ОП.14 **Основы функционирования UNIX - систем**

ЗАНЯТИЕ 06

Файловая система
Права доступа к файлам

В операционной системе UNIX все объекты файловой системы имеют **определенные права доступа**, которые **позволяют или запрещают** выполнять те или иные операции как отдельным пользователям, так и группам пользователей.

С правами доступа (access permission) мы сталкивались при **анализе атрибутов файлов** в предыдущем разделе, сейчас же рассмотрим их более подробно.

Файловая система
Права доступа к файлам

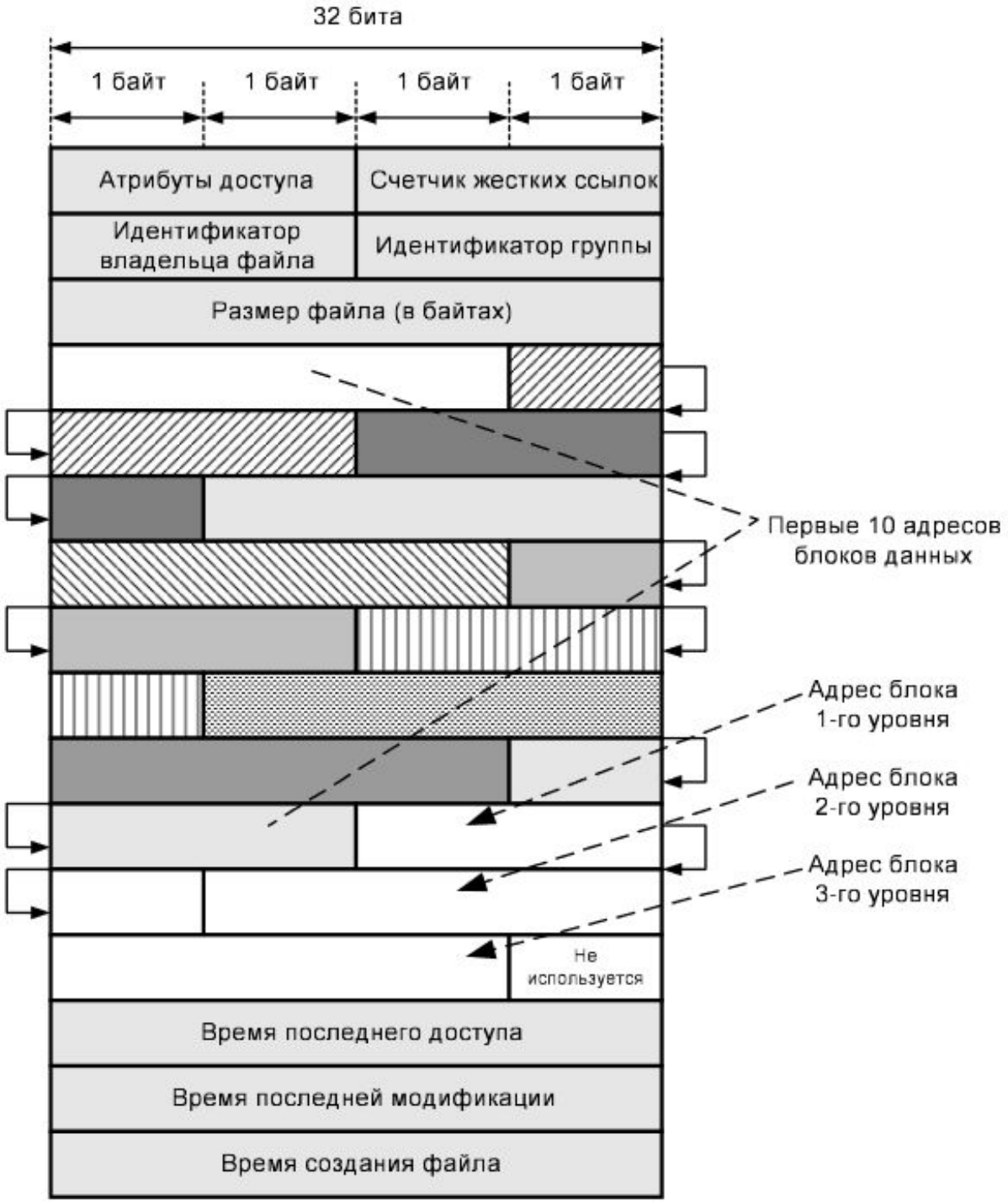
Каждый файл имеет **определенную комбинацию** прав доступа, состоящую из девяти битов.

Эта группа битов **определяет**, например, пользователей, имеющих право на чтение содержимого файла, на запись данных или выполнение, если файл является исполняемым.

Файловая система
Права доступа к файлам

Биты этого набора вместе с другими тремя битами, определяю щими способ запуска исполняемых файлов, образуют **код режима доступа к файлу**.

Все двенадцать битов режима хранятся в 16-битовом поле атрибутов доступа индексного дескриптора вместе с четырьмя дополнительными битами, указывающими тип файла (смотри рисунок).


32 бита
1 байт
1 байт
1 байт
1 байт
Атрибуты доступа
Счетчик жестких ссылок
Идентификатор владельца файла
Идентификатор группы
Размер файла (в байтах)
Первые 10 адресов блоков данных
Адрес блока 1-го уровня
Адрес блока 2-го уровня
Адрес блока 3-го уровня
Не используется
Время последнего доступа
Время последней модификации
Время создания файла

Файловая система
Права доступа к файлам

Четыре последних бита устанавливаются **при создании файла** и не должны изменяться.

Биты доступа может **изменить либо владелец файла, либо суперпользователь *root*** при помощи команды UNIX `chmod`.

Эти биты вместе с остальной информацией отображаются на экране командой `ls`.

Файловая система
Права доступа к файлам

В коде режима доступа есть биты **специального назначения** — им соответствуют восьмеричные значения 4000 и 2000.

Один из этих битов называется битом **замены идентификатора пользователя** (SUID).

Другой называется битом **замены идентификатора группы** (SGID).

Файловая система
Права доступа к файлам

Данные биты позволяют программам пользователя получать доступ к файлам и процессам, которые при иных обстоятельствах были бы пользователю недоступны.

Бит, которому соответствует восьмеричное значение 1000, называется **sticky-битом**.

В ранних системах он использовался, но современные UNIX-системы его игнорируют.

Файловая система
Права доступа к файлам

В UNIX-системах нельзя устанавливать биты прав доступа отдельно для каждого пользователя — вместо этого можно применять различные **комбинации из трех битов** (триады) для **владельца** файла, **группы**, которой принадлежит файл, и **прочих** пользователей.

Каждая триада включает **бит чтения, бит записи и бит выполнения** (для каталога последний называется битом поиска).

Очень часто код режима доступа представляют в виде **восьмеричного числа**, при этом каждая цифра в нем представляется тремя битами:

Файловая система
Права доступа к файлам

- три старших бита (восьмеричные значения 400, 200 и 100) определяют права доступа к файлу со стороны его владельца;
- три средних бита (восьмеричные значения 40, 20 и 10) за дают доступ для пользователей группы;
- три младших бита (восьмеричные значения 4, 2 и 1) опреде ляют права доступа к файлу для остальных пользователей.

Старший бит каждой триады определяет доступ по **чтению**, средний — по **записи**, и, наконец, младший бит определяет права на **выполнение**.

Файловая система
Права доступа к файлам

Каждый пользователь попадает только **в одну из категорий**, соответствующих одной из триад битов режима.

При этом из возможных комбинаций выбираются те, которые устанавливают самые **строгие права доступа**.

Например, права доступа для владельца файла всегда определяются триадой битов владельца и никогда — битами группы.

Файловая система
Права доступа к файлам

Для того чтобы удалить или переименовать файл, необходимо, чтобы были установлены соответствующие **биты прав доступа** для каталога, где хранится имя файла.

Установленный бит выполнения, например, **разрешает выполнить файл** как программу или командный сценарий.

Для каталогов этот бит имеет иной смысл. Если единственным установленным битом является только **бит выполнения**, то разрешается **вход в каталог**, но получить список его содержимого **нельзя**.

Файловая система
Права доступа к файлам

Содержимое каталога можно **просмотреть** только при установленных битах **чтения и выполнения**.

Установленные биты **записи и выполнения** позволяют **создавать**, **удалять** и **переименовывать** файлы в данном каталоге.

В операционной системе UNIX есть несколько команд, позволяющих устанавливать права доступа к файлам и каталогам — это команды `chmod` и `chown`, причем выполнять их может либо **владелец файла**, либо **суперпользователь *root***.

Команда `chmod` устанавливает права доступа к указанным в командной строке файлам и имеет следующий синтаксис:

Файловая система
Права доступа к файлам

Команда `chmod` устанавливает права доступа к указанным в командной строке файлам и имеет следующий синтаксис:

```
chmod [-fR] абсолютные_права  файл  ...
chmod [-fR] символьные_права  файл  ...
```

Файловая система
Права доступа к файлам

Первый параметр команды **указывает права доступа**.

Второй параметр и последующий задают **имена файлов и права доступа, подлежащие изменению**.

Код прав доступа можно задавать в виде **восьмеричного числа**.

В современных версиях поддерживается более наглядная система мнемонических обозначений.

Основным преимуществом этой системы является возможность **сброса/установки** отдельных битов режима.

Файловая система
Права доступа к файлам

В восьмеричной нотации первая цифра относится к **владельцу**, вторая — к **группе**, а третья — к **остальным пользователям**.

При необходимости задать биты SUID/SGID следует указывать не три, а **четыре восьмеричные цифры**, при этом первая цифра будет соответствовать трем специальным битам.

Опция `-f` команды блокирует вывод сообщения о невозможности установки прав доступа.

А с помощью опции `-R` можно **установить или изменить права доступа** для всех подкаталогов, указанных в списке.

Файловая система
Права доступа к файлам

Абсолютные права доступа задаются восьмеричным числом, в то время как символьные права доступа указываются в виде **списка выражений** (через запятую), например:

`[`*`пользователи`*`]  `*`оператор`*`  [`*`права`*`, ...]`

Элемент *`пользователи`* списка определяет, для кого задаются или изменяются права.

Файловая система
Права доступа к файлам

Он может принимать значения `u`, `g`, `o` и `a`, относящиеся к владельцу, группе, остальным пользователям, а также ко всем категориям пользователей соответственно.

При этом:

- символ `u` **(user)** обозначает владельца файла,
- символ `g` **(group)** — группу,
- символ `o` **(others)** — других пользователей,
- символ `a` **(all)** — всех пользователей сразу.

Файловая система
Права доступа к файлам

Если элемент *пользователи* **не указан**, изменения прав действуют для всех категорий пользователей, хотя не переопределяются установки, задаваемые маской создания файлов.

Элемент *оператор* списка может принимать значения +, – или =, означающие **добавление**, **отмену** права доступа и **установку** указанных прав соответственно.

Если после оператора = ничего не указано, то все права доступа для соответствующих категорий пользователей **отменяются**.

Файловая система
Права доступа к файлам

Компонент *права* задается в виде любой совместимой комбинации следующих символов: `g`, `s`, `t`, `u`, `x`. Не все сочетания символов для компонента *пользователи* и компонента *права* допустимы.

Так, `s` можно задавать только для `u` или `g`, а `t` – только для `u`. Права `x` и `s`, например, несовместимы с `t` и т. д.

Изменения прав доступа в списке выполняются последовательно, в порядке их перечисления.

Все возможные комбинации для каждого трехбитового набора перечислены в следующей таблице (символы `r`, `w` и `x` обозначают **чтение**, **запись** и **выполнение** соответственно).

Файловая система

Права доступа в команде *chmod*

Восьмеричное число	Двоичное число	Маска режима доступа
0	000	—
1	001	--x
2	010	-w-
3	011	-wx
4	100	r--
5	101	r-x
6	110	rw-
7	111	rwx

Файловая система
Права доступа к файлам

Например, команда

```
chmod  744  text
```

предоставляет владельцу все права доступа к файлу `text` и право на чтение — остальным пользователям.

Несколько примеров мнемонических обозначений приведено в следующей таблице.

Файловая система

Примеры мнемонических спецификаций команды *chmod*

Спецификация	Описание
u+w	Владельцу файла дополнительно разрешается запись в файл
ug=rwx,o=rx	Владелец и группа получают право на чтение/запись и выполнение, остальные пользователи — право чтения и выполнения
a-wx	Все пользователи лишаются права записи и выполнения
ug=srx,o=	Владелец и группа получают право чтения/выполнения, и устанавливается бит SUID; доступ к файлу остальным пользователям запрещен
g=u	Группе назначаются такие же права, что и владельцу

Файловая система
Права доступа к файлам

Рассмотрим пример изменения прав доступа.

Предположим, что в текущем каталоге имеется файл ***test***, для которого мы будем изменять права доступа, как показано далее, одновременно наблюдая за результатами:

```
$ chmod +w test
$ ls -l test
-rw-r--r--   1 root 50              0 Sep 21 12:07 test
```

Файловая система
Права доступа к файлам

```
$ chmod a+w test
$ ls -l test
-rw-rw-rw-   1 root 50          0 Sep 21 12:10 test
$ chmod u+x,g=x,o= test
$ ls -l test
-rwx--x---   1 root 50          0 Sep 21 12:12 test
```

Файловая система
Права доступа к файлам

```
$ chmod ug-x,og+r,u=rwx test
$ ls -l test
-rwxr--r--   1 root 50            0 Sep 21 12:15 test
$ chmod 644 test
$ ls -l test
-rw-r--r--   1 root 50            0 Sep 21 12:18 test
```

Файловая система
Права доступа к файлам

При создании нового файла ему присваиваются права доступа, определяемые пользовательской маской режима создания файлов.

Для этого используется встроенная команда `umask` интерпретатора ***shell***, которая присваивает пользовательской маске режима создания файлов указанное восьмеричное значение.

Три восьмеричные цифры соответствуют правам на чтение/запись/выполнение для владельца, членов группы и прочих пользователей соответственно.

Файловая система
Права доступа к файлам

Команда имеет следующий синтаксис:

`umask [-S] [`*`маска`*`]`

Выполненная без параметров, команда `umask` отображает текущее значение маски в восьмеричном виде.

Маска используется при указании прав доступа следующим образом: ее значение необходимо "вычесть" из максимальных прав доступа (777 для исполняемых файлов и 666 для обычных файлов), например:

```
$ umask 022
```

Файловая система
Права доступа к файлам

При указанном значении маски обычные текстовые файлы будут создаваться с правами доступа 666 — 022 = 644, как это имеет место в следующем примере при создании

```
$ > test

$ ls -l test

-rw-r--r--   1 root 50              0 Sep 12 14:39 test
```

Операция "вычитания" для значения маски формально выполняется как побитовое логическое И дополнения маски и максимальных прав доступа.

В нашем примере расчет прав доступа с использованием маски показан в следующей таблице.

Файловая система

Расчет маски

Последова-тельность выполнения	Значение	Результат
Шаг 1	Двоичное значение маски	000010010 (022)
Шаг 2	Дополнение маски	111101101 (755)
Шаг 3	Максимальное значение прав	110110110 (666)
Шаг 4	Логическое И предыдущих двух строк	110100100 (644)
Шаг 5	Результирующие биты прав	110100100 (644)

Файловая система
Права доступа к файлам

Опция `-s` позволяет выводить значение маски в символьном виде, при этом отображаются биты прав доступа у вновь создаваемого файла.

Бит выполнения в этом случае также берется во внимание:

```
$ umask -S
```

```
u=rwx, g=rx, o=rx
```

Команду `umask` нередко включают в файлы начального запуска, устанавливающие рабочую среду для командного интерпретатора, используемого по умолчанию.

Файловая система
Права доступа к файлам

Далее приводится еще один пример создания файла при значении маски, равном 257:

```
$ umask 257
$ > test
$ ls -l test
-r-x--w----    root 1   50        0 Sep 14 17:14 test
```

Файловая система
Права доступа к файлам

Владелец файла, а также суперпользователь `root` могут изменить владельца и группу-владельца файла, используя команду `chown`, имеющую следующий синтаксис:

`chown [-h][-R] `*`владелец`*`[:`*`группа`*`] `*`файл`*` ...`

Для замены группы, владеющей файлом, можно использовать команду `chgrp`:

`chgrp [-h][-R] `*`группа`*` `*`файл`*

Файловая система
Права доступа к файлам

В качестве первого параметра обеих команд используется имя нового владельца файла или новой группы соответственно.

Выполнить команду `chgrp` может только владелец файла, пользователь, входящий в назначаемую группу, или суперпользователь `root`.

Опция `-h` требует замены владельца файла, на который указывает символическая ссылка, а не самой ссылки, как это принято по умолчанию.

Файловая система
Права доступа к файлам

В большинстве версий команд `chown` и `chgrp` предусмотрен флаг `-R`, который задает смену владельца или группы не только самого каталога, но и всех его подкаталогов и файлов.

В операционных системах UNIX, совместимых с System V, пользователи могут беспрепятственно поменять владельца собственного файла при помощи команды `chown`, тогда как в BSD-совместимых системах эту команду может выполнять только суперпользователь `root`.

Файловая система
Права доступа к файлам

Необходимо учитывать, что после смены владельца файла бывший владелец будет иметь права доступа, установленные новым владельцем.

Пусть требуется сменить владельца файла `test`, атрибуты которого показаны далее:

```
$ ls -l
total 2
-rw-r--r--  1 user1  others    6 Sep 10 16:19 test
```

Файловая система
Права доступа к файлам

Из результата выполнения команды `ls` видно, что владельцем файла `test` является пользователь `user1`, в сеансе которого и выполняются команды.

Если заменить владельца файла `test` с `user1` на `root`, выполнив команды:

```
$ chown root test
$ ls -l
total 2
-rw-r--r--   1 root others     6 Sep 10 16:19 test
```

Файловая система
Права доступа к файлам

то увидим, что после выполнения команды `chown` пользователем файла `test` является `root`.

Теперь пользователь `user1` никаким образом не сможет в своем сеансе заменить владельца опять на `user1`:

```
$ chown user1 test
```

```
UX:chown: ERROR: testf: Not privileged
```

Полученное сообщение об ошибке говорит о том, что у пользователя `user1` нет прав привилегированного пользователя для смены владельца файла.

Файловая система
Права доступа к файлам

Помимо рассмотренных нами атрибутов доступа, объекты файловой системы операционной системы UNIX имеют и другие атрибуты, которые можно получить, используя команду `ls -l`, как в этом примере:

```
$ ls -l /bin/sh
-rwxr-x--x 1 root  bin  87924 Sep 21 2005 /bin/sh
```

Файловая система
Права доступа к файлам

Проанализируем полученный результат.

Здесь в первом поле задается тип файла и маска режима доступа к нему: поскольку первым символом является дефис, то это обычный файл.

Обозначения различных типов файлов представлены кодами, состоящими из одного символа (смотри таблицу).

Файловая система

Обозначение типов файлов в выводе команды `ls -l`

Тип файла	Символ	Создается командой	Удаляется командой
Обычный файл	—	Редакторы, `cp` и др.	—
Каталог	`d`	`mkdir`	`rmdir`, `rm -r`
Файл байт-ориенти-рованного устройства	`c`	`mknod`	`rm`

Файловая система
Права доступа к файлам

Вернемся к результату выполнения команды `ls`.

Следующие за дефисом девять символов первого поля представляют триады битов режима, обозначенные литерами `r`, `w` и `x` (чтение, запись и выполнение соответственно).

Для данного примера владелец обладает полным доступом к файлу, пользователи группы `bin` — правом на чтение и выполнение, а остальные пользователи могут только выполнить этот файл.

Файловая система
Права доступа к файлам

Если бы был установлен бит смены идентификатора пользователя (SUID), то вместо `x` стоял бы символ `s`.

Аналогично, если бы был установлен бит смены идентификатора группы (SGID), то вместо `x` для группы стоял бы символ `s`.

Последний бит режима (право выполнения для остальных пользователей) представляется буквой `t`, когда для файла задан **sticky**-бит.

Если биты SUID/SGID или **sticky**-бит установлены, а надлежащий бит выполнения — нет, эти биты представляются, соответственно, символами, указывающими на наличие ошибки и игнорирование данных атрибутов.

Файловая система
Права доступа к файлам

В следующем поле вывода команды `ls` отображается количество жестких ссылок на файл — оно равно 1, а это значит, что `/bin/sh` — единственное имя, под которым известен данный файл.

Всякий раз при создании жесткой ссылки на файл зна чение счетчика ссылок увеличивается на единицу, при этом символические ссылки в счетчике не учитываются.

Что же касается каталогов, то любой из них имеет, как минимум, две жесткие ссылки: одну из родительского каталога и одну из специального файла внутри самого каталога.

Файловая система
Права доступа к файлам

Следующие два поля отображают владельца и группу-владельца файла — в данном примере владельцем файла является `root`, а файл принадлежит группе `bin`.

Ядро UNIX хранит эти данные не в виде строк, а как идентификаторы пользователя и группы.

Если получить символьное представление имен по какой-либо причине невозможно, то в этих полях будут отображаться числа.

Такое случается, если запись пользователя или группы была удалена из файла `/etc/passwd` или `/etc/group` соответственно.

Файловая система
Права доступа к файлам

Далее следует поле, отображающее размер файла в байтах: данный файл имеет размер 87 924 байта, т. е. почти 88 Кбайт.

Следующее поле содержит дату последнего изменения: 21 сентября 2005 г., и, наконец, в последнем поле вывода содержится имя файла: `/bin/sh`.

Если мы имеем дело с файлом устройства, то вывод команды `ls` будет другим, например:

```
$ ls -l /dev/ttya
crw-rw-rw-  1   root  daemon     12,  0  Dec  20  1998
/dev/ttya
```

Файловая система
Права доступа к файлам

Результат работы этой команды иной, чем в предыдущем примере.

Вместо размера в байтах показаны старший и младший номера устройства, а в первом поле отображается тип устройства (в данном случае литера `c` в первой позиции означает, что устройство имеет символьный тип).

Имя `/dev/ttya` относится к первому устройству, управляемому драйвером устройства 12 (в данной системе это драйвер терминала).

Файловая система
Права доступа к файлам

При поиске жестких ссылок часто бывает полезной команда `ls -i`, отображающая для каждого файла номер индексного дескриптора. Жесткие ссылки, указывающие на один и тот же файл, будут иметь один и тот же номер.

Операционная система автоматически отслеживает такие атрибуты, как время изменения, число ссылок и размер файла, автоматически устанавливая корректные значения.

В то же время права доступа и идентификаторы принадлежности файла могут быть модифицированы явным образом с помощью команд `chmod`, `chown` и `chgrp`.

Список литературы:

1. Юрий Магда. UNIX для студентов, Санкт-Петербург «БХВ-Петербург», 2007.
2. Unix и Linux: руководство системного администратора, 4-е издание, 2012, Э. Немет, Г. Снайдер, Т. Хейн, Б. Уэйли
3. Организация UNIX систем и ОС Solaris 9, Торчинский Ф.И., Ильин Е.С., 2-е издание, исправленное, 2016.

Спасибо за внимание!

Преподаватель: Солодухин Андрей Геннадьевич

Электронная почта: asoloduhin@kait20.ru