

Файлове и директории и Права за контрол на достъпа

07 октомври 2020 г. 13:04

Файлове и директории

- Блокове, inodes, връзки (hard и symbolic links);
- Намиране на файлове (find, locate, which, whereis);

```
nick@fly:~$ ln -s
```

Блокове, inodes, връзки (hard и symbolic links):

Links: Ефективно, пряк път от един файл в друг, малък файл (съдържащ свой собствен inode и път към оригиналния файл);

Може да „обхваща“ файловите системи и дискове (отдалечени или локални), защото има независим inode.

Symbolic links: Още наречени (soft), Разрешенията не могат да бъдат променени, промените ще бъдат (опитани да бъдат) приложени към оригинала (при условие че са съществували правилните разрешения за потребителя, изпълняващ командата), премахване на символна връзка (понякога наричана "мека" връзка), премахва само малкия файл и свързан инод и не засяга оригинала.

Премахването на оригиналния файл НЕ премахва връзката, а оставя връзката „осиротяла“.

```
nick@fly:~$ ln -P
```

Hard links:

- Добавяне на допълнително име към файл или директория, които всъщност сочат към оригиналният файл или директории inode (всъщност това са същите данни на диска).
- Изтриването на която и да е една от съществуващите твърди връзки просто изтрива препратката към inode, стига съществува всяка друга твърда връзка (или оригиналната), данните в inode остават
- В резултат на това, че всяка твърда връзка споделя един и същ inode, обаче, твърдите връзки НЕ могат да се „обхващат“ навсякъде файлови системи или дискове (локални или отдалечени)
- Напълно изтриването на файл с твърди връзки изисква премахване на ВСИЧКИ твърди връзки
- Твърдите връзки са полезни, за да направят много важни файлове "постоянни" (т.е. поставяне на твърда връзка в Администратор, достъпен само за директория до директория или файл, който никога не трябва да бъде напълно изтрит)

- Например - `ln -s myscript.sh yourscript.sh`
- Създава символична връзка, наречена „yourscript.sh“, насочена към „myscript.sh“

- Например - `ln myscript.sh tvojencrypt.sh`
- Създава твърда връзка (по подразбиране), наречена точка 'tvojencrypt.sh' към данните в посочената inode до от 'myscript.sh'
- `ls -li` [име на файл] - може да се използва за намиране на всички връзки за даден файл

Блокове:

При изчисляването блокът, понякога наричан физически запис, е последователност от байтове или битове, обикновено съдържаща цял брой записи, имащи максимална дължина; размер на блок. Така структурирани данни се казва, че са блокирани (Blocked) .

Inodes:

Inode е структура на данни във файлова система в стил Unix, която описва обект на файлова система като файл или директория. Всеки inode съхранява атрибутите и дисковите блокови местоположения на данните на обекта.

Права за контрол на достъпа



Правата се разпределят по следният начин:

Символни

r = read permission
w = write permission
x = execute permission
- = no permission

Октанли (цифрени)

4 = read
2 = write
1 = execute
0 = no permission

```
nick@fly:~$ ls -l
total 0
drwxrwxrwx 1 nick nick 4096 Jul 30 15:16 permissions
```

- Първият знак, посочва, че това е директория.
- Първата група от три знака показва потребителя (user) който кореспондира на nick в първия случай.
- Втората група показва правата на групата, така и ни казва чия е групата
- Третата е на всички останали.
- Примера показва: Папка на потребителя, върху която той има пълни права, както и потребителите в групата nick имат същите права върху нея (пълни), останалите хора (потребители) могат да променят чета и екзекутират също. Тоест файла е на всички.

```
nick@fly:~$ ls -l permissions.txt
-rw-rw-rw- 1 nick nick 0 Jul 30 15:23 permissions.txt
```

- - показва, че това е файл, включва скриптове, байнарита, архиви и др.
- b показва блоково устройство (хард дискове и дискове)
- с периферни устройства.

Команди за промяна на достъпа:

- chown : Променя собствеността на файл или директория
- chmod :groupname file.extension
- chmod : Променя вида на файла, променяйки неговите права.
- chmod u(user) g(group) o(others) | rwx|
- chgrp : Променя групата собственик на файл.

SUID

SUID: set user id bit. Файлове които съдържат 's' на мястото на 'x'

в колоната на потребителя. Имайте в предвид, че тази опция вече не работи на баш скриптове. Също много файлови системи могат да бъдат маунтнати с 'nosuid' option.

Някои програми го имат. Като например passwd.

Премахнато е с цел повече security.

SGUID

Файлове и папки със "s" в групата на мястото на "x".

Полезно за споделени папки по група.

```
nick@fly:~$ chmod 760 test.sh
nick@fly:~$ ls -l
-rwxrw---- 1 nick nick 3 Jul 30 16:43 test.sh
nick@fly:~$ chmod 0764 test.sh
nick@fly:~$ ls -l
-rwxrw-r-- 1 nick nick 3 Jul 30 16:43 test.sh
nick@fly:~$ chmod 4764 test.sh
nick@fly:~$ ls -l
-rwsrw-r-- 1 nick nick 3 Jul 30 16:43 test.sh
nick@fly:~$ chmod 0764 test.sh
nick@fly:~$ ls -l
-rwxrw-r-- 1 nick nick 3 Jul 30 16:43 test.sh
nick@fly:~$ chmod 760 test.sh
```

```

nick@fly:~$ ls -l
-rwxrw---- 1 nick nick 3 Jul 30 16:43 test.sh
nick@fly:~$ chmod 0764 test.sh
nick@fly:~$ ls -l
-rwxrw-r-- 1 nick nick 3 Jul 30 16:43 test.sh
nick@fly:~$ chmod 4764 test.sh
nick@fly:~$ ls -l
-rwsrw-r-- 1 nick nick 3 Jul 30 16:43 test.sh
nick@fly:~$ chmod 0764 test.sh
nick@fly:~$ ls -l
-rwxrw-r-- 1 nick nick 3 Jul 30 16:43 test.sh

```

- groups: показва групите в които е един user
- groupadd nameofgroup –добавя група
- usermod –G group username добавяте user в група
- sudo chown username:group directory

1. su – root
2. mkdir team
3. groupadd team
4. usermod –G team username
5. useradd test
6. usermod –G team test
7. touch with
8. chown username:group directory
9. chmod -R 2770 team/

Sticky bit

Тази пермисия 't' позволява да може само създателя на файла да го изтрие.

1. mkdir /srv/sticky
2. touch /srv/sticky/important
3. su test
4. chmod 1777 /srv/sticky

```

nick@fly:~$ umask
0002
nick@fly:~$ expr 777 - 002
775
nick@fly:~$ expr 666 - 002
664
nick@fly:~$ mkdir mask.d
nick@fly:~$ ls -l
total 0
drwxrws--- 1 root team 4096 Jul 30 17:36 group
drwxrwxr-x 1 nick nick 4096 Jul 30 17:49 mask.d
drwxr-xr-x 1 root root 4096 Jul 30 17:28 perm
nick@fly:~$ touch mask.file
nick@fly:~$ ls -l
total 0
drwxrws--- 1 root team 4096 Jul 30 17:36 group

```

```
drwxrwxr-x 1 nick nick 4096 Jul 30 17:49 mask.d
-rw-rw-r-- 1 nick nick   0 Jul 30 17:49 mask.file
drwxr-xr-x 1 root root 4096 Jul 30 17:28 perm
```

- umask – показва текущите umask настройки
- ~/.bashrc, може да бъде приложен върху всички потребители
- Попринцип са 777 за папки, 666 за файлове
- Ако обаче си user вадиш от 777-002 = 775, 666-002 = 664
- umask стойност = изваждаш стойността на default за да получиш текущо използваните.
- Настройките са зададени от /etc/bashrc файла за цялата система
- Настройките зададени от /etc/user/.bashrc е индивидуално за целият user.
- Метод за модифициране на обикновенните пермисии приложени върху файлове и директории които един потребител създава.

```
if [ $UID -gt 199 ] && [ "`/usr/bin/id -gn`" = "`/usr/bin/id -un`" ]; then
    umask 002
else
    umask 022
fi
```

Ролята на файловата система в линукс :

Когато данните на един компютър се запазят на записващото устройство по зададен метод:

- Данните са организирани и лесно откриваеми.
- Данните се съхраняват по устойчив метод.
- Целостта на данните се запазва.
- Данните лесно могат да бъдат достъпени от потребителя по всяко време.

Директорна структура:

- обърната дървовидна структура с един корен (боаб)
- Разграничението на малки от големи букви (case sensitive)
- Всяка папка или файл, чието име започва с . (точка) е скрита от нормалният „поглед“
- Просто . В папката (директорията) обозначава текущата директория.
- Две .. Са предходната (една папка назад или нагоре към първият клон)

Команди с цел откриване и локализиране:

- locate : комадна която търси в локаната база данни за папки и файлове които съвпадат определен критерии.
- updatedb : опреснява локаната база с данни която locate ползва.
- whereis : това е команда която локализира бинарките, източници и наръчници за команда.