

# Занятие 1. Базовые понятия и методы работы в ОС Linux

## 1.1. Введение.

### 1.1.1. Разновидности операционных систем семейства Unix.

В 1969 году Кен Томпсон и Денис Ритчи, работники корпорации AT&T, создали небольшую операционную систему для компьютера PDP-7. Эта операционная система получила название Unix. Однако в планы компании AT&T не входило распространение этой операционной системы, и она предоставила ее за символическую плату учебным заведениям США, не организовав при этом службы технического сопровождения, исправления ошибок и вообще не дав никаких гарантий.

Вследствие этого пользователи, почти все являвшиеся представителями университетских вычислительных центров, были вынуждены сотрудничать друг с другом. Они сами устраняли ошибки, создавали полезные программы и утилиты и совместно их использовали. Результатом их работы стала целая серия версий Unix, распространяемых под эгидой компании Bell Labs вплоть до 1990 года (Последней версией была Unix System V Release 4 – SVR4).

Одна из групп пользователей Unix находилась в калифорнийском университете в Беркли. В 1977 году специалисты этого учебного заведения сделали следующий шаг в истории Unix и приступили к распространению магнитных лент с операционной системой 2BSD (Berkeley Software Distribution). С тех пор было продано 75 копий.

На основании Unix SVR4 и BSD были созданы все современные разновидности Unix.

Существует множество разновидностей Unix и Unix-подобных систем. К наиболее известным из них относятся Solaris (ранее SunOS) корпорации SUN Microsystems, AIX компании IBM, DEC Unix фирмы DEC, SCO UnixWare и прочие. Все вышеназванные системы являются коммерческими, и многие из них имеют высокую цену. Они работают на различных архитектурах (Intel, Sparc, Alpha, PowerPC и т.д.). Однако наибольший интерес сегодня в мире Unix приобрели операционные системы, построенные на модели открытого кода, такие как Linux.

### 1.1.2. История создания и развития Linux.

Linux изначально была разработана как свободно распространяемая версия Unix. В 1991 году студент Хельсинского университета Линус Торвалдс выпустил первую версию Linux. Она была основана на операционной системе Minix – ограниченном аналоге Unix для ПК. После выпуска первого "почти безошибочного" релиза в марте 1992 года, многие программисты мира подключились к разработке этой операционной системы, и она стала расти.

### 1.1.3. Основные дистрибутивы Linux. Различия и особенности.

#### Slackware Linux

Старейший из существующих дистрибутивов. Создан Патриком Фолкердингом в 1992 году как ветка проекта SLS, в настоящее время уже завершенного. Его доля достигала 80% от всех инсталляций Линукс в 1995 году. До настоящего времени сохраняет приверженность базовым принципам UNIX.

Единственное средство конфигурирования - редактирование текстовых файлов.

Использует простую систему управления пакетами pkgtool, не позволяющую устанавливать зависимости.

Считается наиболее стабильным и надежным дистрибутивом Линукс.

Поддерживает ограниченное число приложений. Подход к добавлению базовых пакетов консервативный. Процедура обновления сложна.

В настоящее время в основном выступает в качестве базовой системы для многочисленных кастомных решений.

#### Red Hat/CentOS/Fedora Linux

Ведущий дистрибутив Линукс, создан в 1994 году. Развивается компанией Red Hat, которая предлагает платную поддержку и услуги по интеграции.

Широко используется как в качестве серверной, так и десктопной ОС.

Использует систему управления пакетами RPM, впервые представленную в этом дистрибутиве в 1997 году. Система поддерживает структуру зависимостей.

В 2004 году, после выпуска Red Hat 9, компания Red Hat радикально изменила продуктовую линейку, выделив в отдельный проект дистрибутив Fedora, бесплатный и ориентированный на персональных пользователей. Основной дистрибутив стал коммерческим, изменив название на Red Hat Enterprise Linux. Примерно в это же время создан дистрибутив CentOS, который фактически представляет собой свободно распространяемый клон RHEL.

В настоящее время все три дистрибутива развиваются параллельно в достаточно тесном контакте друг с другом, под общей координацией компании Red Hat. Несомненно, Fedora с полугодовым циклом выхода основных версий является наиболее передовым и содержит максимальное количество нововведений. RHEL служит надежным, хорошо протестированным и поддерживаемым решением для корпоративных пользователей. CentOS также стабилен и хорошо тестируется, но развивается медленнее и уступает многим другим дистрибутивам по скорости внедрения нововведений, в том числе поддержки нового аппаратного обеспечения.

Pros: Highly innovative; outstanding security features; large number of supported packages; strict adherence to the free software

philosophy; availability of live CDs featuring many popular desktop environments

Cons: Fedora's priorities tend to lean towards enterprise features, rather than desktop usability; some bleeding edge features, such as early switch to KDE 4 and GNOME 3, occasionally alienate some desktop users

Software package management: YUM graphical and command line utility using RPM packages

## **Debian/Ubuntu/Mint Linux**

Впервые выпущен в 1993 году. Разрабатывается как совместный проект большого сообщества (свыше 1000 разработчиков-добровольцев). Поддерживает 11 процессорных архитектур. Цикл разработки включает 3 стадии: unstable, testing, stable. Стабильная версия выходит достаточно редко, поэтому может не включать некоторые актуальные нововведения.

В 2004 году выпущен основанный на Debian дистрибутив Ubuntu, ориентированный преимущественно на использование в качестве десктопа. Его разработка ведется гораздо более быстрыми темпами, чем основного дистрибутива Debian. Это приводит к тому, что зачастую Ubuntu используется в качестве серверной ОС администраторами, лучше знакомыми с базовым дистрибутивом, но желающими получить быстрый доступ к новым возможностям.

В 2006 году выпущен альтернативный, основанный на Ubuntu дистрибутив Linux Mint. Он поддерживается и разрабатывается собственным сообществом программистов. В настоящее время приобретает популярность в качестве альтернативы Ubuntu для десктопного использования в связи с неоднозначными инновациями в интерфейсе Ubuntu.

Pros: Very stable; remarkable quality control; includes over 20,000 software packages; supports more processor architectures than any other Linux distribution

Cons: Conservative - due to its support for many processor architectures, newest technologies are not always included; slow release cycle (one stable release every 1 - 3 years); discussions on developer mailing lists and blogs can be uncultured at times

Software package management: Advanced Package Tool (APT) using DEB packages

## **SUSE Linux**

Впервые анонсирован в 1992 году как ответвление проекта Slackware. Окончательно оформился как отдельный дистрибутив в 1996 году. Качественная документация и частые обновления способствовали росту популярности дистрибутива в начале 2000-х. Использует графический инструмент управления пакетами RPM: YaST. После приобретения дистрибутива компанией Novell в 2003 году, он стал основой для коммерческих продуктов этой компании. В то же время свободно распространяемая ветка проекта осталась таковой, сменив имя на openSUSE.

Pros: Comprehensive and intuitive configuration tool; large repository of software packages, excellent web site infrastructure and printed documentation

Cons: Novell's patent deal with Microsoft in November 2006 seemingly legitimised Microsoft's intellectual property claims over Linux; its resource-heavy desktop setup and graphical utilities are sometimes seen as "bloated and slow"

Software package management: YaST graphical and command-line utility using RPM packages

## **Mageia/Mandriva/Mandrake Linux**

Дистрибутив Mandrake впервые вышел в 1998 году как ветка дистрибутива Red Hat, использовавшая в качестве десктопа по умолчанию KDE. В дальнейшем дистрибутив был коммерциализирован как Mandriva, но в 2010 году компания обанкротилась. После этого основные разработчики сформировали команду и выпустили дистрибутив Mageia. Будущее этого дистрибутива пока туманно, поскольку он не имеет инфраструктуры и устойчивого цикла разработки, свойственных другим дистрибутивам Линукс.

Pros: Beginner-friendly; excellent central configuration utility; very good out-of-the-box support for dozens of languages; installable live media

Cons: Lacks reputation and mindshare following its fork from Mandriva, some concern over the developers ability to maintain the distribution long-term on a volunteer basis

Software package management: URPMI with Rpmdrake (a graphical front-end for URPMI) using RPM packages

### **1.1.4. Диаграмма развития дистрибутивов Linux**

Здесь можно увидеть полную диаграмму истории развития и связей между различными дистрибутивами Linux с 1992 года:

<http://futurist.se/gldt/wp-content/uploads/11.10/gldt1110.svg>

## **1.2. Основные средства и приёмы работы.**

### **1.2.1. Типы пользовательских интерфейсов**

- GUI (graphical user interface)– графический пользовательский интерфейс, в котором пользователь

взаимодействует с системой посредством графических элементов (иконки, кнопки, окна и пр.)  
В Unix основные графические среды: GNOME, KDE, Xfce

- TUI (text-based/textual/terminal user interface) – схватывает разные события от клавиатуры, может размещать на экран разноцветные символы
- CLI (command-line interface) – пользователь управляет системой, вводя текстовые команды

### 1.2.2. Unix shell

Unix shell – CLI в Unix-like системах, может группировать команды, представляет собой интерпретатор языка программирования.

В настоящее время существует много различных командных оболочек. Некоторые из них:

- sh – Bourne shell – первоначальный командный интерпретатор Unix, написанный Стефаном Борном (AT&T), предоставляет базовые возможности для всех командных оболочек
- csh – C shell – создан Биллом Джоем – выпускником университета Berkeley – C-style shell
- tcsh – улучшенный csh – много возможностей и удобств: встроенная история команд с возможностью использовать её в новых командах, автодополнение имён файлов и переменных и т.д.
- zsh – ориентированная на интерактивную работу пользователя оболочка
- bash – основная командная оболочка в большинстве дистрибутивов Linux – Bourne-again shell – написанный Брайаном Фоксом для GNU проекта как замена sh (1989) – включает в себя многие возможности из других оболочек.  
по умолчанию работают сочетания клавиш – emacs key binding. Можно переключить на vi-binding: командой "set -o vi", но особого удобства в этом нет.

### 1.2.3. Руководство пользователя

Для просмотра руководства пользователя для любой команды, которую можно запустить из командной строки, используется команда `man`. Эта команда выводит на экран руководство, написанное обычно разработчиком приложения, который запускается соответствующей командой.

`man имя_команды` – выдает справочную информацию о команде.

`man man` – выдает справочную информацию о себе.

## 1.3. Файловая структура.

### 1.3.1. Основные понятия

В Линукс используется единая древовидная файловая структура, не разделенная на диски/разделы. В основе структуры лежит корневой каталог (root) - это уровень файловой структуры, выше которого подняться по дереву невозможно. В Линукс корневой каталог обозначается символом `'/'` (прямой слэш).

Для подключения устройства хранения данных к файловой структуре используется *точка монтирования*. Точка монтирования представляет собой обычный пустой каталог. При выполнении команды **mount** с определенными параметрами (в общем случае, имя устройства и точка монтирования), структура файловой системы, существующей на устройстве, встраивается в общую структуру файлов начиная от точки монтирования.

Файловая структура Линукс содержит также *домашние каталоги*. Каждый пользователь по умолчанию получает при создании домашний каталог, обычно совпадающий по названию с именем пользователя и расположенный в директории `/home`.

### 1.3.2. Типы файлов

В файловой структуре Линукс различаются несколько типов файлов. Поскольку с точки зрения операционной системы файл представляет собой поток байтов, в качестве файлов могут быть представлены также физические устройства и некоторые другие объекты. Наряду с файлами и каталогами, файлами с точки зрения Линукс являются также:

- файлы физических устройств
- именованные каналы (named pipes)
- сокет (sockets)
- символические ссылки (symlinks)

Определить тип файла можно по первому символу вывода команды `ls -l`:

```
[student@ns lesson_2]$ ls -l
total 40
-rwxr-xr-x  1 root  root    2872 Aug 27 2001 arch
-rw-rw-rw-  1 root  root    612 Jun 25 2001 chain.b
brw-rw----  1 root  disk    3,  1 Feb  3 15:38 hda1
drwxrwxrwx  2 root  root   32768 Feb  3 15:38 ida
```

Интерпретация типов файлов:

- обычный файл, двоичная запись на дисковом носителе; лист в дереве файловой структуры
- d каталог (директория); узел в дереве файловой структуры
- l ссылка на файл или каталог
- b файл - блочное устройство
- c файл - символьное устройство
- s файл-сокет

## Файлы устройств

Физические устройства, представленные в виде файлов, могут быть символьными или блочными. Символьные устройства позволяют производить чтение и запись посимвольно, в режиме потока байтов. Запись и чтение блочных устройств производятся, как следует из названия, блоками, то есть группами байтов.

К символьным в основном относятся устройства ввода/вывода. Блочные - устройства хранения, например, магнитные диски.

Взаимодействием с физическими устройствами в Линукс управляют драйверы устройств, которые либо встроены в ядро, либо подключаются к нему в виде модулей. Об этом подробнее будет рассказано ниже. Для взаимодействия с остальными частями операционной системы каждый драйвер образует коммуникационный интерфейс, который выглядит как файл. Большинство таких файлов располагаются в каталоге /dev.

Наиболее часто используются следующие специальные файлы:

Имя файла	Описание
/dev/console	Системная консоль, т.е. монитор и клавиатура, физически подключенные к компьютеру
/dev/hd, /dev/sd	Жесткие диски с различными интерфейсами (IDE, SATA, SAS)
/dev/null	Устройство-черная дыра. Все, что записывается в него, теряется навсегда. Обычно используется для перенаправления вывода нежелательных сообщений
/dev/pty	Файлы поддержки псевдо-терминалов. Используются для обеспечения работы удаленных подключений (telnet, ssh)
/dev/random	Файл генератора случайных чисел
/dev/tty	Файлы поддержки пользовательских консолей. В Linux эти файлы обеспечивают работу виртуальных консолей
/dev/ttyS	Файлы, обеспечивающие работу с последовательными портами

Большинство из этих файлов кроме того имеют в составе цифру. Это требуется для того, чтобы различать множество однотипных устройств.

## Именованные каналы

Файлы этого типа служат для организации обмена данными между разными приложениями. Каналы представляют собой буферы типа FIFO (first in - first out). Все, что помещается одним процессом в канал, может быть прочитано другим процессом.

## Сокеты

Фактически представляют собой метки, позволяющие связать между собой несколько программ. После того, как связь установлена, данные передаются непосредственно между процессами с помощью ядра ОС. В целом представляют собой примерно то же, что и именованные каналы.

## Символические ссылки

Любой файл в Линукс может иметь более одного имени. В этом случае все имена файла называются "жесткими ссылками" (hard link). Файл физически остается на диске до тех пор, пока у него есть хотя бы одно имя (жесткая ссылка). Особенностью жестких ссылок является то, что они прямо указывают на номер индексного дескриптора, а, следовательно, такие имена могут указывать только на файлы внутри той же самой файловой системы (т.е., на том же самом носителе, на котором находится каталог, содержащий это имя). Другая особенность жестких ссылок - это то, что невозможно различить исходный файл и жесткие ссылки на него, созданные позднее.

Другой тип ссылок в Линукс - символические ссылки (symbolic links). В отличие от жестких ссылок, символические могут указывать на файлы, расположенные на другой файловой системе. Кроме того, символические ссылки не препятствуют удалению исходного файла. В этом случае символическая ссылка становится "висячей", то есть просто указывает в никуда.

Для создания ссылок используется команда **ln**. Символьную ссылку позволяет создать ключ **-s**:

```
ln /path/to/file /path/to/link
```

```
$ ln -s /home/user/file1 ~/link1
```

```
$ ls -la ~/link1
```

```
lrwxrwxrwx 1 user user 31 Dec 13 21:13 links1 -> /home/user/file1
```

### 1.3.3. Навигация по файловой структуре

Основные команды навигации по файловой структуре: **ls**, **cd**, **pwd**, **pushd**, **popd**.

- **ls** Команда отображает содержимое текущего каталога. Позволяет использовать различные ключи, наиболее важные: **-l** - расширенный формат вывода, содержит различные атрибуты файлов; **-a** - выводит данные о текущем каталоге (**..**) и родительском каталоге (**..**).
- **cd** Команда позволяет изменить текущий каталог на указанный параметром команды. При запуске без параметров позволяет перейти в домашний каталог текущего пользователя. Специальный параметр **..** позволяет перейти в каталог на один уровень выше по дереву каталогов, то есть в родительский каталог.
- **pwd** Команда показывает полное имя текущего каталога.
- **pushd** Команда изменяет текущий каталог на указанный параметром, аналогично команде **cd**. Кроме того, команда позволяет сохранять историю посещенных каталогов в виде стека.
- **popd** Команда изменяет текущий каталог, выбирая каталог назначения из стека истории посещенных каталогов, сформированного командой **pushd**.

### 1.3.4. Создание, удаление, копирование файлов

Копирование существующих файлов осуществляется командой **cp**. Эта команда принимает два параметра: имя файла источника и имя файла приемника. Если в качестве имени файла источника указывается директория, производится копирование файла в эту директорию без изменения имени. Часто используются следующие ключи:

- **p** сохранить атрибуты файлов при копировании (режим, владелец, время изменения)
- **R**, - **r** рекурсивное копирование директорий
- **s** создать символическую ссылку вместо копирования файла
- **l** создать жесткую ссылку вместо копирования файла

Для создания файлов может использоваться команда **touch**. Кроме того, создать файл позволяют некоторые программы, например, текстовые редакторы. Наконец, можно использовать для создания файлов оператор перенаправления вывода (**>**).

Удаление файлов осуществляется командой **rm**. Наиболее часто используются ключи:

- **f** принудительное удаление файлов
- **r** рекурсивное удаление директории и файлов, которые в ней содержатся

Перемещение файлов производится командой mv.

### 1.3.5. Операции с каталогами. Обзор структуры каталогов Linux.

#### Команды операций с каталогами

Основные операции с каталогами в целом производятся теми же командами, что и операции с файлами. Чтобы создать каталог, используется команда mkdir. Ключ -p позволяет создать также все промежуточные каталоги, вместо того, чтобы выдать ошибку, если какой-то из них не существует.

Для удаления каталога можно использовать команду rmdir. Эта команда позволяет удалить только пустой каталог.

#### Структура каталогов

Только что установленная система Linux имеет дерево каталогов следующей структуры:

Каталог	Описание
/	корневой каталог
/bin	основные системные программы
/boot	загрузочные файлы ядра ОС
/dev	файлы устройств
/etc	конфигурационные файлы системы и подкаталоги с конфигурационными файлами прикладных программ
/home	домашние каталоги пользователей
/lib, /lib64	динамические библиотеки (для архитектур i686 и x86_64 соответственно)
/lost+found	информация об удаленных файлах, при некоторых условиях помогающая восстановить данные
/media	стандартная точка монтирования сменных устройств
/mnt	точка монтирования постоянных устройств хранения
/root	домашний каталог суперпользователя
/sbin	системные программы, доступные для выполнения суперпользователем
/usr	прикладные программы и библиотеки
/tmp	временные файлы, используемые запущенными процессами, общедоступный каталог
/var	рабочие каталоги программ
/var/log	файлы журналов системы и отдельных процессов

## 1.4. Учётные записи и права доступа.

### 1.4.1. Понятие учётной записи и аутентификации. Учётная запись root.

Линукс - многопользовательская операционная система. Это означает, что для каждого файла в системе заданы следующие группы атрибутов: имя пользователя-владельца, имя группы владельцев, а также права на действия, которые имеют право произвести с файлом владелец, пользователи-члены группы владельцев и прочие пользователи: чтение, запись и исполнение.

Учетная запись пользователя представляет собой пару "имя-пароль", а также ряд связанных с именем пользователя свойств, в том числе:

- группа, к которой принадлежит пользователь,
- цифровой идентификатор пользователя (UID),
- командная оболочка,

- путь к домашнему каталогу,
- полное имя пользователя.

Данные об учетной записи пользователя хранятся в нескольких файлах:

**/etc/passwd** - параметры учетной записи, помимо пароля

Пример записи в файле:

```
root:x:0:0:root:/root:/bin/bash
```

Значения полей:

имя пользователя:пустое поле, в ранних версиях здесь хранился пароль пользователя:идентификатор пользователя (UID):идентификатор группы (GID):полное имя пользователя:домашний каталог пользователя:командная оболочка

**/etc/shadow** - пароль пользователя в зашифрованном виде

Пример записи в файле:

```
root:
$6$1sLl.zsTdfHXKgtl$wdBf0PWOWXMh9HNSMVeY58dzqwRp6Xe3PtAq2qBbJx02NoP
fy86yjoZUp.v5DnKbr0ZgoGMwAnlBsIGJMIM1P1:15324:0:99999:7:::
```

Значения полей:

имя пользователя:шифрованный пароль или символ '!' в случае, если пароль не установлен:число дней с последнего изменения пароля,начиная с 1 января 1970 года:число дней перед изменением пароля:число дней, после которого пароль должен быть изменен:число дней, за которое пользователь начнет получать предупреждение о том, что пароль устаревает:число дней между устареванием пароля и блокировкой учетной записи:число дней до блокировки учетной записи (с 1 января 1970):зарезервированное поле

**/etc/group** - идентификаторы групп и пользователи, входящие в группу

Для того, чтобы войти в систему с определенным именем пользователя, необходимо указать это имя, а также соответствующий пароль - аутентифицироваться.

В системе всегда существует учетная запись суперпользователя. Для нее зарезервировано имя **root**.

Суперпользователь может выполнять любые допустимые действия с любым файлом, независимо от того, кто является его владельцем и какова таблица прав этого файла. Кроме того, суперпользователь может изменять информацию о владельце любого файла и его таблицу прав. Также суперпользователь обладает значительно более широкими правами, не связанными с файловыми операциями.

Паролем может служить любая последовательность символов. Пароль хранится в системе в виде зашифрованной строки. Шифрование в современных системах производится алгоритмом md5 с использованием "соли" (salt) - короткой случайной последовательности символов, позволяющей повысить устойчивость шифра. "Соль" хранится вместе с зашифрованным паролем в файле **/etc/shadow**. При входе в систему строка, введенная пользователем, шифруется тем же алгоритмом и сравнивается с хранящейся зашифрованной строкой. Если строки совпадают, пароль введен правильно, и в системе запускается командная оболочка от имени этого пользователя.

Создание и изменение пароля осуществляются командой **passwd**. Суперпользователь **root** может также устанавливать пароль для любого другого пользователя, добавляя имя пользователя к команде **passwd**.

### 1.4.2. Создание и удаление учётных записей

Учетная запись суперпользователя, а в некоторых дистрибутивах - также запись обычного пользователя, создается в процессе инсталляции системы. В дальнейшем добавить пользователя в систему можно с помощью команды **adduser** (в некоторых системах также существует команда **useradd**, которая выполняет ту же функцию). По умолчанию эта команда создает пользователя с указанным именем, группу с тем же именем, добавляет пользователя в эту группу и создает для него домашний каталог. Для того, чтобы пользователь мог получить доступ к системе, ему необходимо также задать пароль. Это можно сделать командой **passwd**, или указав зашифрованную строку пароля вместе с ключом **-p** в команде **useradd**.

Для удаления учетной записи пользователя используется команда **userdel**.

### 1.4.3. Права доступа. Управление правами.

Базовые механизмы разграничения доступа сохранились практически в неизменном виде с первых версий ОС Линукс. В индексном дескрипторе каждого файла записаны имя так называемого владельца файла и группы, которая имеет права на этот файл. Первоначально, при создании файла его владельцем объявляется тот пользователь, который этот файл создал. Точнее — тот пользователь, от чьего имени запущен процесс, создающий файл. Группа тоже назначается при создании файла — по идентификатору группы процесса, создающего файл.

Владельца и группу файла можно поменять командами **chown** и **chgrp** соответственно.

Также в дескрипторе файла определяются права доступа для владельца, группы и остальных пользователей. Каждой



из этих трех категорий пользователей соответствуют три бита в формате дескриптора. Старший бит определяет право на чтение файла для категории пользователей, средний бит - право на запись в файл, младший бит - право на выполнение программы, записанной в файле.

Таким образом, различные комбинации битов определяют права доступа категории пользователей к файлу. Например, комбинация 100 дает право на чтение, 101 - на чтение и исполнение, и так далее. В результате для трех категорий пользователей в дескрипторе оказываются записаны три группы битов, то есть, фактически, три двоичных числа, которые можно перевести в десятичную систему. Так, праву на чтение соответствует десятичное число 4, праву на чтение и выполнение - 5 и так далее.

При использовании команды `ls -l` можно также увидеть права доступа к файлу для групп пользователей. Они записываются как три группы по три символа. Право на чтение обозначается символом `r`, на запись - `w`, на выполнение - `x`. Символ `}` означает отсутствие соответствующего права. Рассмотрим пример:

```
[user]$ ls -l /bin/ls
-rwxr-xr-x 1 root root 49940 Sep 12 1999 /bin/ls
```

Здесь для файла установлены права на чтение, запись и выполнение для владельца, чтение и выполнение для группы владельцев, чтение и выполнение для остальных пользователей.

Для изменения прав доступа к файлу используется команда `chmod`. Установить права с ее помощью можно двумя способами:

1) указать права в формате `XOY` или `XOY`, где `X` определяет категорию пользователей (`u` - владелец, `g` - группа, `o` - прочие, `a` - все категории); `O` определяет операцию (`+` - предоставить право, `--` лишить права, `=` - установить указанные права вместо имеющихся); `Y` определяет соответствующее право (`r`, `w`, `x`).

Например, команда, дающая всем право на выполнение файла `file_name`:

```
[user]$ chmod a+x file_name
```

Эта команда эквивалентна команде `chmod +x file_name`, поскольку по умолчанию права меняются для всех категорий пользователей.

Следующая команда удаляет право на чтение и запись для всех, кроме владельца файла:

```
[user]$ chmod go-rw file_name
```

2) указать права в виде десятичных цифр, соответствующих двоичной записи прав в дескрипторе, например:

```
[user]$ chmod 760 file_name
```

Эта команда задает права на чтение, запись и выполнение для владельца (`7 = 111`), чтение и запись группе (`6 = 110`) и никаких прав для остальных пользователей (`0 = 000`).

Использовать команду `chown` может только владелец файла и суперпользователь. Использовать команду `chgrp` могут также члены группы владельцев файла.

### Права на каталоги

Права на каталоги обозначаются так же, как права доступа к файлу, но имеют немного отличное значение. Флаг `r` дает право на просмотр содержимого каталога (например, командой `ls`). Флаг `[[w}}` позволяет создавать и удалять файлы в каталоге. При этом необязательно иметь право на изменение собственно файла. Флаг `x` дает право сделать этот каталог текущим (например, командой `cd`).

### Бит прикрепления

Бит прикрепления (sticky bit) - этот бит в формате дескриптора файла устанавливает специальные права. При выполнении программы из файла с установленным битом прикрепления копия её кода остается в памяти, что ускоряет последующий запуск. Однако в настоящее время бит прикрепления чаще используется для каталогов. В этом случае он запрещает удаление любого файла из этого каталога для всех, кроме владельца этого файла при условии, что у него есть права на изменение файла. Бит прикрепления устанавливается командой:

```
$ chmod +t filename
```