

---

# Управление пользователями и группами

В Linux есть два типа пользователей: пользователи-люди и системные пользователи. Каждый из них имеет уникальный идентификатор (User ID, UID) и по крайней мере один идентификатор группы (Group ID, GID). Каждый пользователь входит в одну основную группу и может входить в несколько дополнительных.

Каждый пользователь-человек владеет домашним каталогом со своими личными файлами. Домашние каталоги пользователей находятся в `/home`, и их названия совпадают с именами владельцев, как в нашем примере пользователя `Duchess`, которому принадлежит каталог `/home/duchess`. Помимо основной группы, пользователи могут входить в несколько других, которые называются *дополнительными*. Пользователи в группе получают все ее привилегии. (Чтобы узнать больше о привилегиях, см. главу 6.) Привилегии управляют доступом к файлам и командам и являются основой безопасности системы.

Системные пользователи — это системные службы и процессы. Учетные записи системных пользователей нужны лишь для управления привилегиями, и у них нет паролей и каталогов в `/home`.

Пользователи-люди делятся на две категории: пользователь `root`, или супер-пользователь, обладает неограниченными привилегиями и может делать в системе все что угодно. Все остальные пользователи называются обычными, или непривилегированными. Обычным дается достаточно прав для управления их файлами и выполнения команд, которые разрешается использовать этой категории людей. Обычным пользователям могут быть предоставлены ограниченные или полные привилегии `root`, о которых вы узнаете из рецептов, описывающих команды `su` и `sudo`.

Увидеть список всех пользователей в системе можно в файле `/etc/passwd`, а все группы — в `/etc/group`.



### Централизованное управление пользователями

Файлы `/etc/passwd` и `/etc/group` достались в наследство от Unix и практически не изменились с тех пор, как переключались в Linux в 1992 году. С тех пор появились новые инструменты для управления пользователями и группами, например централизованные базы данных, обслуживающие целые организации. В этой главе мы не будем рассматривать инструменты централизованного управления пользователями.

В Linux есть несколько команд для управления пользователями и группами:

- `useradd` — создает новых пользователей;
- `groupadd` — создает новые группы;
- `userdel` — удаляет пользователей;
- `groupdel` — удаляет группы;
- `usermod` — изменяет настройки существующего пользователя;
- `passwd` — создает и изменяет пароли.

Они являются частью набора *Shadow Password Suite*, и основным конфигурационным файлом для них служит `/etc/login.defs`.

Команда `useradd` действует по-разному в разных системах, в зависимости от настроек. Традиционно эта команда объединяла всех новых пользователей в одну и ту же основную группу `users` (100). Это означало, что пользователи должны были проявлять осторожность при выборе разрешений для своих файлов, чтобы случайно не раскрыть свои секреты другим членам группы. В Red Hat изменили данную ситуацию, разработав схему *User Private Group*, согласно которой для каждого нового пользователя создается личная основная группа. Большинство дистрибутивов Linux используют эту схему по умолчанию, хотя есть исключения, такие как openSUSE.

Набор команд *Shadow Password Suite* был создан Джулианной Фрэнсис Хо (Julianne Frances Haugh) в 1980-х годах, еще до появления Linux, для повышения безопасности паролей Unix и упрощения управления учетными записями пользователей. В 1992-м этот набор был перенесен в систему Linux, когда ей едва исполнился год.

До появления *Shadow Password Suite* все файлы, имеющие отношение к учетным записям пользователей, приходилось редактировать по отдельности, имелось несколько команд управления паролями, а хешированные пароли хранились в файлах `/etc/passwd` и `/etc/group`. Но, поскольку `/etc/passwd` должен оставаться доступным для чтения всем пользователям, хранение паролей в нем, пусть и в зашифрованном виде, чревато неприятностями. Скопировав этот файл, любой желающий теоретически сможет взломать пароли. Перемещение хешированных паролей в файлы `/etc/shadow` и `/etc/gshadow`, доступные только пользователю

root, дополнило защиту. Долгожительство Shadow Password Suite свидетельствует о том, насколько хорошо был проработан и реализован данный пакет.

Относительно недавно в Debian появились *adduser* и *addgroup*. Это сценарии-обертки на Perl для команд *useradd* и *groupadd*. Они по шагам проведут вас через процесс создания нового пользователя и новой группы.

В этой главе вы узнаете, как создавать и удалять обычных и системных пользователей, управлять паролями, определять идентификаторы UID и GID, устанавливать желаемые значения по умолчанию настроек для создания новых пользователей, изменять принадлежность к группам, настраивать общие файлы для новых пользователей, очищать каталоги после удаления пользователей, получать привилегии root и предоставлять ограниченные полномочия root обычным пользователям.

## 5.1. Определение UID и GID пользователя

### Задача

Определить UID и GID пользователя.

### Решение

Выполнив команду *id* без параметров, можно узнать собственные UID и GID. Ниже представлен пример определения идентификаторов пользователя Duchess:

```
duchess@pc:~$ id
uid=1000(duchess) gid=1000(duchess)
groups=1000(duchess),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),118(lpadmin),
126(sambashare),131(libvirt)
```

Чтобы узнать UID и GID другого пользователя, нужно передать команде *id* его имя:

```
duchess@pc:~$ id madmax
uid=1001(madmax) gid=1001(madmax) groups=1001(madmax),1010(composers)
```

С помощью *id* можно также узнать свой эффективный идентификатор пользователя, когда вы выступаете от имени другого пользователя. Ниже представлен пример, в котором задействована с команда *sudo*:

```
duchess@client4:~$ sudo id -un
root
```

```
duchess@client4:~$ sudo -u madmax id -gn
madmax
```

## Комментарий

В Linux есть три типа идентификаторов пользователя/группы:

- реальный UID/GID;
- эффективный UID/GID;
- сохраненный UID/GID.

*Реальный идентификатор* — это UID и GID основной группы, присвоенные пользователю при создании. Это то, что вы видите, когда запускаете команду `id` без параметров от своего имени.

*Эффективный идентификатор* — это UID, используемый для запуска процесса, которому требуются привилегии, отличные от привилегий пользователя, запускающего процесс. Примером может служить команда `passwd`, которая требует привилегий суперпользователя и применяет специальные режимы разрешений, чтобы дать пользователям возможность изменять свои собственные пароли.

Вы можете убедиться в этом сами. Во-первых, взгляните на разрешения команды `passwd`:

```
$ ls -l /usr/bin/passwd
-rwsr-xr-x 1 root root 68208 May 27 2020 /usr/bin/passwd
```

Как видите, файл `passwd` принадлежит пользователю и группе `root`. Теперь введите команду `passwd` и нажмите `Enter`.

Откройте второй терминал, найдите идентификатор процесса `passwd`, а затем по этому идентификатору определите эффективный и реальный идентификатор пользователя и эффективный идентификатор группы:

```
$ ps -a|grep passwd
12916 pts/1 00:00:00 passwd

$ ps -eo pid,euser,ruser,rgroup | grep 12916
12916 root root root
```

Несмотря на то что команда `passwd` была запущена непривилегированным пользователем, она работает с правами `root`. (См. рецепт 6.11, в котором рассказывается о режимах специальных разрешений.)

*Сохраненный идентификатор* используется процессами, которым требуются повышенные привилегии, обычно привилегии `root`. Когда для выполнения работы достаточно привилегий обычного пользователя, он может временно переключиться на непривилегированный идентификатор пользователя. Эффективным становится UID с пониженными привилегиями, а исходный эффективный UID

сохраняется в SUID — Saved User ID (сохраненный идентификатор пользователя). Когда процессу снова требуются повышенные привилегии, он назначает эффективным сохраненный SUID.

Команда `id` поддерживает несколько параметров:

- `-u` выводит эффективный числовой UID;
- `-g` выводит эффективный числовой GID;
- `-G` выводит все числовые идентификаторы групп;
- `-n` выводит имя пользователя вместо числового идентификатора UID. Этот параметр можно использовать в комбинации с `-u`, `-g` и `-G`;
- `-un` выводит эффективный числовой UID и имя пользователя;
- `-gn` выводит имя эффективной группы;
- `-Gn` выводит все имена эффективных групп;
- `-r` выводит реальный числовой идентификатор вместо эффективного. Этот параметр можно использовать в комбинации с `-u`, `-g` и `-G`.

## Дополнительная информация

- Рецепт 6.11.
- `man 1 id`
- `man 1 ps`

## 5.2. Создание учетной записи для пользователя-человека с помощью команды `useradd`

### Задача

Создать новую учетную запись пользователя с личной группой и домашний каталог с набором файлов по умолчанию, таких как `.bashrc`, `.profile`, `.bash_history`, и любыми другими необходимыми файлами.

### Решение

В большинстве дистрибутивов Linux для этой цели имеется команда `useradd`, которую можно настроить под свои требования. Конфигурация по умолчанию

различается в разных дистрибутивах, поэтому самый простой способ узнать используемые настройки — создать новую пробную учетную запись:

```
$ sudo useradd test1
```

После этого запустите команду `id` и проверьте — был ли создан домашний каталог. Следующий пример был получен в Fedora 34:

```
$ id test1
uid=1011(test1) gid=1011(test1) groups=1011(test1)

$ sudo ls -a /home/test1/
.  ..  .bash_logout  .bash_profile  .bashrc
```

В этом примере конфигурация по умолчанию соответствует требованиям, перечисленным выше в подразделе «Задача». Теперь осталось только установить пароль:

```
$ sudo passwd test1
Changing password for user test1.
New password: пароль
Retype new password: пароль
passwd: all authentication tokens updated successfully.
```

При необходимости можно заставить пользователя сменить пароль при первом входе в систему после того, как вы создали пароль:

```
$ sudo passwd -e test1
Expiring password for user test1.
passwd: Success
```

Сообщите данные для входа вашему пользователю, и он начнет использовать свою учетную запись. Новая учетная запись в файле `/etc/passwd` представлена следующим образом:

```
test1:x:1011:1011::/home/test1:/bin/bash
```

В некоторых дистрибутивах, например в openSUSE, команда `useradd` настроена так, что по умолчанию не создает домашний каталог пользователя и включает всех пользователей в группу `users (100)`. Вследствие этого другие пользователи смогут получить доступ к файлам друг друга, если разрешения файлов для группы позволяют это. Следующий пример создает личную группу пользователя:

```
$ sudo useradd -mU test2
```

Параметр `-m` позволит создать домашний каталог пользователя, а параметр `-U` — личную группу с именем, совпадающим с именем пользователя.

## Комментарий

Все новые учетные записи пользователей остаются неактивными до установки пароля.

Первая группа, в которую добавляется пользователь, будь то его личная группа или общая группа для всех пользователей, становится его *основной* группой. Все остальные группы, в которые включается пользователь, считаются *дополнительными*.

Ниже представлено еще несколько полезных параметров:

- `-G, --groups` — добавляет пользователя в несколько дополнительных групп, перечисленных через запятую. Группы должны существовать к моменту выполнения команды:

```
$ sudo useradd -G group1,group2,group3 test1
```

- `-c, --comment` — принимает любую текстовую строку и сохраняет ее как полное имя пользователя, комментарий или описание:

```
$ useradd -G group1,group2,group3 -c 'Test 1, , , ' test1
```

Четыре запятых в данном примере определяют пять полей: имя, номер кабинета, рабочий телефон, домашний телефон и прочее (произвольная информация). В прошлом эти поля назывались данными GECOS, где GECOS (General Electric Comprehensive Operating Supervisor) — название операционной системы для мейнфрейма. Вы можете ввести в эти поля любую информацию по своему усмотрению или оставить их пустыми, хотя иногда имеет смысл указать полное имя пользователя. Изучите свой файл `/etc/passwd` и посмотрите, как другие учетные записи задействуют поля GECOS.

Настройки по умолчанию для команды `useradd` разбросаны по нескольким конфигурационным файлам; см. рецепт 5.4, чтобы узнать, как их изменить.

## Дополнительная информация

- `man 8 useradd`
- `man 5 login.defs`
- `/etc/default/useradd`
- `/etc/skel`
- `/etc/login/defs`

## 5.3. Создание системной учетной записи с помощью команды `useradd`

### Задача

Создать системного пользователя с помощью команды `useradd`.

### Решение

Следующий пример создаст нового системного пользователя без домашнего каталога, без оболочки входа и с UID из диапазона, предназначенного для системных пользователей:

```
$ sudo useradd -rs /bin/false service1
```

Параметр `-r` создает системного пользователя с реальным UID из диапазона, предназначенного для системных пользователей, а параметр `-s` задает оболочку входа `/bin/false` — команду, которая ничего не делает и не позволяет выполнить вход в систему с именем этого пользователя.

Дополнительную информацию о диапазонах UID и GID вы найдете в подразделе «Комментарий» рецепта 5.6.

### Комментарий

Раньше большинство служб выполнялись с привилегиями пользователя `nobody`. В настоящее время общепринято создавать для служб своих отдельных пользователей, так как это обеспечивает более высокий уровень безопасности, чем применение одного пользователя `nobody`, владеющего несколькими службами. Вам редко придется создавать учетные записи системных пользователей, поскольку службы делают это автоматически при установке.

Пользователь `nobody` всегда получает UID 65534 и GID 65534.

### Дополнительная информация

- `man 8 useradd`
- `man 1 false`
- Подраздел «Комментарий» в рецепте 5.6.

## 5.4. Изменение настроек по умолчанию для команды `useradd`

### Задача

Настройки по умолчанию для команды `useradd` вам не подходят, и их нужно изменить.

### Решение

Настройки команды `useradd` разбросаны по множеству конфигурационных файлов, таких как `/etc/default/useradd`, `/etc/login.defs` и файлы в каталоге `/etc/skel`.

В файле `/etc/default/useradd` находятся следующие настройки. Этот пример взят из `openSUSE`:

```
$ useradd -D
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/bash
SKEL=/etc/skel
CREATE_MAIL_SPOOL=yes
```

`GROUP=100` назначает единую группу с идентификатором 100 как основную всем новым пользователям. Чтобы обеспечить включение всех новых пользователей в общую группу, нужно создать эту группу, выключить параметр `USERGROUPS_ENAB` в `/etc/login.defs` и присвоить `GID` общей группы параметру `GROUP=` в `/etc/default/useradd`. Если, например, допустить, что наш пользователь `Duchess` включен в общую группу, то команда `id` выведет для него `uid=1000(duchess) gid=100(users)`.

Чтобы обеспечить создание личных групп для всех новых пользователей, нужно присвоить параметру `USERGROUPS_ENAB` в `/etc/login.defs` значение `yes` и закомментировать параметр `GROUP=` в `/etc/default/useradd`. Если, например, допустить, что наш пользователь `Duchess` имеет личную группу, то команда `id` выведет для него `uid=1000(duchess) gid=1000(duchess)`.

- `HOME=` задает каталог по умолчанию для размещения домашних каталогов пользователей. По умолчанию `/home`.

- `INACTIVE=-1` задает срок действия пароля в днях, по истечении которого учетная запись будет заблокирована. Значение `0` сразу же отключает учетную запись, так как срок действия пароля истекает немедленно, а значение `-1` запрещает блокирование учетных записей.
- `EXPIRE=` назначает конечную дату действия учетной записи в формате `YYYY-MM-DD`. Например, если установить значение `2021-12-31`, то учетная запись будет заблокирована в эту дату. Если параметр `EXPIRE=` оставить пустым, то это будет означать отсутствие конечной даты.
- `SHELL=/bin/bash` назначает командную оболочку по умолчанию. Наиболее широко используется оболочка `/bin/bash`. Вдобавок в этом параметре можно назначить любую другую командную оболочку, установленную в системе, например: `/bin/zsh` или `/usr/bin/tcsh`. Получить список установленных командных оболочек можно с помощью команды `cat /etc/shells`.
- `SKEL=/etc/skel` определяет каталог с файлами, которые должны автоматически копироваться в домашние каталоги новых пользователей. В большинстве дистрибутивов Linux такие файлы помещаются в `/etc/skel`. К ним относятся: `.bash_logout`, `.bash_profile`, `.profile`, `.bashrc` и любые другие файлы, которые должны иметься у новых пользователей. Вы можете отредактировать эти файлы в соответствии со своими требованиями. `SKEL` — это сокращение от `skeleton` («каркас, основа»).
- `CREATE_MAIL_SPOOL=yes` — пережиток прошлого, и этому параметру всегда следует присваивать значение `yes`, так как некоторые устаревшие процессы могут все еще нуждаться в нем.

Ниже представлены наиболее актуальные параметры со значениями по умолчанию в `/etc/login.defs`:

- `USERGROUPS_ENAB yes` включает создание личной группы для каждого пользователя;
- `CREATE_HOME yes` требует от `useradd` автоматически создавать домашние каталоги для новых пользователей. Не применяется к системным пользователям (см. рецепт 5.3).

## Комментарий

Диапазоны UID определены в `/etc/login.defs`. Каждый UID должен быть уникальным, поэтому команды создания учетных записей пользователей назначают UID из диапазона, определенного в данном файле. Обычно диапазон UID для учетных записей людей начинается с `1000` и автоматически используется