Защита на файловете в Unix-like операционни системи

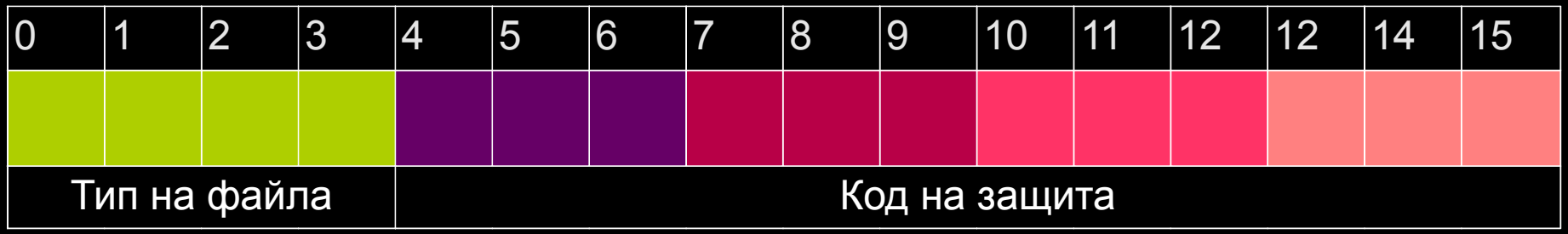
```
struct ext3_inode {
    __le16  i_mode;         /* File mode */
    __le16  i_uid;          /* Low 16 bits of Owner Uid */
    __le32  i_size;         /* Size in bytes */
    __le32  i_atime;        /* Access time */
    __le32  i_ctime;        /* Creation time */
    __le32  i_mtime;        /* Modification time */
    __le32  i_dtime;        /* Deletion Time */
    __le16  i_gid;          /* Low 16 bits of Group Id */
    __le16  i_links_count;  /* Links count */
    __le32  i_blocks;       /* Blocks count */
    __le32  i_flags;        /* File flags */

    __le32  i_block[EXT3_N_BLOCKS]; /* Pointers to blocks */


};
/* linux-2.6.29 */
```

i_mode (mode дума): първото ниво на защита

16 бита:



i_mode

В първите 4 бита се кодира типа на файла:

- 0001 (001_8) - FIFO
- 0010 (002_8) - Character device
- 0100 (004_8) - Directory
- 0110 (006_8) - Block device
- 1000 (010_8) - Regular
- 1010 (012_8) - Symbolic link
- 1100 (014_8) - Socket

Права за достъп до файл

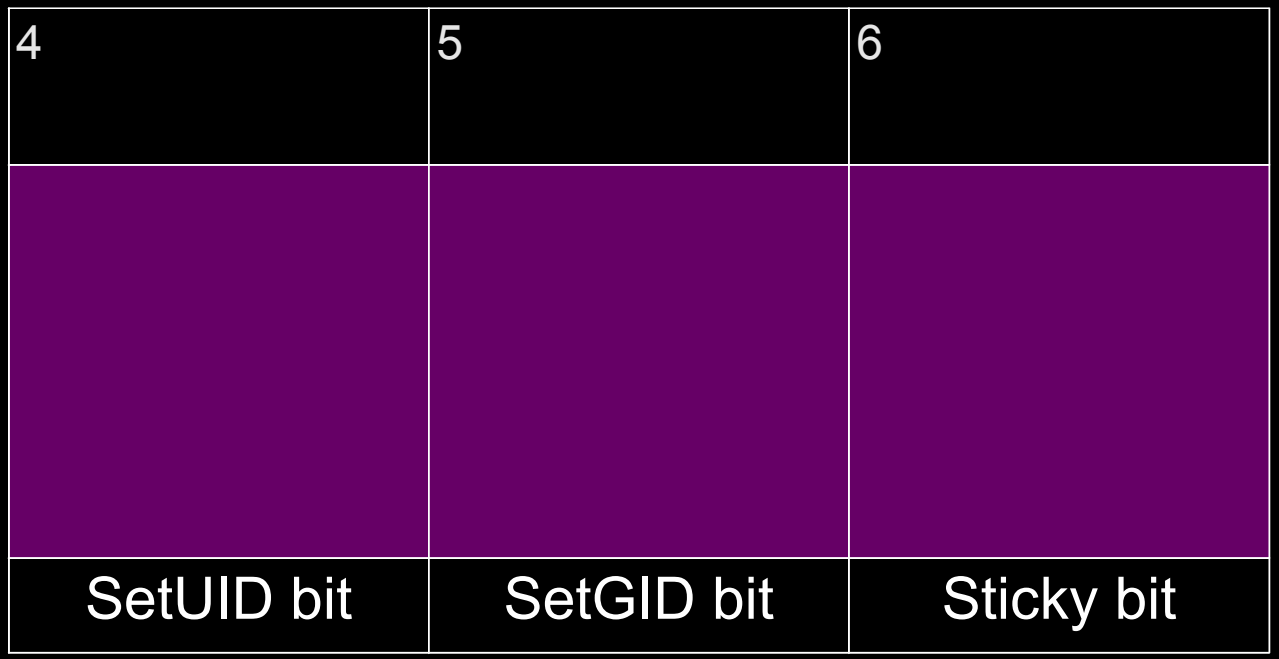
- Видовете права за достъп до файл са read (за четене от файла), write (за писане във файла) и execute.
  - Execute правото за обикновени файлове дава разрешение те да бъдат изпълнявани, т.е. да бъдат подавани като първия аргумент на системните примитиви от exec семейството
  - Execute правото за директории дава разрешение за позициониране в тях и търсене на файлове в тях.
- Всеки файл различава 4 основни вида потребители: администраторът, потребител-собственик, група-собственик и други потребители.
  - Администраторът има неограничени права за достъп до файловата система.

i_mode

В последните 9 бита се кодират правата r,w,x за достъп на потребителя-собственик, групата-собственик и останалите потребители.
Там, където битът има стойност 1, съответното право е дадено (вдигнато), а където има стойност 0 е отнето (свалено).

7	8	9	10	11	12	12	14	15
1	1	1	1	0	1	1	0	0
r	w	x	r	w	x	r	w	x
owning user			owning group			others		

i_mode



i_mode

В битове 4 и 5 се кодират две възможности, свързани с изпълнението на файла – SetUID и SetGID.

С всеки процес са свързани два потребителски и два групови идентификатора: реални – ruid, rgid и ефективни – euid, egid. Реалните са идентификатори на потребителя/групата, който е създал процеса, а по ефективните се определят правата на процеса при работа с файлове, при изпращане на сигнали и др.

Когато се стартира процес, ако не е вдигнат SetUID или SetGID бита на executable файла му, euid=ruid и egid=rgid, т.е. процесът има права за достъп като потребителя, който го е създал. В хода на изпълнение, значенията на euid и egid могат да бъдат изменени със системни примитиви (значенията на ruid и rgid, разбира се, не се променят никога).

Ако при стартиране на процес executable файлът му има вдигнат например SetGID бит, то egid на процеса ще бъде идентификаторът на групата-собственик на executable файла и процесът ще има нейните групови права за достъп, а не тези на създателя си.

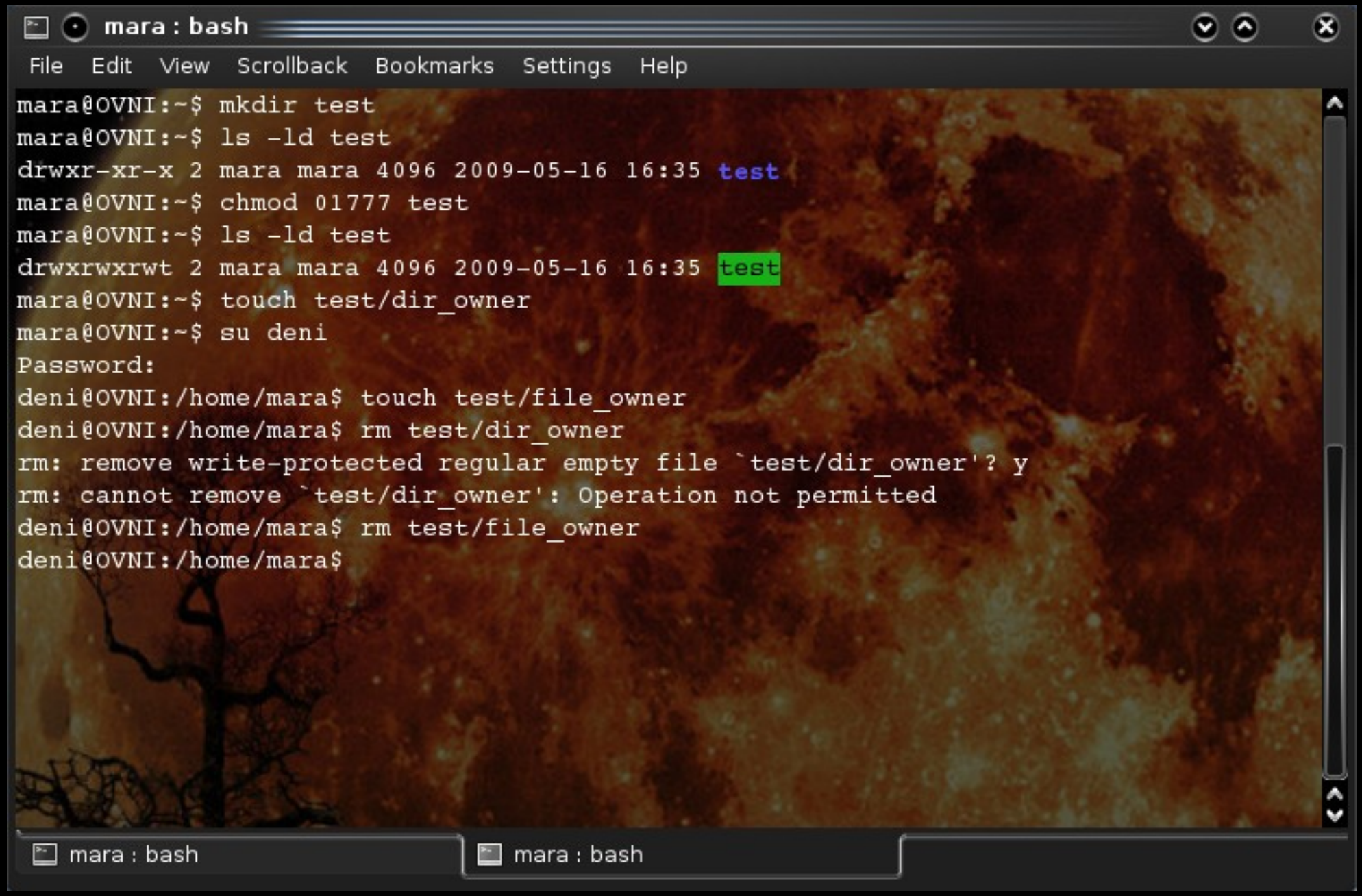
Пример за SetUID програма е passwd.

i_mode

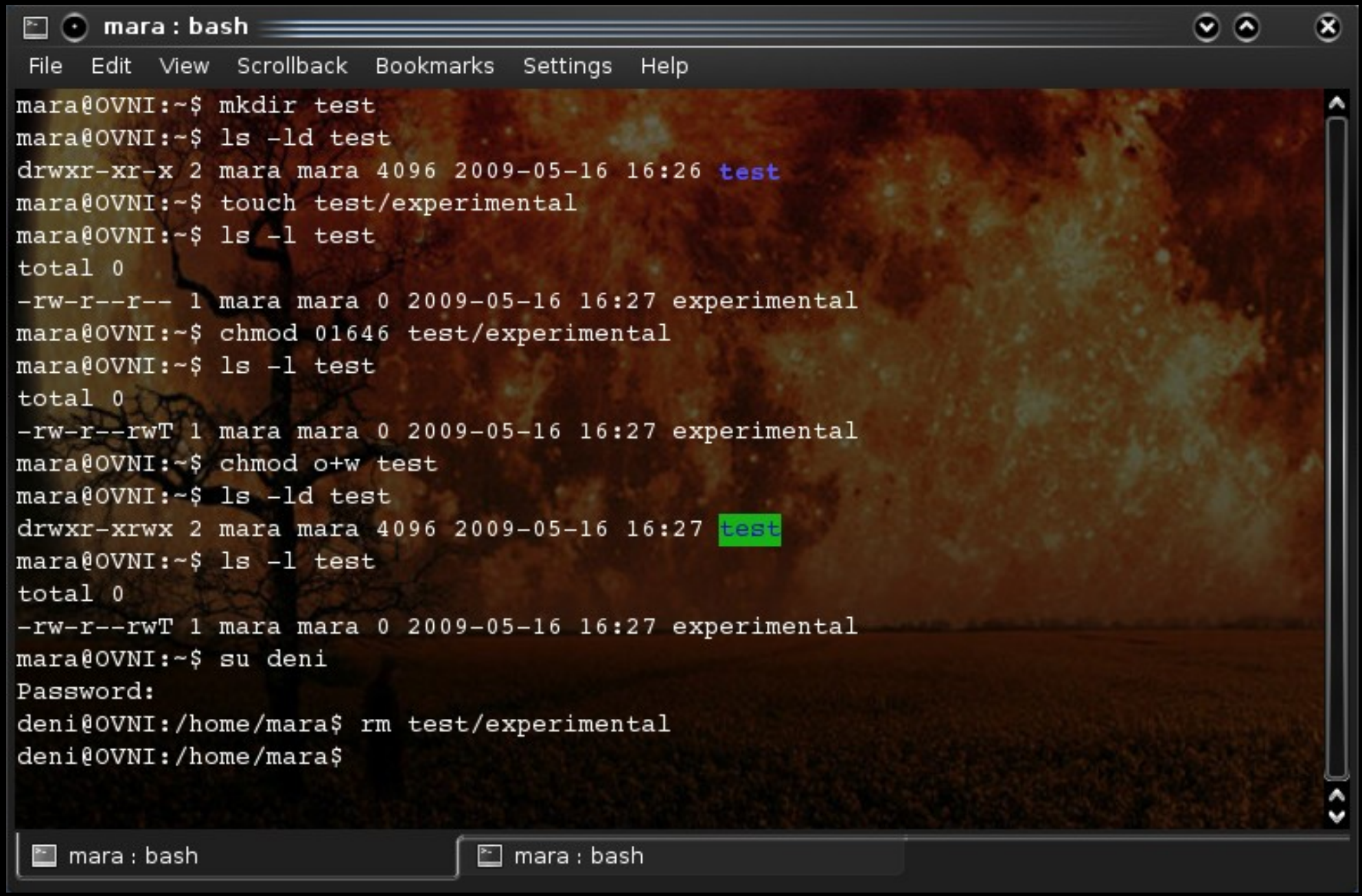
В 6-тия бит, наречен sticky bit, се кодира опция, налагаща ограничения върху правата за изтриване на файлове.

Ако за някоя директория този бит е вдигнат, то даден потребител (процес), за да изтрие даден файл от тази директория, освен задължителното право w за нея (и право x за всички директории от пълния път до нея), трябва или да е администратор, или да е собственика на каталога, или да е собственика на файла.

Пример за директория, за която е вдигнат sticky bit-а е /tmp.



Sticky bit-ът на директорията е вдигнат. Макар и всички потребители да могат да пишат и да се позиционират в нея, файловете в нея могат да се изтриват само от собственика й или от собствениците си.



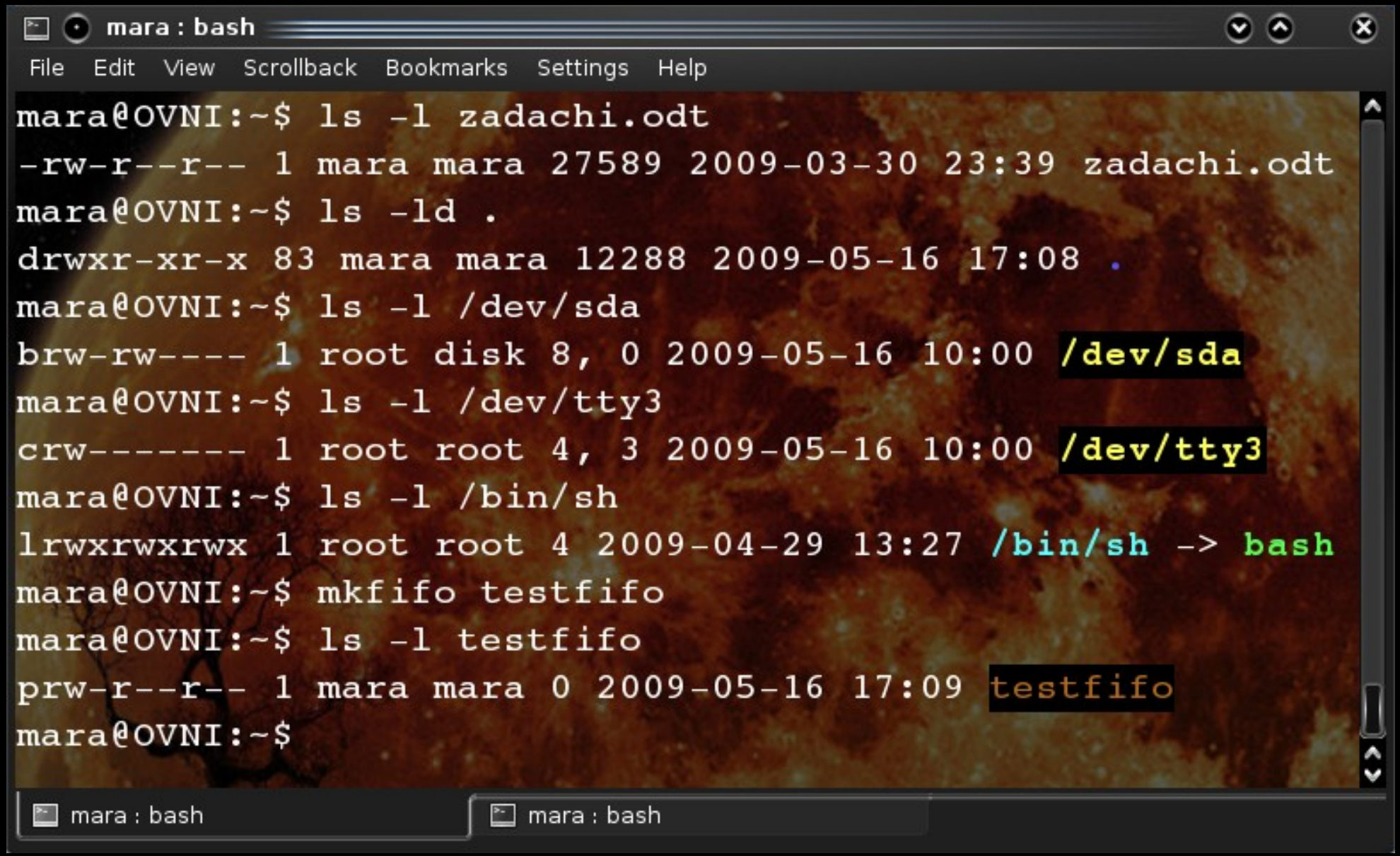
Вдигането на sticky bit-а само за файл, не и за родителския му каталог, не променя стандартните права за достъп, нужни за изтриването му

`ls` и mode-думата на файл

Изходът на командата **`ls -l`** ни дава разширена информация за посочените файлове.

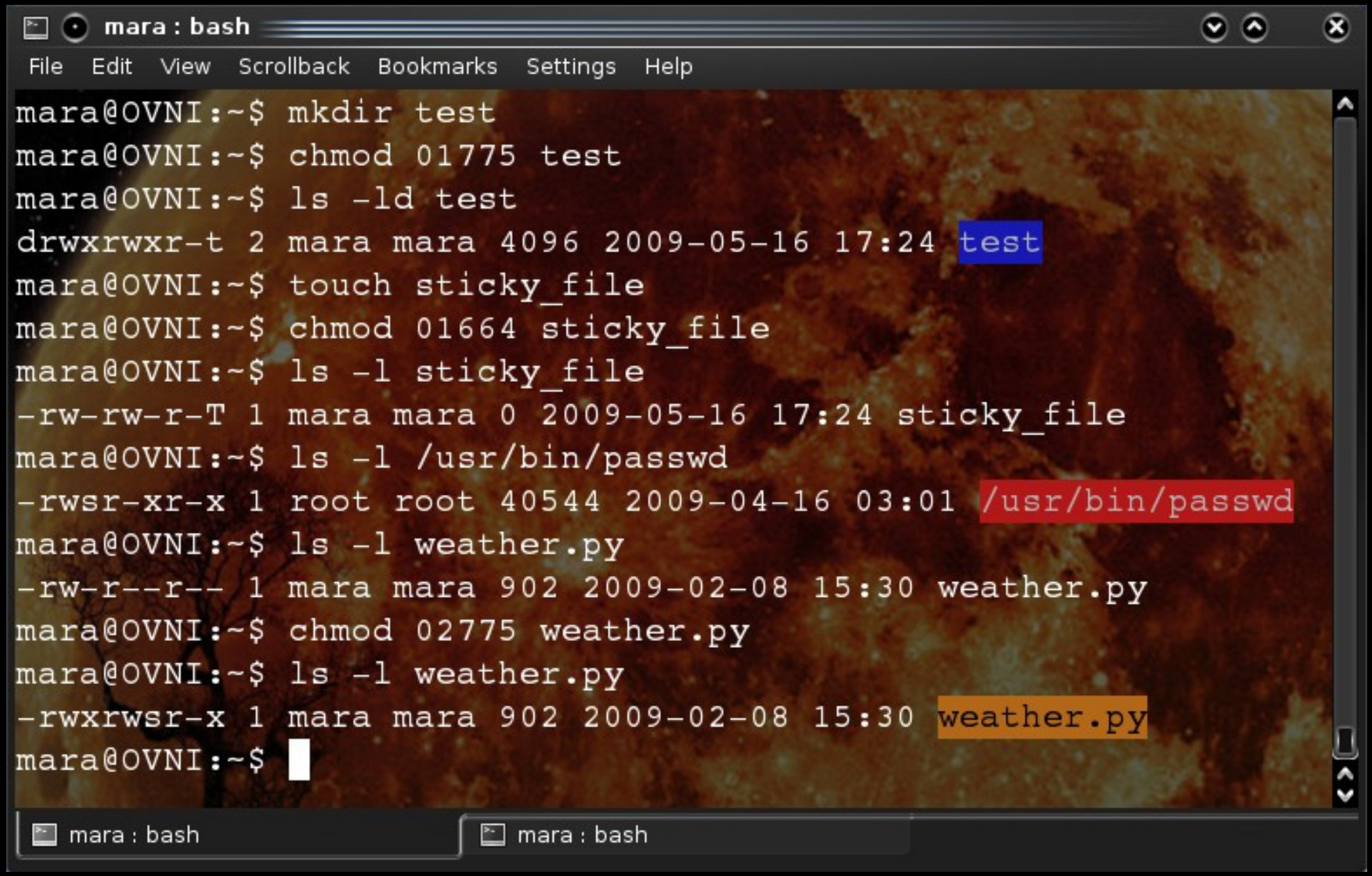
В първата колонка е указано съдържанието на mode-думата на файла – типът му и правата за достъп на собственика, групата-собственик и останалите потребители.

В третата и четвъртата колонка са имената (идентификаторите) на собственика и групата-собственик.



Първият знак указва типа на файла:

- за обикновен файл	d за директория
b за block device	c за character device
l за символна връзка	p за FIFO файл
S за сокет	



Следващите 9 символа показват r, w, x правата съответно на собственика, групата-собственик и останалите потребители. Където правото не е вдигнато, стои -.
Ако е вдигнат SetUID/SetGID/sticky бит, това се отразява с буква s/s/t на мястото на x за собственика/групата/останалите. Ако някоя от тези букви е главна, то значи битът е вдигнат, но това не указва влияиние на правата на достъп (пример: sticky bit за обикновен файл, вдигнат SetGID бит за файл, който няма x право за групата).

Промяна на кода на защита

- **`chmod OCTAL filename`**
  OCTAL е осмично число с 4 цифри. Първата цифра съответства на SetUID, SetGID, sticky тройката битове, втората на битовете за r, w, x на собственика, третата за битовете r, w, x на групата, четвъртата за битовете r, w, x за останалите.

setUID	setGID	sticky	usr r	usr w	usr x	grp r	grp w	grp x	oth r	oth w	oth x
0	0	1	1	1	1	1	0	1	1	0	0
1			7			5			4		

Промяна на кода на защита

- **`chmod [ugoa...][[+-=][perms...]...] file`**
  - u=user, g=group, o=others, a=all
  - +: добавяне на право, -: отменяне на право, =: приравняване на право към друго
  - perms
    - 0 или повече букви измежду rwxXst
    - Точно една буква измежду ugo.
  - Чрез разделяне със запетайки може да се укажат няколко израза

Примери

```
mara@OVNI:~/test$ ls -l
total 0
-rwxr-xr-x 1 mara mara 0 2009-05-16 18:10 a
-rw-rw-r-x 1 mara mara 0 2009-05-16 18:10 b
-rw-r--r-- 1 mara mara 0 2009-05-16 18:10 c
-rw-r--r-- 1 mara mara 0 2009-05-16 18:10 d
-rw-r--r-- 1 mara mara 0 2009-05-16 18:11 e
-rw-r--r-- 1 mara mara 0 2009-05-16 18:11 h
mara@OVNI:~/test$ chmod a+x c
mara@OVNI:~/test$ chmod u+rwx b
mara@OVNI:~/test$ chmod ug+rwx,o-r h
mara@OVNI:~/test$ chmod g=o a
mara@OVNI:~/test$ chmod a-rwx d
mara@OVNI:~/test$ chmod u-rwx,g-rwx,o+rwx e
mara@OVNI:~/test$ ls -l
total 0
-rwxr-xr-x 1 mara mara 0 2009-05-16 18:10 a
-rwxrw-r-x 1 mara mara 0 2009-05-16 18:10 b
-rwxr-xr-x 1 mara mara 0 2009-05-16 18:10 c
---------- 1 mara mara 0 2009-05-16 18:10 d
-------rwx 1 mara mara 0 2009-05-16 18:11 e
-rwxrwx--- 1 mara mara 0 2009-05-16 18:11 h
mara@OVNI:~/test$
```

Проверка за правата на достъп на процес до файл

1) Ако процесът е с правата на root, достъпът се разрешава.
2) Ако процесът е с правата на собственика на файла
   - ако в кода на защита на файла бита за съответния тип достъп за собственика е вдигнат, достъпът се разрешава
   - иначе не се разрешава.
3) Ако процесът е с правата на групата на файла
   - ако в кода на защита на файла бита за съответния тип достъп за групата е вдигнат, достъпът се разрешава
   - иначе не се разрешава.
4) Ако в кода на защита на файла бита за съответния тип достъп за другите е вдигнат, достъпът се разрешава, иначе не се разрешава.

По-високи нива на защита

- Extended файловата система поддържа флагове в от полето *i_flags* на структурата inode. Някои от тези флагове са: *immutable*, *append_only*, *synchronous_write*, *secure_delete*, *undelete*, *compress_file*. Командата **`lsattr`** показва кои от тях са вдигнати за файла. Могат да се променят с командата **`chattr`**.
- Механизми Extended File Attributes и Access Control List