4. Права за контрол на достъпа  
• Собственост на файлове ( chown , chgrp );  
• Права на файлове и директории (umask, chmod, SUID, SGID, Sticky bit);

Част 4

Права за контрол на достъпа

• Собственост на файлове ( chown , chgrp );  
• Права на файлове и директории (umask, chmod, SUID, SGID, Sticky bit);

# **Контролен списък за достъп**

Списъкът за контрол на достъпа (ACL) по отношение на компютърна файлова система е списък с разрешения, прикрепени към обект. ACL определя кои потребители или системни процеси имат достъп до обекти, както и какви операции се разрешават на дадени обекти. Всеки запис в типичен ACL определя предмет и операция. Например, ако даден файлов обект има ACL, който съдържа (Алис: четене, запис; Боб: прочетете), това ще даде на Алис разрешение да чете и записва файла и Боб да го чете само.

Много видове операционни системи въвеждат ACL или имат историческа реализация.

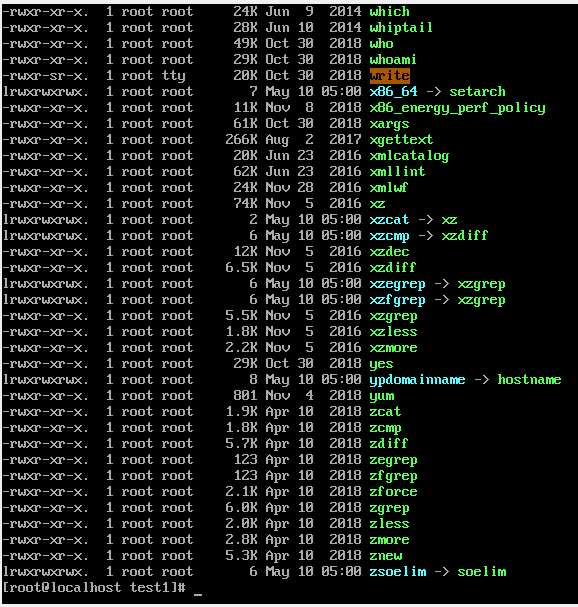
ACL на файловата система

ACL на файлова система е структура от данни (обикновено таблица), съдържаща записи, които определят индивидуалните потребителски или групови права за определени системни обекти, като програми, процеси или файлове. Тези записи са известни като записи за контрол на достъпа (ACE) в Microsoft Windows NT, OpenVMS, Unix-подобни и Mac OS X операционни системи. Всеки достъпен обект съдържа идентификатор за неговия ACL. Привилегиите или разрешенията определят специфични права за достъп, като например дали потребителят може да чете, пише или изпълнява обект. В някои реализации ACE може да контролира дали потребителят или група потребители могат да променят ACL върху даден обект.

Работна група POSIX 1003.1e / 1003.2c направи опит да стандартизира ACL, и POSIX 1003.1e проект 17 ACL са достъпни от ядрото на Linux версия 2.5.46 през ноември 2002 г.

Повечето от Unix и Unix-подобни операционни системи (напр. Linux,BSD или Solaris) поддържат ACL от POSIX.1e, базирани на ранен проект на POSIX, който беше оттеглен през 1997 г. Много от тях, например AIX, FreeBSD, Mac OS X, започващ с версия 10.4 ("Tiger") или Solaris с файлова система ZFS, поддържат ACL от NFSv4, които са част от стандарта NFSv4. Има две експериментални реализации на ACL от NFSv4 за Linux: ACLs NFSv4 за файлова система Ext3 и по-новите Richacls, които поддържат NFSv4 ACL поддръжка за файлова система Ext4.

За да стане по ясно ще илюстрираме в нашата линукс работна машина за какво ще си говориме



лалтп забелязвате от ляво има едни букви с – между тях например –rwxr-xr-x. Това са правата на файлът ( след малко ще ги обясниме по подробно както и ка кда ги променяме според нуждите си ).

Във втората колона със номер е обозначен броят на хард линковете а третата и четвъртата колкона ( root root ) е собственикът на файлът .

За начало ще се научим да сменяме собственикът и групата на даден файл с командите chown , chgrp

Простото обяснение на chown  е change owner (смени собственика )

А на chgrp change group ( смени групата)

Но първо да се запознаем с правата за които споменахме по нагоре или какво означават rxw..

Всяка Linux операционна система има определена структура на директориите и всяка директория има определена функция. **В Линукс има два типа директории: системни и потребителски.** Системни са тези директории, които се създават при инсталирането на операционната система като например директориите /dev, /bin, /lib и т.н. Потребителски са директориите, които се създават от потребителите на операционната система с цел да пазят собствена информация в тях. Такава е директорията /home/user и всички поддиректории вътре в нея.

Много важно място в структурата на операционната система заемат **правата на директориите и файловете.** От тях **зависи сигурността на цялата система** и дали даден потребител може да изпълнява скриптове или да чете файлове. В повечето директории обикновеният потребител има права само за четене, но не и за изпълнение и запис. В някои от директориите като /root например обикновените потребители нямат никакви права. Те имат пълни права на файловете единствено в собствените си директории, благодарение на което е невъзможно инсталирането и изпълнението на зловреден софтуер, който може да навреди на цялата операционна система. В този случай вирусите биха навредили единствено на самия потребител, който ги изпълнява.

## ВИДОВЕ ПРАВА НА ДОСТЪП ПРИ ОПЕРАЦИОННИТЕ СИСТЕМИ

Сега е моментът да уточним какво е това право на потребителя, видовете права и как се задават.

Всеки файл има 3 вида права за потребителя, 3 за групата и 3 за всички останали. Тези права са:

* право за четене (r) - read
* право за запис (w) - write
* право за изпълнение (х) - execute

В графичен вид изглежда по следния начин:

**Собственик** |**Група** |**Всички останали**| r |w |x |r |w |x |r |w |x| | | | | | | | ||

Правата на всеки потребител може да се видят с командата **ls –l**

drwxr-x--x 11 dani dani 4096 Apr 22 21:35 mail

dr-xr-xr-x 2 dani dani 4096 Jun 7 17:42 perl5

-rw-r----- 1 dani users 739 May 9 2008 php.ini

drwxr-xr-x 5 dani dani 4096 Apr 21 12:45 public\_ftp

drwxr-x--- 13 dani nobody 4096 May 31 12:04 public\_html

drwxr-xr-x 8 dani dani 4096 Apr 22 21:20 tmp

drwx------ 2 dani dani 4096 Apr 18 12:02 work

-rwxr-xr-x 2 root root 4096 Apr 18 12:02 .htaccess

Буквата “d” в началото обозначава директориите, а ”-” – същинските файлове.

В горния пример собственик на файла php.ini е „dani”, като той може да чете и променя този файл. Групата на файла е „users” и всички от тази група могат да четат файла, но нищо друго. Всички останали нямат никакви права.

В Linux операционна система правата се задават със следната команда:

**chmod 755 php.ini**

За улеснение на потребителите всички [хостинг доставчици](https://www.icn.bg/bg/shared-hosting/" \t "_blank) използват някакъв вид контролен панел, през който също има възможност да се променят правата на файловете.

Друг начин за промяна правата е чрез FTP клиент, където също има такава опция.

**Всяка цифра отговаря на определени права:**

0 - Няма право да чете, няма право да пише, няма право да изпълнява

1 - Няма право да чете, няма право да пише, може да изпълнява --x

2 - Няма право да чете, може да пише, не може да изпълнява -w-

3 - Няма право да чете, може да пише и изпълнява -wx

4 - Има право да чете, не може да пише и няма право да изпълнява r--

5 - Има право да чете, няма право да пише и може да изпълнява r-x

6 - Има право да чете, има право да пише, не може да изпълнява rw-

7 - Има права за четене, писане и изпълняване rwx

Освен промяна на правата, понякога се налага да се променят и собственика или групата на файла. Това става с командата:

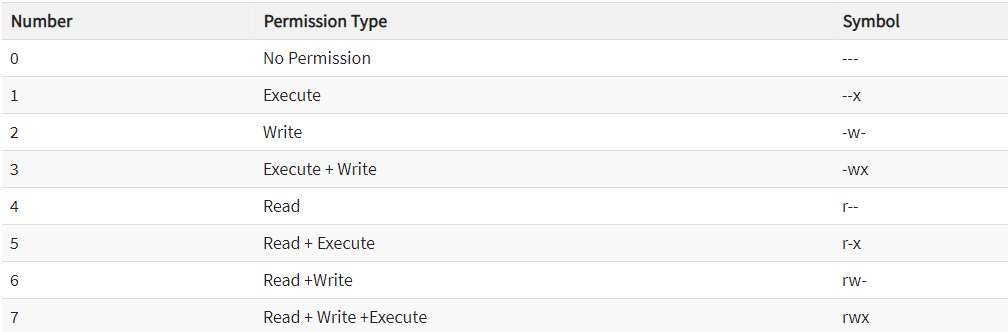
**chown dani:users php.ini**

“dani” е собственикът, а “users” е групата.

**За по-голяма сигурност препоръчителните права са 755 за директории и 644 за файлове.**

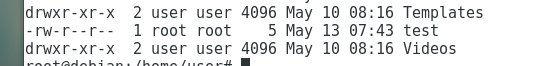
За да намалят възможността от пробиви някои администратори настройват WEB сървъра да дава грешка 500 при задаване на максималните права 777 и когато получите такава грешка първото нещо е да видите дали нямате файлове или директории с подобни права.

Ето и как реално изглеждат въпросните права и как се задават



Да кажем че сме създали файл тест в директорията на user само че с root

Съответно потребителят не може да променя файлът или да го трие мести и т.н



Обаче трябва да му дадем права както се вижда от картинката собственикът е root а потребителят има само права за четене за начало ще направиме нашият потребител собственик на файлът

Командата е chown

А синтаксисът е както споменахме потребител:група

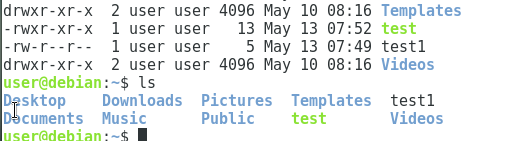
Или в нашият случай chown user:user test



Сега вече нашият потребител е собственик на файла както се вижда от следващата илюстрация



Вече потребителят има права за четене и запис а групата само за четене обаче да кажеме че е някъква програма която трябва да се стартира трявбат и права за срартиране или x права за целта ще изплазваме командата chmod +x test

Командата добавя права за стартиране на файлът както виждате цветът на файлът от черен стана зелен 

Това означава че всеки потребител в група user може да го стартира файлът

Съответно да махнеме правата да се стартира командата е chmod –x test 

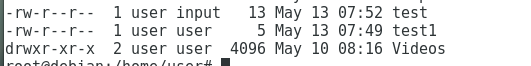
Както виждате правата се промениха на само за четене

От гледна точка на сигурност не правете файловете да бъдат executable освен ако не се налага да бъдат стартирани от потребители .

Съответно за да смениме групата да кажеме че нашият юзър трябва да сподели файлове със шефът си който се намира в група input например

Командата е chgrp input test





Тоест потребители от въпросната група вече могат да четат въпросният файл без да имат достъп до дтугите файлове в папката .

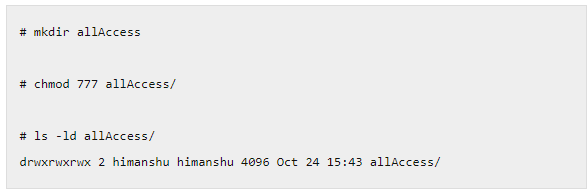
Да кажеме че по погрешка сме дапи права на потребителят пешо да променя файловете в нашата папка съответно да може и да трие . Обаче пешо е решил да напуска фирмата и тъй като е злонамерен а сме работили по този файл от дълго време решава да го изтрие . Както знаеме изтрити файлове в линукс не могат да бъдат възстановени . Има една малко по особенна категория правила освен изброените или така наречените **Sticky Bit**

**Ще кажете е пешо изтри файлът какво толкова от това . Смисълът е точно в това че дори и дадена група потребители да има достъп да редактира файлове в дадена папка само собственикът и root потребителят имат право да трият файлове от дадена папка или самата папка .**

**Ще го илюстрирам в следният пример**

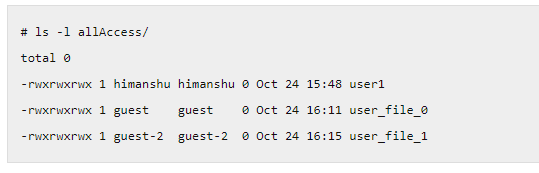
**За целта ще си създадеме нова папка някъде в системата**

**И ще дадеме пълни права на всички да правят всичко в папката**

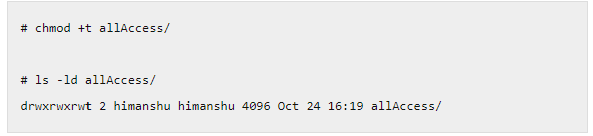


Както виждате всички имат пълни правя върху папката

Да кажем че има файлове в папката както споменах по нагоре файлът върху който работите с пешо и един файл на шефа



В случаят пешо може да изтрие и двата файла за да се предпазиме ще добавим върху папката въпросната протекция



Да кажем че пешо преди да изтрие файла иска да си го копира в неховата хоум директория

Ще получи следното съобщение

$ mv /home/himanshu/allAccess/user\_file\_1 /home/himanshu/allAccess/user\_file\_0

mv: cannot move `/home/himanshu/allAccess/user\_file\_1' to `/home/himanshu/allAccess/user\_file\_0': Operation not permitted

както виждате операцията не е разрешена .

За да махнем въпросната опция е небходимо само да изпълниме същата команда само че с –

