# Тема:
# Права доступа и атрибуты файлов

# План занятия:

1. Права доступа к файлам и директориям. Категории пользователей. Типы файлов. Атрибуты. Права доступа.

2.Таблица прав доступа. Символьное и числовое представление прав доступа.

3.Изменение прав доступа. Команды chmod, ls –l.

4. Атрибуты файлов и директорий. Специальные атрибуты файлов и каталогов.

5. Символьные ссылки (Symbolic links).

# 1. Права доступа к файлам и директориям. Категории пользователей. Типы файлов. Права доступа.

**Атрибуты** файлов и каталогов - это дополнительные метаданные, связанные с файлами и каталогами в операционных системах, таких как Linux. Они содержат такую информацию о файлах или каталогах, как права доступа, владельца, временные метки и другие характеристики.

**Права доступа** (permissions) - это механизм в операционных системах, таких как Linux, который регулирует уровень доступа к файлам и каталогам. Они определяют, какие операции могут быть выполнены с файлом или каталогом, и кем они могут быть выполнены.

Права доступа определяют, какие операции могут быть выполнены с файлом (например, чтение, запись, выполнение) и кто может выполнять эти операции.

# Зачем нужно разграничивать права доступа

**Безопасность**. Одной из основных функций прав доступа является обеспечение безопасности системы. Установка соответствующих прав предотвращает несанкционированный доступ к файлам и директориям.

**Предотвращение конфликтов**. Правильная установка прав доступа помогает предотвратить конфликты между пользователями и группами, особенно в многопользовательской среде.

**Совместное использование ресурсов**. Права доступа и атрибуты файлов обеспечивают возможность совместного использования ресурсов между пользователями и группами, предотвращая конфликты при одновременном доступе.

и др.

# Типы файлов в Linux

1. Обычный файл ( **-** **).** Обычные файлы содержат данные, такие как текстовые документы, изображения, исполняемые файлы и т. д.

2. Каталог ( **d ).** Каталоги используются для организации и хранения файлов и других каталогов.

3. Символическая ссылка ( **l ).** Символические ссылки представляют собой ссылки на другие файлы или каталоги в файловой системе.

4. Файл устройств ( **c** или **b ).** Файлы устройств символьного типа представляют символьные устройства, такие как терминалы или модемы.

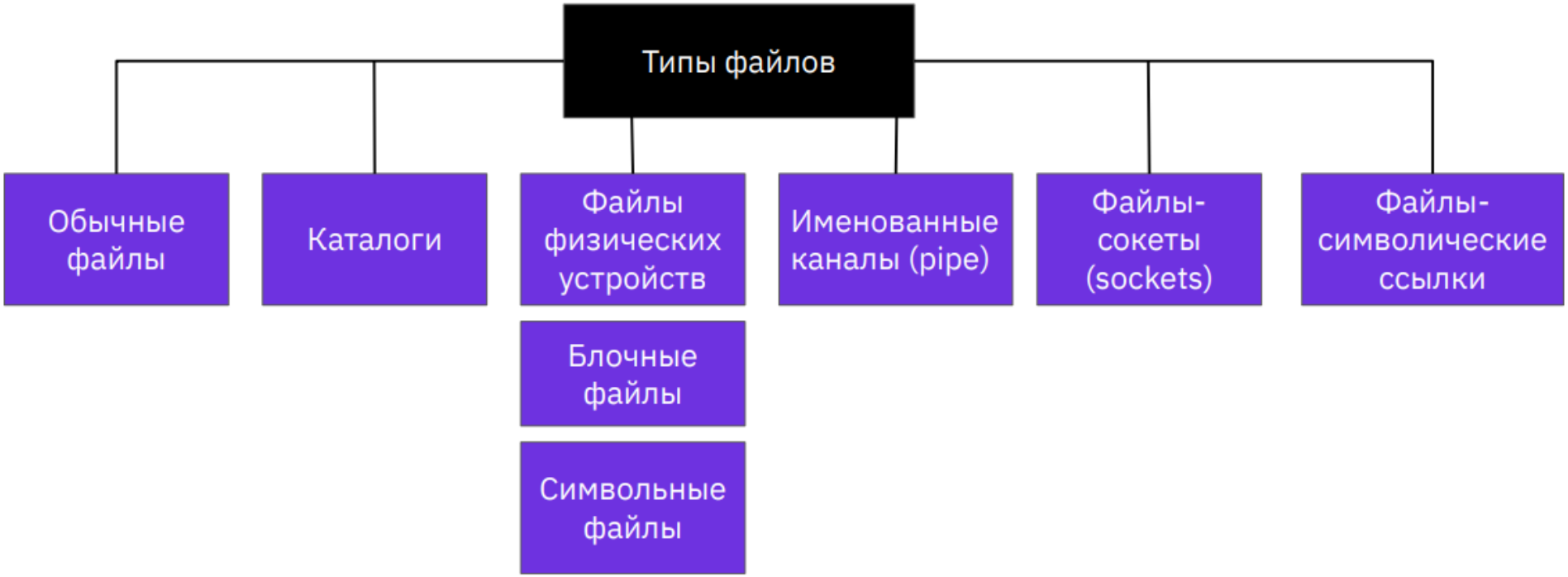
Файлы устройств блочного специального типа представляют блочные устройства, такие как жесткие диски или флэш-накопители.

6. FIFO (First-In, First-Out) ( **p ).** FIFO-файлы (или именованные каналы) используются для передачи данных между процессами.

7. Сокет ( **s ).** Сокеты представляют собой точки соединения, используемые для обмена данными между процессами через сеть или между процессами на локальном компьютере.

8. Специальные файлы, например /dev/null, который используется для отбрасывания вывода.

# Типы файлов в Linux



# Типы файлов в Linux

| Типы файлов | | Назначение |
|---|---|---|
| Обычные файлы | — | Хранение символьных и двоичных данных |
| Каталоги | d | Организация доступа к файлам |
| Символьные ссылки | l | Предоставление доступа к файлам, расположенных на любых носителях |
| Блочные устройства | b | Предоставление интерфейса для взаимодействия с аппаратным обеспечением компьютера |
| Символьные устройства | c | |
| Каналы | p | Организация взаимодействия процессов в операционной системе |
| Сокеты | s | |

# Категории пользователей в Linux

**Владелец файла** в Linux - это пользователь, который создал файл или директорию. Владелец имеет особые привилегии и полный контроль над файлом. Он может читать, записывать и выполнять файл, а также изменять его права доступа и атрибуты.

**Группа владельца файла** - это группа пользователей, к которой принадлежит файл или директория. Пользователи, принадлежащие к этой группе, могут иметь определенные привилегии доступа к файлу. При создании файла в Linux, система присваивает ему владельца (пользователя) и группу владельца. По умолчанию, группа владельца файла соответствует основной группе пользователя, создавшего файл.

**Остальные пользователи** - это все пользователи в системе, которые не являются владельцем файла и не принадлежат к группе владельца. Права доступа для остальных пользователей могут быть установлены отдельно и отличаться от прав доступа владельца и группы владельца.

# Основные права доступа

Каждый из этих трех типов пользователей имеет свои права доступа к файлам и директориям, которые представлены символами:

- **r (read):** Право на чтение файла или содержимого директории. Если установлено для файла, пользователь или группа могут прочитать содержимое файла.
- **w (write):** Право на запись в файл или в директорию, а также создание или удаление файлов в директории. Если установлено для файла, пользователь или группа могут изменять содержимое файла или создавать новые файлы в директории.
- **x (execute):** Право на выполнение файла или вход в директорию. Если установлено для файла, пользователь или группа могут запускать исполняемые файлы или входить в директорию.

Эти права доступа отображаются в виде строки, состоящей из девяти символов. Три символа для владельца, три для группы владельца и три для остальных пользователей.

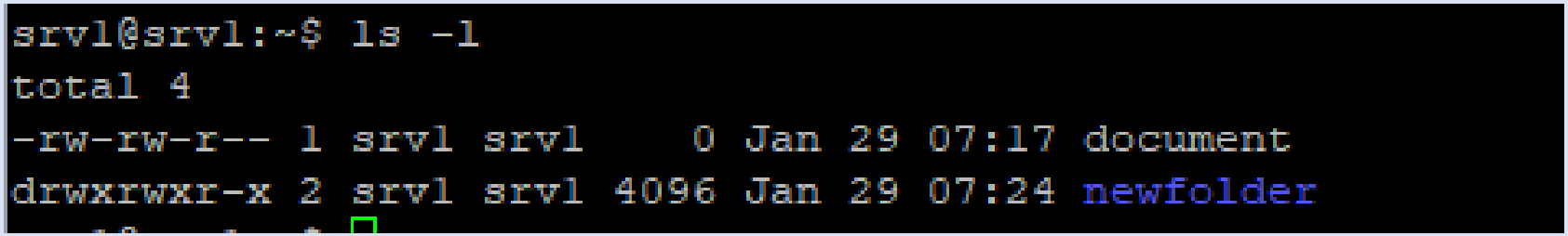
# 2.Таблица прав доступа. Символьное и числовое представление прав доступа. Основные права доступа

Права доступа являются одним из основных атрибутов файлов в Unix-подобных системах.

Права доступа определяются для каждого файла или каталога.

Права доступа представлены в виде трех групп: владелец файла, группа пользователя и остальные пользователи. Для каждой группы определены три типа прав доступа: чтение (read), запись (write) и выполнение (execute).

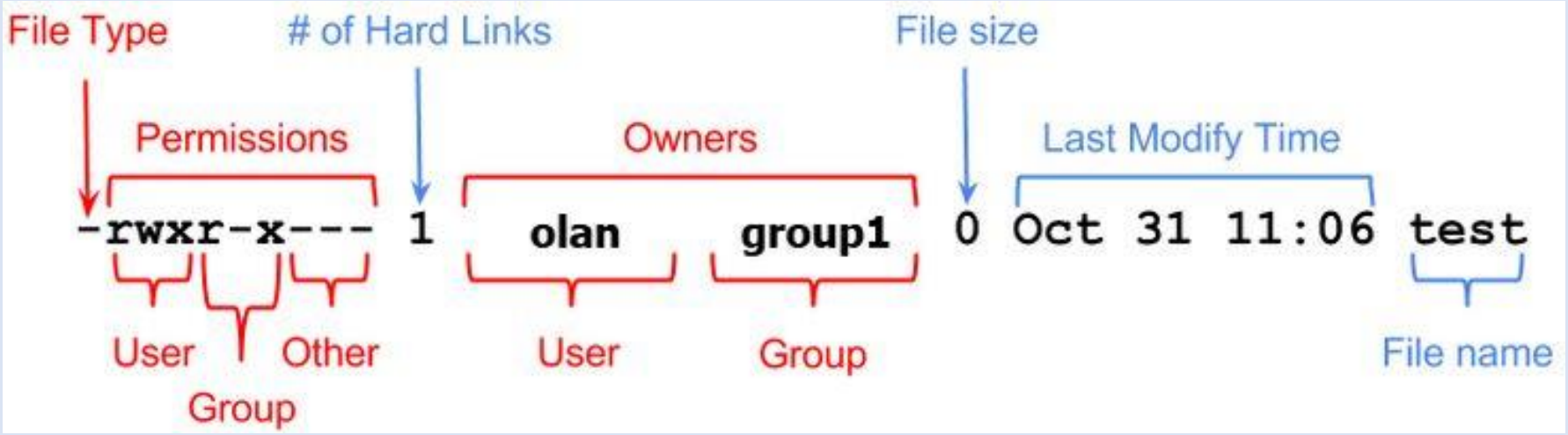
# Представление прав доступа




В этом примере:

- Владелец файла document (srv1) имеет права на чтение и запись (rw).
- Группа владельца (srv1) имеет права на чтение и запись (rw).
- Остальные пользователи имеют права на чтение (r).

Право выполнения на каталог определяет, есть ли у пользователя или группы пользователей разрешение на переход внутрь каталога и выполнять операции с его содержимым, такие как просмотр файлов внутри каталога или доступ к подкаталогам.




Permissions (права доступа)

Hard links (количество жестких ссылок)

User (владелец)

Group (группа)

Other (остальные пользователи)

File size (размер)

Last modify time (дата и время модификации)

File name (имя)

# Представление прав доступа

Помимо символьного представления прав доступа в Linux, существует также числовое представление, известное как восьмеричная нотация (octal notation). Восьмеричные числа отображают комбинацию трех битов для каждого из трех типов пользователей: владельца, группы владельца и остальных пользователей.

Каждое право доступа (чтение, запись, выполнение) представлено числом:

r (read) - 4 (100)

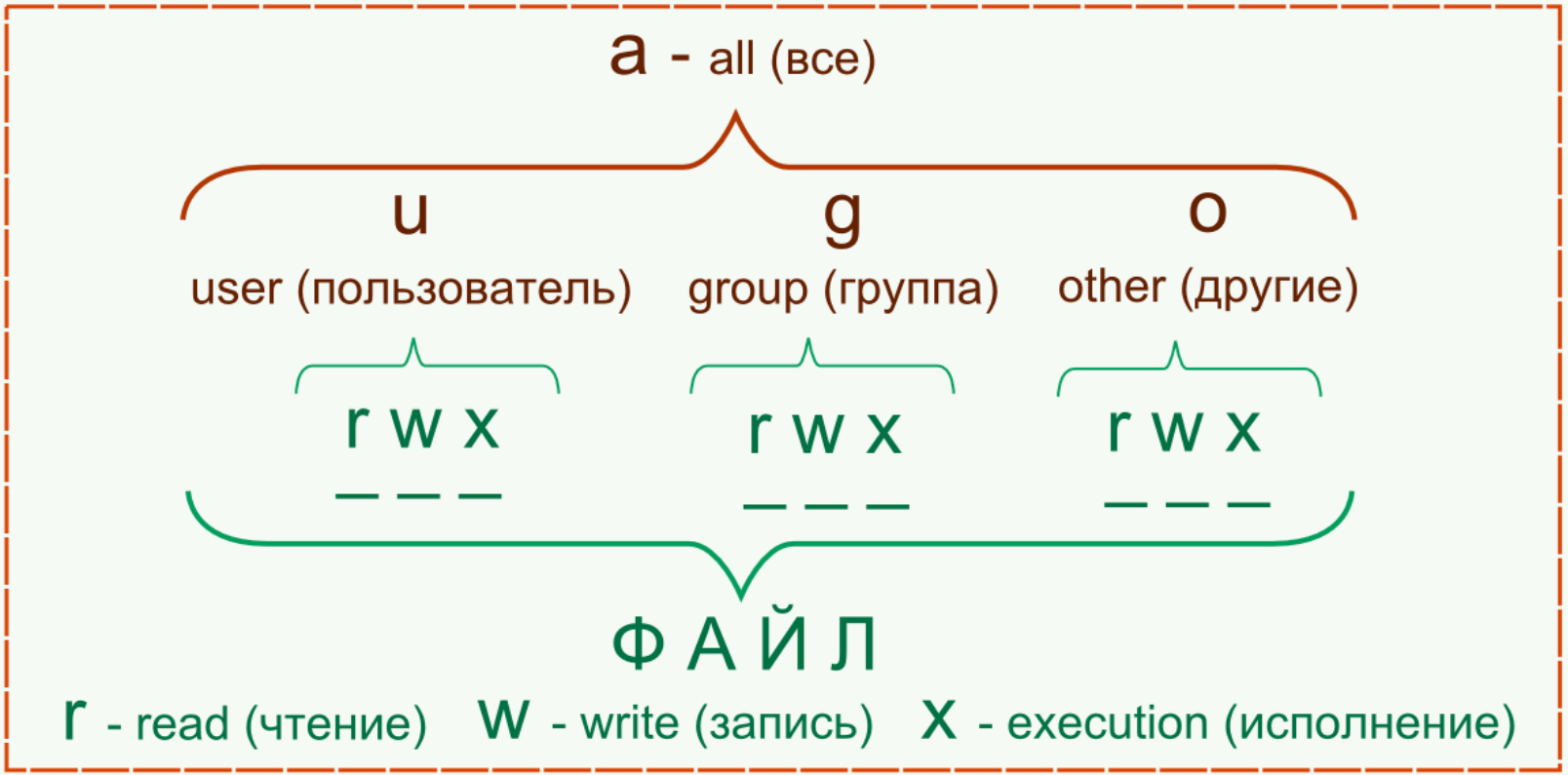
w (write) - 2 (010)

x (execute) - 1 (001)

Сумма этих чисел дает комбинации прав доступа для каждого типа пользователя. Вот несколько примеров:

- Права доступа "rwxr-xr--" в символьном представлении соответствуют восьмеричному числу 754.
- Права доступа "rw-rw-r--" в символьном представлении соответствуют восьмеричному числу 664.
- Права доступа "rwx------" в символьном представлении соответствуют восьмеричному числу 700.

# Права доступа к файлам (rwx)




*Примеры:*

| | |
|---|---|
| r w - r w - r w - | (все могут читать и изменять ) |
| r w x - - - - - - | (полный доступ имеет владелец файла) |
| r w - r - - r - - | (все могут читать, владелец также изменять) |
| r w x r - x r - x | (все могут читать и исполнять, владелец также изменять) |

| Три варианта записи прав пользователя | | | | |
|---|---|---|---|---|
| **двоичная** | **восьмеричная** | **символьная** | **права на файл** | **права на директорию** |
| 000 | 0 | `---` | нет | нет |
| 001 | 1 | `--x` | выполнение | чтение файлов и их свойств |
| 010 | 2 | `-w-` | запись | нет |
| 011 | 3 | `-wx` | запись и выполнение | всё, кроме чтения списка файлов |
| 100 | 4 | `r--` | чтение | чтение имён файлов |
| 101 | 5 | `r-x` | чтение и выполнение | доступ на чтение |
| 110 | 6 | `rw-` | чтение и запись | чтение имён файлов |
| 111 | 7 | `rwx` | все права | все права |

# 3.Изменение прав доступа. Команды chmod, ls –l.

Команда chmod (change mode) в Linux используется для изменения прав доступа к файлам и каталогам.

Она позволяет изменять права доступа для владельца файла, группы владельца и остальных пользователей.

Синтаксис команды chmod:

```
chmod [опции] права_доступа файлы_или_каталоги
```

Права доступа в chmod могут быть заданы с использованием различных форматов:

Численный формат:

0 - нет прав доступа.

1 - выполнение (x).

2 - запись (w).

3 - выполнение и запись (x+w).

4 - чтение (r).

5 - чтение и выполнение (r+x).

6 - чтение и запись (r+w).

7 - чтение, запись и выполнение (r+w+x).

В численном формате права доступа задаются суммой значений для владельца, группы и других пользователей.

Например, chmod 755 file.txt устанавливает права доступа для владельца на чтение, запись и выполнение, а для группы и остальных пользователей на чтение и выполнение.

Символьный формат:

r - чтение.

w - запись.

x - выполнение.

+ -добавить права.

- -удалить права.

= -установить права точно.

В символьном формате права доступа задаются в виде комбинации символов для владельца, группы и других пользователей.

Например, chmod u=rw,g=r,o=r file.txt устанавливает чтение и запись для владельца, чтение для группы и остальных пользователей.

Опции, часто используемые с командой chmod:

-R (рекурсивно): Применить изменение прав доступа рекурсивно ко всем файлам и подкаталогам внутри указанного каталога.

-v (verbose): Выводить подробный вывод о каждом изменении прав доступа.

-c (changes): Выводить информацию только о файлах, для которых были изменены права доступа.

```
chmod -Rv -c 755 directory
```

# 4. Атрибуты файлов и директорий. Специальные атрибуты файлов и каталогов.

**SetUID (SUID)** – это бит разрешения, который позволяет пользователю запускать исполняемый файл с правами владельца этого файла. Другими словами, использование этого бита позволяет нам поднять привилегии пользователя в случае, если это необходимо. Классический пример использования этого бита в операционной системе это команда sudo.

Установка бита SUID обычно обозначается символом "s" вместо бита для выполнения владельцем (x) в выводе ls -l.

```
[vvv@localhost ~]$ ls -l /bin/sudo
---s--x--x. 1 root root 151424 Jan 25  2023 /bin/sudo
[vvv@localhost ~]$ _
```

**SetGID (SGID):** Установка бита SGID для исполняемого файла позволяет запускать его с правами группы владельца файла вместо прав вызывающего пользователя. Это часто используется для обеспечения совместного доступа к файлам и каталогам для пользователей в одной группе.

Установка бита SGID также обозначается символом "s" вместо бита для выполнения группой (x) в выводе ls -l.

```
svv@server22:~$ ls -l /usr/bin/crontab
-rwxr-sr-x 1 root crontab 39568 Mar 23  2022 /usr/bin/crontab
svv@server22:~$
```

**Sticky Bit:** Установка "sticky bit" для каталога предотвращает удаление или переименование файлов в этом каталоге другими пользователями, за исключением владельца файла, владельца каталога или суперпользователя. Это полезно для общих каталогов, где нужно предотвратить случайное удаление файлов другими пользователями.

Установка sticky bit обозначается символом "t" вместо бита для выполнения остальными пользователями (x) в выводе ls -l.

Например, системная папка /tmp, эта папка разрешена на запись любому пользователю, но удалять файлы в ней могут только пользователи, являющиеся владельцами этих файлов.

```
[vvv@localhost ~]$ ls -ld /tmp
drwxrwxrwt. 7 root root 93 Feb 15 19:08 /tmp
[vvv@localhost ~]$
```

Установка и удаление специальных атрибутов.

Для установки **SUID**-бита используйте chmod с опцией u+s:

```
chmod u+s file
```

Для установки **SGID** -бита используйте chmod с опцией g+s:

```
chmod g+s directory
```

Для установки sticky **bit** используйте chmod с опцией +t:

```
chmod +t directory
```

# Команда stat

Команда stat в Linux используется для отображения подробной информации о файле или каталоге, включая его метаданные.

```
stat [опции] файл
```

Некоторые распространенные опции команды stat:

-f или --file-system: Отображает информацию о файловой системе, на которой находится файл.

-c или --format: Позволяет определить пользовательский формат вывода информации.

-t или --terse: Выводит информацию в более компактном формате.

-L или --dereference: Если файл является символической ссылкой, отображает информацию о файле, на который ссылается символическая ссылка, а не саму ссылку.

Примеры:

```
svv@server22:~$ stat hello.sh
  File: hello.sh
  Size: 79              Blocks: 8          IO Block: 4096   regular file
Device: fd00h/64768d    Inode: 394656      Links: 1
Access: (0775/-rwxrwxr-x)  Uid: ( 1000/     svv)   Gid: ( 1000/     svv)
Access: 2024-02-15 11:39:05.191805033 +0000
Modify: 2023-12-12 11:07:30.609093949 +0000
Change: 2023-12-12 11:07:30.609093949 +0000
 Birth: 2023-12-12 05:21:47.767908294 +0000
```

```
svv@server22:~$ stat -f hello.sh
  File: "hello.sh"
    ID: ab559b2c13f94a93 Namelen: 255     Type: ext2/ext3
Block size: 4096       Fundamental block size: 4096
Blocks: Total: 2102113    Free: 4096       Available: 0
Inodes: Total: 540672     Free: 391446
```

```
svv@server22:~$ stat -t hello.sh
hello.sh 79 8 81fd 1000 1000 fd00 394656 1 0 0 1707997145 1702379250 1702379250 1702358507 4096
```

# 5. Жесткие (Hard links) и символьные ссылки (Symbolic links).

Жесткие ссылки и символические ссылки - это два разных типа ссылок в Linux, которые позволяют создавать несколько имен для одного и того же файла.

Жесткая ссылка (hard link) - это дополнительное имя файла, которое указывает на ту же inode (уникальный идентификатор) в файловой системе, что и оригинальное имя файла.

Жесткие ссылки не имеют отдельной метаданных и считаются полноценными файлами.

Если вы удалите одно имя файла, оно не повлияет на доступность файла через другие имена.

Жесткие ссылки могут быть созданы только в пределах одной файловой системы.

```
ln original_file hard_link
```

## Индексный дескриптор (inode).

Inode (индексный дескриптор) - это структура данных в файловых системах, которая содержит метаданные о файле или каталоге.

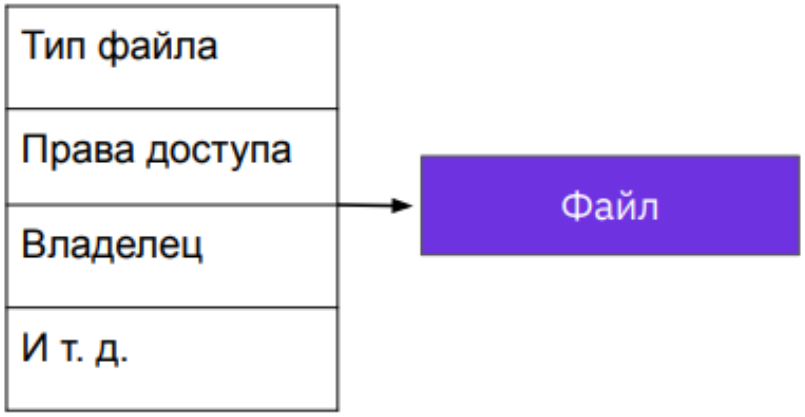
Когда вы создаете файл или каталог в операционной системе, ему автоматически назначается индексный дескриптор (inode).

Он представляет собой своего рода "карточку" файла, где хранится информация о его местонахождении, размере, правах доступа и других свойствах.

Вместо хранения метаданных непосредственно в имени файла, операционная система сохраняет их в соответствующем индексном дескрипторе.

Каждый индексный дескриптор имеет уникальный идентификатор (индекс), который позволяет операционной системе отслеживать и управлять файлами.

# inode и каталог




С каждым файлом в операционной системе Linux связана особая структура данных — индексный дескриптор (inode).

Эти данные хранят метаинформацию о файле: владелец, права доступа, время последнего изменения и т. д.

inode также содержит информацию о физическом расположении данных.

# inode и каталог

Каталог — это файл особого типа, который содержит таблицу соответствия **имя_файла → inode**. В этой таблице требуется уникальность имён, но не уникальность номеров inode. Благодаря этому каждый объект файловой системы может иметь несколько имён. Счётчик имён хранится в inode объекта.

<table>
<tr><th colspan="2">Каталог</th></tr>
<tr><td>file-1</td><td>inode-1</td></tr>
<tr><td>file-2</td><td rowspan="2">inode-2</td></tr>
<tr><td>file-3</td></tr>
<tr><td>file-4</td><td>inode-3</td></tr>
</table>

Символическая ссылка (symbolic link или soft link) - это специальный тип файла, который содержит путь к оригинальному файлу или каталогу.

При обращении к символической ссылке операционная система следует пути, указанному в ссылке, чтобы найти целевой файл или каталог.

Если оригинальный файл или каталог перемещается или переименовывается, символическая ссылка останется недействительной.

Символические ссылки могут указывать на файлы или каталоги, находящиеся в других файловых системах.

```
ln -s target_file symbolic_link
```

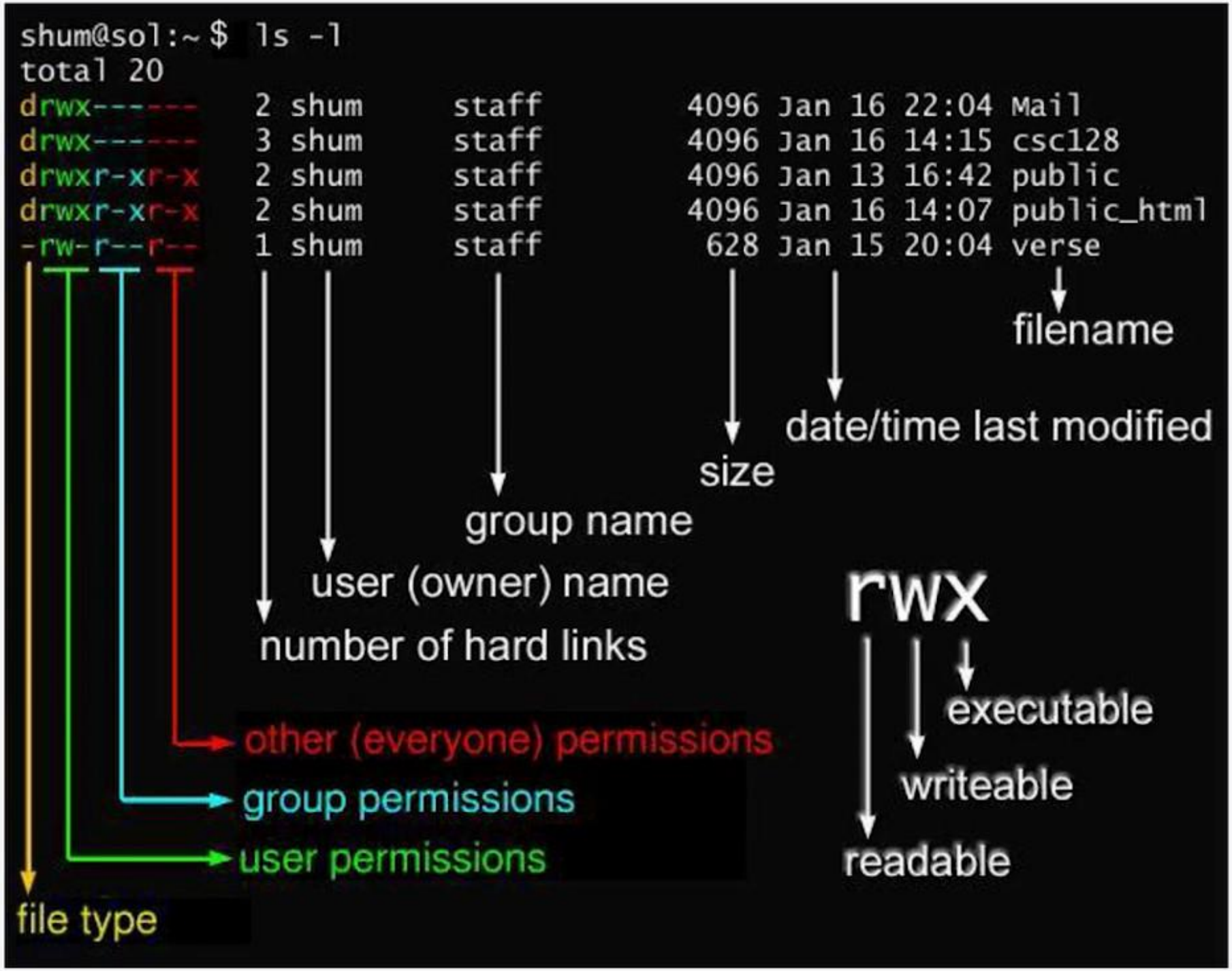
# Жёсткие и символические ссылки

**Ссылки** — это особенность файловой системы, которая позволяет размещать один и тот же файл в разных каталогах.

**Жёсткая ссылка** — это запись в каталоге, указывающая на inode. Создаётся только для файлов, за исключением специальных записей, указывающих на саму директорию (.) и родительскую директорию (..). Жёсткие ссылки используются только в пределах одного раздела.

**Символическая ссылка** — это запись в каталоге, указывающая на имя объекта с другим inode. Наиболее близка к ярлыку в Windows. Она может ссылаться на файл и на каталог. Символические ссылки могут существовать на разных разделах.

# Заключение

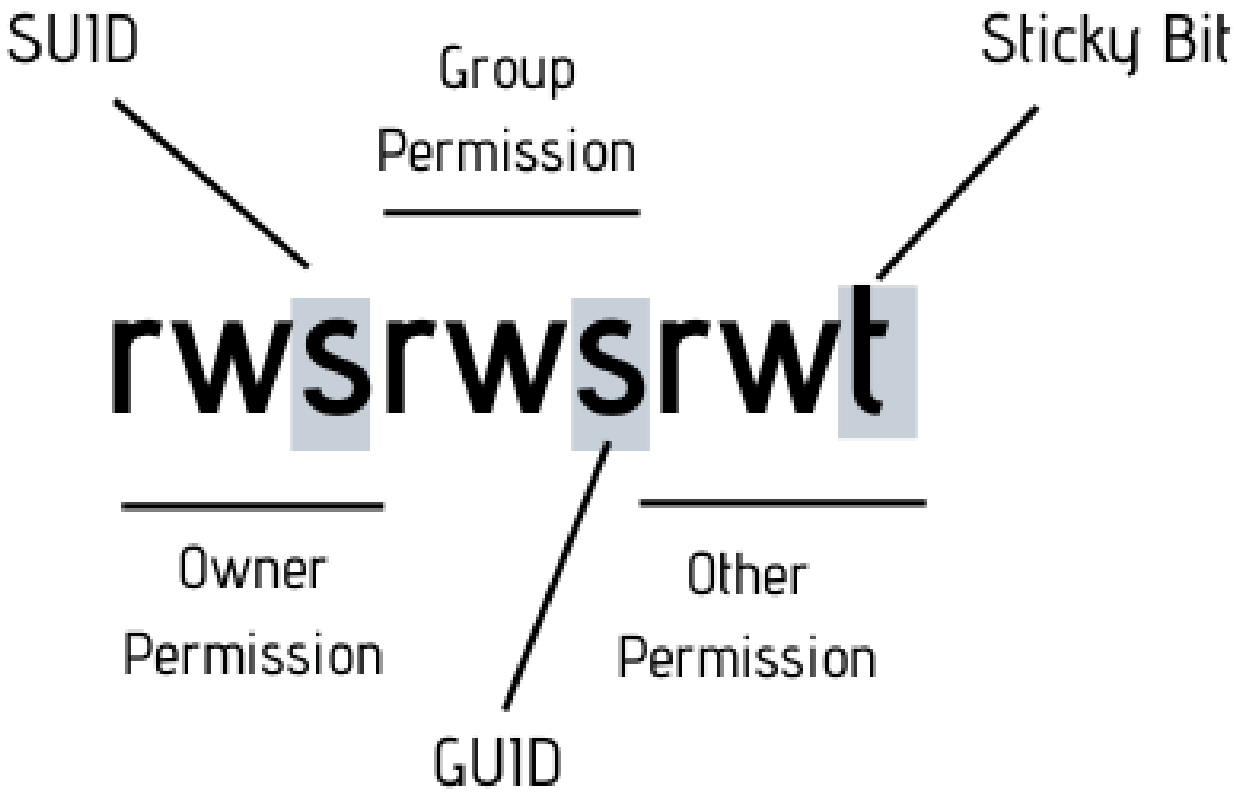

shum@sol:~ $ ls -l
total 20
drwx------ 2 shum staff 4096 Jan 16 22:04 Mail
drwx------ 3 shum staff 4096 Jan 16 14:15 csc128
drwxr-xr-x 2 shum staff 4096 Jan 13 16:42 public
drwxr-xr-x 2 shum staff 4096 Jan 16 14:07 public_html
-rw-r--r-- 1 shum staff 628 Jan 15 20:04 verse
filename
date/time last modified
size
group name
user (owner) name
number of hard links
other (everyone) permissions
group permissions
user permissions
file type
rwx
executable
writeable
readable

# Специальные биты

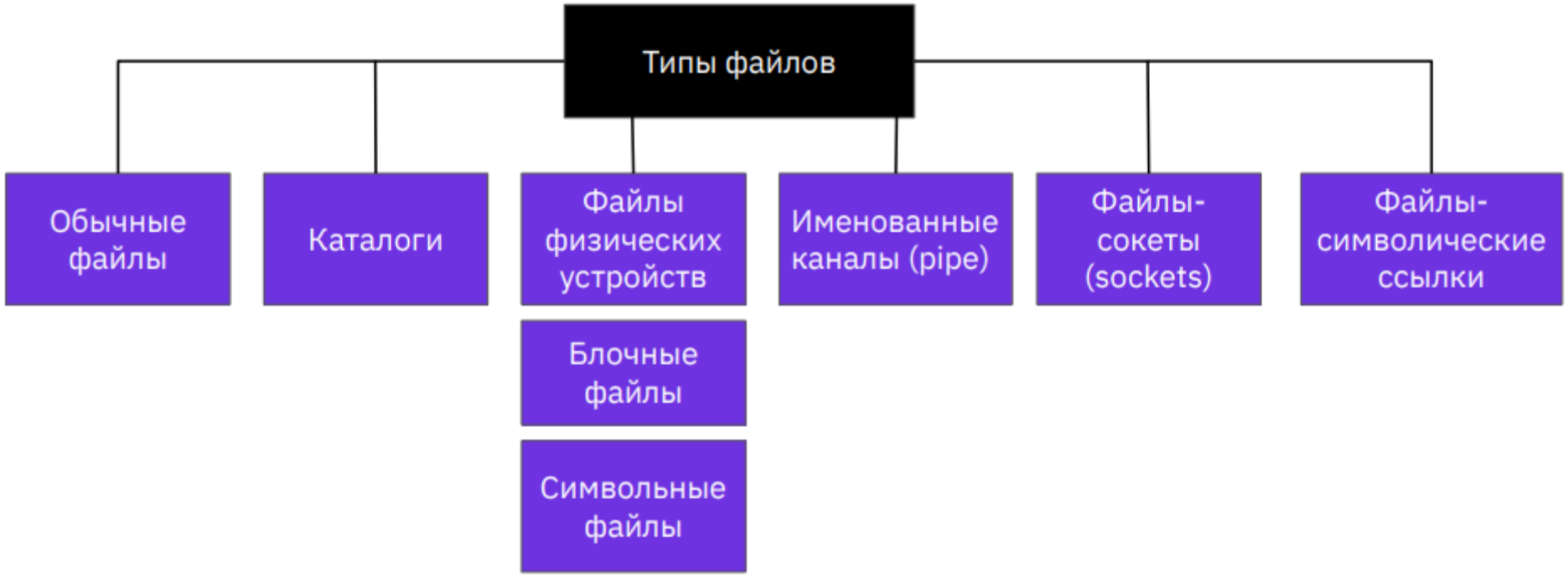
**SUID (set user ID upon execution)** — установка ID пользователя во время выполнения. Разрешает пользователям запускать файл на исполнение с правами того пользователя, которому принадлежит данный файл. SUID работает с файлами.

**SGID (set group ID upon execution)** — установка ID группы во время выполнения, применяется преимущественно к каталогам. Этот атрибут устанавливает идентификатор группы каталога, а не группы владельца, который создал файл в этом каталоге.

**Sticky** — дополнительный атрибут, который устанавливается для каталогов. Файлы из каталога с таким битом может удалить только владелец (пользователь, создавший этот файл).


SUID
Group
Permission
Sticky Bit
rwsrwsrwt
Owner
Permission
GUID
Other
Permission

# Типы файлов в Linux



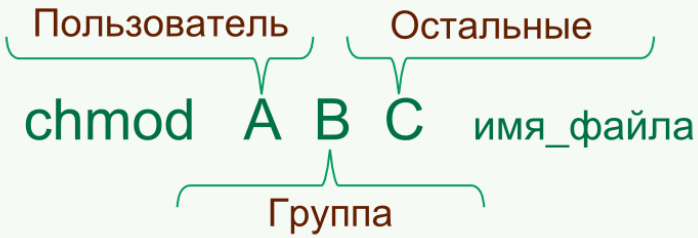
# Команда chmod

## при буквенном обозначении прав


*Примеры:*

| | |
|---|---|
| chmod g+w hello.c | (группе разрешается изменять файл ) |
| chmod a-wx a.out | (всем запрещается изменять и выполнять файл) |
| chmod go=rw docu.odt | (группе и остальным устанавливаются разрешения на чтение и запись) |

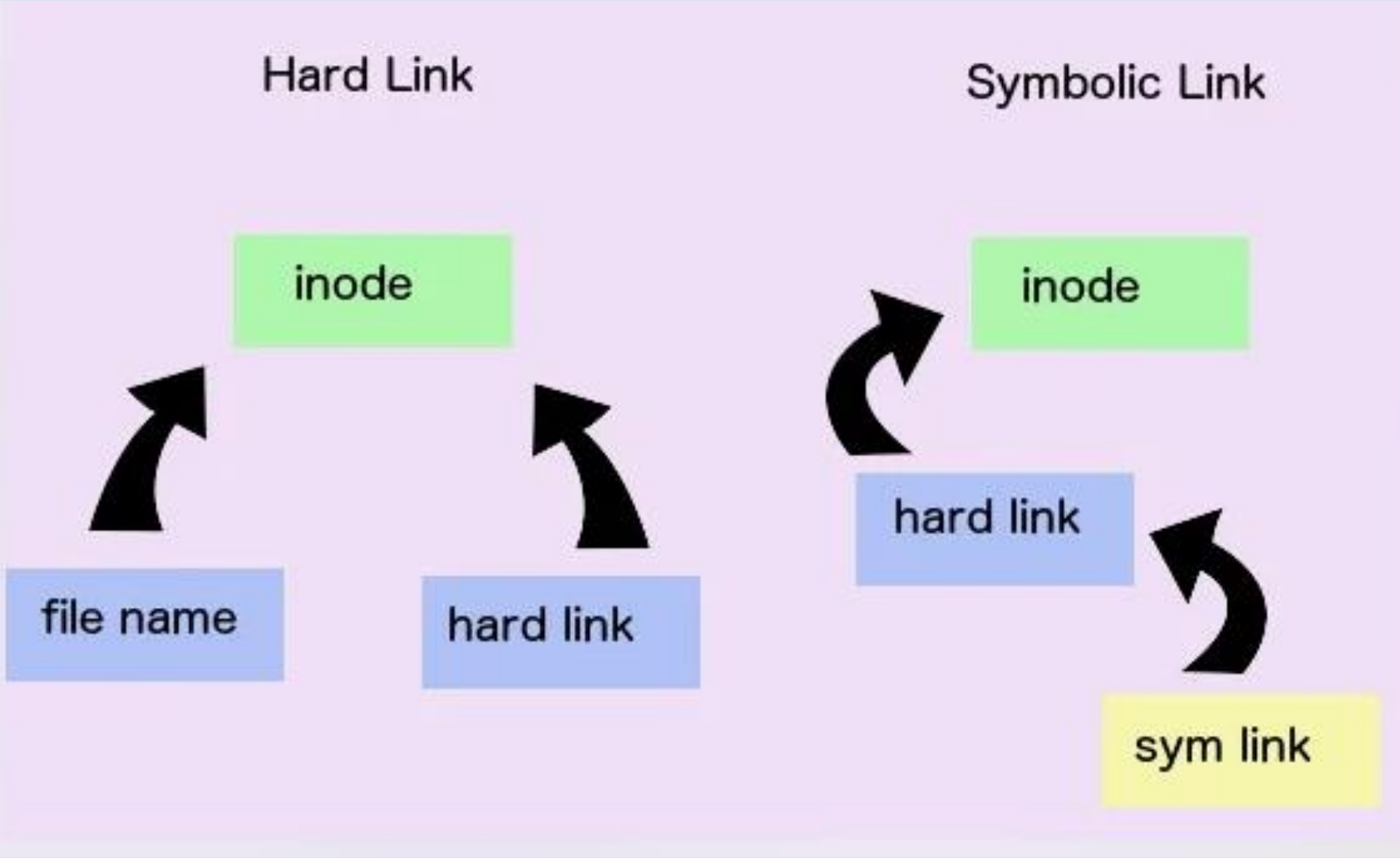
## при числовом обозначении прав




7 - всё разрешено
6 - чтение и запись
5 - чтение и исполнение
4 - только чтение
0 - всё запрещено

*Примеры:*

| | |
|---|---|
| chmod 660 hello.c | (владелец и группа могут читать и изменять файл) |
| chmod 555 a.out | (чтение и исполнение для всех) |
| chmod 777 docu.odt | (все могут всё) |


Hard Link
inode
file name
hard link
Symbolic Link
inode
hard link
sym link

# Домашнее задание.

1. Изучить методичку.
2. Самостоятельно изучить команды: chgrp, umask