этого файла можно почитать, выполнив команду

```
man sudoers
```

### 2.1.7 Конфигурационные файлы

Практически все настройки приложений в Ubuntu, включая системные компоненты, хранятся в виде обычных текстовых файлов различного формата, называемых конфигурационными файлами. Это очень удобно, поскольку позволяет просто читать и менять их не только из конкретного приложения. Программы обычно содержат встроенный редактор параметров, основные настройки системы так же можно легко изменить с помощью графических утилит, доступных из меню «Система». Однако некоторые операции

требуют редактирования системных файлов конфигурации, к которым нет доступа из графического окружения.

### 2.1.8 Редактирование конфигурационных файлов

Большинство конфигурационных файлов, которые приходится редактировать вручную, доступны для изменения только пользователю с привилегиями root. Если необходимо отредактировать подобный файл, то можно поступить несколькими способами:

Самый простой путь: вызвать диалог запуска программ (по умолчанию - Alt+F2) и запустить обычный текстовый редактор с правами суперпользователя командой:

**gksu gedit /путь к файлу/**

Пользователи Kubuntu должны подставить вместо *gedit* текстовый редактор KDE *kate*.

Необходимо ввести пароль и, если вы являетесь администратором компьютера, откроется для редактирования нужный файл.

То же самое можно сделать из терминала, запустив редактор командой

**sudo gedit /путь к файлу/**

В этом случае вместо графического окна с запросом пароля появится запрос непосредственно в терминале.

При введении пароля в терминале на экране ничего не отображается, ни звёздочек, ни чёрточек, ни каких-либо других символов.

Можно отредактировать текстовый файл непосредственно из терминала, не открывая графических приложений вообще. Существует масса текстовых редакторов для терминала, самыми популярными в среде линуксоидов являются *vi* и *emacs*, но для начала лучше всего использовать простой в освоении редактор *nano*, доступный по умолчанию в любой версии Ubuntu. Для открытия текстового файла с правами суперпользователя в *nano* надо просто набрать

**sudo nano путь к файлу**

Если для доступа к конфигурационному файлу требуются права администратора, то, скорее всего, в нём содержатся какие-то важные системные настройки. Надо быть предельно внимательным при редактировании таких файлов, ошибка может привести к неработоспособности всей системы. Если всё же случилось так, что вы записали в конфигурационный файл что-то не то, то всегда можно загрузиться с LiveCD и исправить любой файл.

Для редактирование некоторых конфигурационных файлов права администратора не требуются и поэтому являются излишними, в этом случае достаточно просто убрать `sudo` или `gksu` из начала команды и всё делать так же, как уже описано.

### 2.1.9 Критические системные файлы

Существует несколько критических конфигурационных файлов, от содержимого которых зависит в системе очень многое, классическим примером является файл `/etc/sudoers`. Для редактирования конкретно этого файла существует специально адаптированная версия редактора *vi*, которую можно вызвать командой

**`sudo visudo`**

Надо быть предельно внимательным при изменении подобных файлов, неправильная информация в `/etc/sudoers` может крайне просто привести к невозможности выполнить что-либо в системе.

## 2.2 Безопасность в операционных системах Unix

### 2.2.1 Концепция безопасности UNIX

Вся система безопасности UNIX изначально строилась на трех принципах:

- разделение всех пользователей по отношению к объекту на владельца объекта, группу объекта и всех остальных,
- назначение им прав доступа по отдельности



- обязательное наличие у каждого объекта владельца и группы.

В современных системах UNIX для большей гибкости прав доступа введены дополнительные свойства объектов, такие как

- флаги для файлов и каталогов,
- списки управления доступом (ACL) для файлов и каталогов,
- аутентификация и авторизация с использованием различных служб аутентификации подобных TACACS и RADIUS,
- модули аутентификации и авторизации (Pluggable Authentication Modules - PAM).

Естественно, надежность усовершенствованной системы безопасности снизилась, а сложность администрирования выросла, как и при любом другом усложнении любой системы. Чем более гибко настраивается система, тем внимательнее надо быть администратору при настройке.

### 2.2.2 Классическая концепция безопасности.

**Объект.** Объектом в контексте безопасности называется файл, каталог или процесс.

Файл и каталог хранятся на устройствах внешней памяти.

Процесс - выполнение некоторой программы.

**Разделение всех пользователей по отношению к объекту.**

У каждого объекта есть владелец. Это - один из пользователей данной системы UNIX. Право владения объектом в UNIX передается по наследству от процесса к процессу.

При создании файла или каталога его владельцем становится тот пользователь, от чьего имени запущен процесс, создающий файл или каталог.

В ряде случаев запускаемый процесс будет принадлежать не тому, кто запустил процесс-родитель, а иному пользователю.

- Например, при входе пользователя в систему он сообщает имя и пароль программе login. Она работает от имени root (т. е. ее владельцем является пользователь root). Но программа login запускает для пользователя

командный интерпретатор так, чтобы владельцем процесса командного интерпретатора был входящий в систему пользователь.

Любой объект имеет не только владельца, но и группу.

Иногда владельца называют «хозяином», а группу - «группой владельца», «групповым владельцем», «групповым хозяином» и т. п.

- Каждому объекту в UNIX сопоставляется не только UID (user identifier), который идентифицирует владельца объекта, но и GID (Group Identifier), который идентифицирует группу пользователей, имеющую особые права на объект.

Таким образом, реализуется разделение всех пользователей по отношению к объекту на

- владельца объекта,
- группу, имеющую особые права на объект,
- и всех остальных.

Группу, обладающую особыми правами на объект, называют *группой объекта*. Соответственно, говорят о

- группе файла,
- группе каталога
- или группе процесса.

При этом надо помнить, что группа файла - это не группа, в которую входит файл или его владелец. В общем случае это - произвольная группа, объединяющая произвольных пользователей и имеющая особые права на этот объект.

### 2.2.3 Назначение прав доступа

Владельцу файла, группе файла и всем остальным пользователям могут быть по отдельности назначены разные права доступа к файлу. То же справедливо и в отношении каталога. Так реализуется назначение прав доступа к объекту в отдельности его владельцу, группе и «всем остальным».

Каждый объект имеет владельца и группу. Любой файл, каталог или процесс имеет владельца и группу. Это означает, что файлу, каталогу и процессу обязательно сопоставлены два идентификатора, которые называются UID (User ID) и GID (Group ID) соответственно.

Администрировать систему легче, если все объекты имеют UID и GID из числа представленных в */etc/passwd* и */etc/group* соответственно. «Бесхозные» объекты вносят сумятицу в стройные ряды прав доступа и делают права доступа к ним владельца и группы бессмысленными, ведь никто из пользователей не может получить «чужое» право доступа.

Файлы */etc/passwd* и */etc/ggroup* в Ubuntu имеют такой же формат, как и в других системах UNIX

#### 2.2.4 Структура файла */etc/passwd*.

*cat /etc/passwd* - просмотр учетных записей пользователей

Файл состоит из записей, каждая из которых описывает одного пользователя и занимает одну строку. Поля записей разделяются двоеточиями.

- Первое поле - имя пользователя в системе. Это имя пользователь вводит для входа в систему. Имя должно иметь длину от 2 до 8 символов и содержать только латинские буквы и цифры. Имя пользователя может содержать прописные латинские буквы, однако из соображений совместимости с другими системами UNIX рекомендуется использовать только строчные (маленькие) буквы.
- Второе поле - признак наличия пароля. Пустое поле означает отсутствие пароля. Для фактического отсутствия пароля у пользователя необходимо, кроме того, чтобы второе поле в файле */etc/shadow* в описании этого пользователя имело значение NP.
- Третье поле - идентификатор пользователя, UID.
- Четвертое поле - идентификатор главной группы пользователя. GID.

- Пятое поле (иногда его называют GECOS) — описание пользователя. Обычно оно содержит полное имя пользователя (имя и фамилию) и координаты для связи с ним — номер офиса, адрес, телефон.
- Шестое поле — домашний каталог пользователя. При интерактивном входе в систему пользователь попадает именно в этот каталог сразу после успешного входа. Кроме этого, некоторые сетевые службы (например, ftpd) требуют, чтобы у каждого пользователя, пытающегося получить доступ к сетевой службе, был «честный», т. е. на самом деле существующий и доступный для пользователя домашний каталог.
- Седьмое поле - командный процессор, который будет запущен для пользователя при интерактивном входе в систему. Некоторые сетевые службы (например, ftpd) требуют, чтобы у каждого пользователя, пытающегося получить доступ к сетевой службе, был существующий в системе на самом деле командный процессор. Файл */etc/shells* описывает доступные в системе командные процессоры, которые следует назначать пользователям. Сразу после установки системы файла */etc/shells* не существует. Системный администратор должен создать его вручную, если он требуется для каких-то программ в системе, например для ftpd.

Поле GECOS часто называют полем комментария, и это верно: в нем следует записывать контактную информацию о пользователе. Чтобы точно знать, кто скрывается за лаконичным *username*, следует заполнять поле комментария. Каждый элемент этого поля (например, полное имя, номер комнаты в офисе, контактный телефон) принято отделять от других запятой, но это необязательное требование. В настоящее время, к сожалению, не существует строго определенного правила заполнения поля комментария. Хорошим тоном является указание в этом поле следующей информации:

- полные фамилия, имя и (если это необходимо для однозначной идентификации) отчество пользователя или, если это учетная запись предопределенного псевдопользователя типа `bin`, полное название приложения, использующего эту запись;

- номер комнаты и ее местоположение, или контактное лицо, ответственное за запуск приложения, использующего запись;

- рабочий телефон;

- другая контактная информация (пейджер, факс, мобильный телефон и т. п.).

Не всегда требуется указывать всю эту информацию. На деле достаточно иметь ровно столько сведений, сколько требуется системному администратору для однозначной идентификации владельца учетной записи (не одному-единственному конкретному системному администратору, а любому администратору, которому придется управлять созданной системой).

Делайте записи в поле комментария полезными сведениями, а не отписками.

- Последнее, что следует сказать о поле `GECOS`, - это почему оно так называется. В свое время компания General Electric владела компьютером, операционной системой которого была `GECOS` (General Electric Comprehensive Operating System). Компьютеры под управлением `UNIX` использовались для подготовки задач печати для этого компьютера. Изначальным назначением поля комментария было хранение идентификационной информации для задач, которые были предназначены для системы `GECOS`.

## 2.2.5 Структура файла `/etc/shadow`.

### **`cat /etc/shadow` - просмотр зашифрованных паролей**

Этот файл тоже описывает пользователей. В нем хранятся зашифрованные пароли пользователей. Формат файла таков:

- первое поле - имя пользователя;

- второе поле - зашифрованный пароль; \*LK\* означает, что учетная запись заблокирована (locked), а NP - что пароль отсутствует (no password);
- третье поле - число дней между 1 января 1970 года и датой послед него изменения пароля;
- четвертое поле - минимальное количество дней, которое должно пройти от одной смены пароля до другой;
- пятое поле - максимальное количество дней, которое пароль считается действительным; по истечении этого количества дней система попросит ввести новый пароль, так как старый утратит силу;
- шестое поле — количество дней, за которое система предупредит пользователя о необходимости смены пароля;
- седьмое поле - количество дней, которое пользователь может не работать в системе, но считаться активным; по истечении этого количества дней учетная запись автоматически блокируется;
- восьмое поле - дата, до которой учетная запись считается действительной: после этой даты пользователь не сможет войти в систему;
- девятое поле зарезервировано и сейчас не используется.

Данный формат `/etc/shadow` характерен для подобных System V систем UNIX и Linux.

В `/etc/shadow` обязательно присутствуют только первые три поля в каждой записи, остальные могут отсутствовать - все или часть из них.

Рекомендуется не вносить исправления в `/etc/shadow` вручную, а делать это с помощью команд **usermod**, **useradd**, **passwd**. Однако в одном случае -

если забыли пароль root — придется исправлять `/etc/shadow` в текстовом редакторе.

### 2.2.6 Пользовательские файлы конфигурации

У каждого пользователя в домашнем каталоге есть несколько файлов конфигурации.

Обычно присутствуют файлы конфигурации командного процессора.

Если в системе используется несколько командных процессоров, имеет смысл сделать такие файлы конфигурации для каждого из них.

Кроме них, могут быть файлы конфигурации

графической среды (`.Xsession` и другие),

файлы конфигурации почтовой системы (`.elm`, `.forward` и другие),

файлы с историей команд (`.history`, `.bash_history`) и прочие.

Их объединяет то, что их имена практически всегда начинаются с символа «.» (точка). Можно увидеть их в списке файлов каталога, если дать команду

**ls -a**

Пользовательские файлы конфигураций создаются заранее системным администратором.

Стандартные пользовательские файлы конфигураций по умолчанию поставляются вместе с операционной системой и в Ubuntu располагаются в `/etc/skel` (от слова *skeleton* — скелет, т. е. основа). При создании нового пользователя они автоматически копируются из каталога `/etc/skel` в домашний каталог нового пользователя.

При создании нового пользователя или модификации существующей учетной записи можно указать другой каталог с файлами конфигурации, чтобы копировать не файлы по умолчанию, а другие файлы. Их предварительно следует создать и модифицировать в соответствии с желаемыми настройками для новых пользователей.

Модификация файлов в каталоге `/etc/skel` повлияет только на настройки новых пользователей, которые будут созданы после модификации этих файлов.

`/etc/skel` называется каталогом базовых пользовательских файлов конфигурации. После того как пользователь войдет в систему, он может изменить настройки, сделанные для него в файлах конфигурации, или добавить новые, если системный администратор не запретил ему запись в файлы конфигурации.

### 2.2.7 Группы пользователей

Группы, определенные в системе, перечислены в файле `/etc/group`. В системе могут быть определены и другие группы, если для аутентификации и авторизации помимо файлов `/etc/passwd` и `/etc/group` используются и другие источники (например каталог LDAP ).

#### **cat /etc/ group - просмотр групп**

- Первое поле — имя группы.
- Второе поле — зашифрованный пароль (это устаревшее поле - в настоящее время нет команды, которая бы позволила установить пароль на группу, и, обычно, нет необходимости это делать). Если все же такая необходимость появится, то можно установить требуемый пароль какому-нибудь пользователю с помощью программы `passwd`. а затем копировать поле пароля из `/etc/shadow` в `/etc/group`. Пароль группы используется в Solaris только программой `newgrp`. Эта программа требуется для изменения эффективного группового идентификатора пользователя в ходе его интерактивной работы. Если группа, которой соответствует новый групповой идентификатор, имеет пароль, то программа `newgrp` его запросит.
- Третье поле - идентификатор группы (GID). Этот идентификатор должен быть уникальным в пределах системы, а в случае использования общих файлов групп и паролей - в пределах



всей сети организации. Номера от 0 до 99 и от 60001 до 60002 зарезервированы для системных групп. Создавайте свои группы с идентификаторами от 100 до 60000 включительно.

- Четвертое поле - список пользователей через запятую; для этих пользователей данная группа будет являться дополнительной. В Solaris принято по умолчанию, что один пользователь может принадлежать не более чем к 15 дополнительным группам.


## 2.3 Создание учётных записей пользователей и групп пользователей

### 2.3.1 Создание новой учётной записи пользователя в графической оболочке GUI

1. **Открыть** на панели меню **Параметры системы**.

2. **Открыть** Учётные записи.

3. Для добавления учётных записей пользователей необходимы привилегии администратора. **Щёлкнуть** Разблокировать в верхнем правом углу и ввести пароль.

4. Слева, в списке учётных записей, **щёлкнуть** кнопку  для добавления новой учётной записи пользователя.

5. Если нужно, чтобы новый пользователь обладал **доступом с правами администратора**, выбрать из выпадающего меню тип учётной записи **Администратор**. Администраторы могут совершать действия, такие как добавление и удаление пользователей, установка программ и драйверов, а также изменять дату и время.

6. Ввести полное имя нового пользователя. Имя пользователя будет заполнено автоматически на основе полного имени. Имя пользователя можно изменить его при желании.

7. **Щёлкнуть** **Создать**.

8. Первоначально учётная запись отключена до тех пор, пока не выбрано

что делать с паролем пользователя. Под надписью **Параметры входа в систему** щёлкнуть **Учётная запись отключена** рядом с надписью **Пароль**. Выбрать из выпадающего списка действий **Установите пароль сейчас**. Ввести свой пароль в поля **Новый пароль** и **Подтвердить пароль**.

9. Можно также щёлкнуть кнопку, расположенную рядом с полем **Новый пароль** для выбора произвольно сгенерированного надёжного пароля. Такие пароли сложно подобрать, но и сложно запомнить, так что надо быть внимательным.

#### 10. Щёлкнуть **Изменить**.

В правой части окна Учётных записей можно щёлкнуть изображение рядом с именем пользователя для выбора изображения для учётной записи. Это изображение будет отображаться в окне входа. GNOME предоставляет для использования галерею фотографий, но можно выбрать и свою собственную или сфотографироваться с помощью веб-камеры.

### Изменение пароля

1. Открыть на панели меню **Параметры системы**.
2. Открыть Учётные записи.
3. Щёлкнуть табличку рядом с надписью **Пароль**.

Эта табличка выглядит как ряд точек или квадратиков, если пароль уже есть.

4. Ввести действующий пароль, а затем новый пароль. Ввести новый пароль снова в поле **Подтвердить пароль**.

5. Можно также щёлкнуть кнопку, расположенную рядом с полем **Новый пароль** для выбора произвольно сгенерированного надёжного пароля.

6. Щёлкнуть **Изменить**.

### 2.3.2 Создание новой учётной записи пользователя с помощью командной строки

Для управления учетными записями используются команды **useradd**, **usermod**, **userdel**, в окне терминала.

Любая программа для работы со свойствами учетной записи пользователя в графическом режиме все равно вызывает простые системные программы с интерфейсом командной строки для выполнения любых операций с этими свойствами.

- для создания новой учетной записи пользователя (добавления пользователя в систему) используется команда **useradd**,
- для модификации учетной записи пользователя **-usermod**,
- для удаления учетной записи — **userdel**.

### Создание нового пользователя

1. Открыть окно терминала командой **Ctrl+Alt+T**
2. Выполнить команду **useradd**

При добавлении пользователя с помощью команды **useradd** пользователь попадает в группу **other**, если явно не указано иное. Эта группа имеет идентификатор 1.

- При добавлении пользователя происходит добавление соответствующей строки в файлы **/etc/passwd**, **/etc/shadow**
- При указании ключа **-G** запись о пользователе помещается в **/etc/group** в строку тех групп, которые указываются в качестве дополнительных для него.
- С помощью ключа **-m** можно указать, что для пользователя следует создать домашний каталог.

Новые учетные записи пользователей остаются заблокированными до тех пор, пока пользователю не будет назначен пароль с помощью программы **passwd**.

Имя пользователя или роли не может быть длиннее 8 символов, и должно содержать только латинские символы, цифры, точку, знак подчеркивания и дефис. Точка является допустимым символом не для всех систем UNIX, поэтому ее не рекомендуется использовать в целях совместимости, хотя это требование не является жестким. Первый символ

имени должен быть буквой, по крайней мере одна буква в имени должна быть буквой нижнего регистра.

Следующие ключи изменяют значения свойств учетной записи пользователя, которые задаются по умолчанию в отсутствие этих ключей:

**-b base\_dir**

каталог, в котором должен быть создан домашний каталог пользователя; не требуется указывать, если используется ключ -d;

**-c comment**

поле GECOS общей информации о пользователе; как минимум, следует указать полное имя пользователя;

**-d dir**

домашний каталог пользователя; по умолчанию создается в каталоге /home и носит то же имя, что и пользователь (для пользователя student создается домашний каталог /home/student);

**-D**

показать значения по умолчанию для группы, системного каталога для домашних каталогов, каталога базовых пользовательских файлов конфигурации, командного процессора и ряда других свойств.

Значения свойств учетной записи пользователя по умолчанию, принятые в системе, перечислены в таблице .

Группа	other (GID=100)
Каталог с домашними каталогами пользователей	/home
Командный процессор	/bin/sh
Профиль	не назначается
Роль	не назначается

**-e expire**

дата истечения срока действия учетной записи; формат даты определен, умолчанию не задано, после наступления этой даты пользователь не сможет

войти в систему; используется для создания учетных записей временных сотрудников или гостей;

**-f inactive**

максимальное число дней, в течение которых пользователь может не входить в систему; по истечении этого срока учетная запись блокируется; удачное решение для забывчивых системных администраторов, которые не удаляют учетные записи при увольнении сотрудника, позволяет бороться с накоплением «мертвых душ в системе»;

**-g group**

главная группа пользователя, по умолчанию — other;

**-k skel\_dir**

каталог с базовыми пользовательскими файлами конфигураций; указанный каталог должен существовать, копии всех содержащихся в нем файлов будут помещены в домашний каталог нового пользователя. По умолчанию — /etc/skel;

**-m**

требуется создать домашний каталог нового пользователя; если каталог уже есть, у главной группы пользователя должно быть право на чтение, запись и поиск в этом каталоге

**-o**

разрешить пользователю иметь uid, совпадающий с uid существующего пользователя; может использоваться для дублирования учетной записи root. Это может понадобиться для того, чтобы от имени пользователя root входить в систему только в экстренных случаях, а обычно пользоваться другой учетной записью. Во FreeBSD для этого по умолчанию предоставляется учетная запись toor (root наоборот). Это может быть удобно, например, для назначения администратору командного процессора, отличного от /bin/sh, например, /usr/local/bin/bash. Назначить такой командный процессор пользователю root нельзя, т. к. его командный процессор должен работать даже тогда, когда доступна только файловая система /, а /usr г даже не смонтирована;

### **-s shell**

полное имя файла, который будет назначен пользователю в качестве командного процессора при входе в систему; файл должен существовать;

### **-u uid**

явное указание значения UID.

Для изменения свойств пользователя следует запускать программу **usermod**.

Ее синтаксис предполагает явное задание изменяемого свойства. При запуске без ключей только с указанием имени пользователя, чью учетную запись надо изменить, программа **usermod** выдает сообщение об ошибке и краткую подсказку по использованию:

## **2.3.3 Работа с группами пользователей**

Каждый новый пользователь приписывается к одной или нескольким группам. Можно создавать группы в любое время и добавлять в них пользователей. Права доступа к файлам и каталогам, которые получает каждая группа в Linux, зависят от того, какие биты прав доступа заданы для каждого элемента. Приписка пользователей к группе позволяет назначить владение файлами, каталогами и приложениями, чтобы эти пользователи смогли работать вместе над некоторым проектом или имели общий доступ к ресурсам.

Для управления группами доступны команды, подобные применяемым для работы с пользователями. Можно добавлять группы (**groupadd**), изменять настройки групп (**groupmod**), удалять группы (**groupdel**), а также добавлять и удалять членов этих групп (**groupmems**).

Используя команду **groupmod**, можно изменить имя или идентификатор существующей группы.

Удаление группы или пользователя не приводит к удалению файлов, каталогов, устройств или других элементов, которыми владеет эта группа или пользователь. При выводе на экран длинного списка (**ls -l**) файлов или каталогов, которыми владели пользователь или группа, подвергнутые

удалению, отобразится идентификатор удаленных пользователя или группы.

### 2.3.4 Наблюдение за пользователями

Создав учетные записи пользователей и предоставив этим пользователям доступ к ресурсам системы, можно использовать команды для отслеживания, как они используют ресурсы.

Команду **find** можно использовать для повсеместного поиска в системе файлов, которыми владеют выбранные пользователи;

Команду **du** можно использовать, чтобы узнать, сколько дискового пространства занимают домашние каталоги выбранных пользователей;

Команды **fuser**, **ps** и **top** можно использовать для выяснения того, какие процессы запущены пользователями.

С помощью команды **last** можно увидеть информацию о каждом пользователе, который вошел в систему (или открыл новый интерпретатор команд), а также узнать, как долго он находился в системе или все еще находится в ней (*still logged in*). В строках терминалов *tty1* и *tty3* показаны пользователи, работающие из виртуальных терминалов в консоли. В строках *pts* указаны имена пользователей, открывших интерпретатор команд с удаленного компьютера (*thompson*) или локального экрана X (*:0.0*). Рекомендуется использовать параметр *-a* для обеспечения удобочитаемости.

Команда **lastb** показывает неудачные попытки входа в систему, а также откуда они исходили.

Команды **who -u** и **users** отображают информацию о пользователях, находящихся в системе на данный момент.

Помимо отображения основной информации о пользователе (логин, имя, домашний каталог, интерпретатор команд и т. д.), команда **finger** также выведет на экран любую информацию, хранящуюся в специальных файлах в домашнем каталоге этого пользователя.

### 3 Задание на выполнение работы

1. Войти в систему под учётной записью **studXX**(XX –индекс группы).
2. Запустить программу Oracle VM VirtualBox.
3. Запустить виртуальную машину Ubuntu.
4. Создать в группе пользователей **stud3k** учетные записи пользователей **Stud51, Stud52, Stud53, Stud54**

### 4 Контрольные вопросы

1. Назовите принципы безопасности Unix.
2. Как назначаются права доступа к объектам Unix?
3. В чём назначение утилиты **sudo** ?
4. Как создаются учетные записи пользователей и групп пользователей?
5. Как можно включить пользователя в состав группы?

### 5 ЛИТЕРАТУРА

1. Робачевский А.М. Операционная система UNIX.-СПб.: БХВ-Петербург, 2001. – 528 с.:ил.
2. Сергей Ивановский Операционная система Linux. М.: Познавательная книга плюс, 2001. – 512 с.
3. Негус К. Ubuntu и Debian Linux для продвинутых. 2-е изд. – СПб.: Питер, 2014. -384 с.: ил.