ИКОНОМИЧЕСКИ УНИВЕРСИТЕТ - ВАРНА

КАТЕДРА „ИНФОРМАТИКА”

ОС UNIX

УПРАЖНЕНИЕ 6: РАБОТА С КОМАНДИ, 2 ЧАСТ

Изготвил: гл. ас. д-р Радка Начева

ДАТА: 21 ОКТОМВРИ 2019 Г.

УПРАЖНЕНИЯ 6: РАБОТА С КОМАНДИ, 2 ЧАСТ

I. ЦЕЛ И ТЕМИ НА УПРАЖНЕНИЕТО

Целта на упражнението е да въведе студентите в използването на базови команди в операционни системи Linux, свързани с управление на правата за достъп до файлове и директории, пренасочване на входа и изхода на командите и управление на процеси. В темата са приложени и подходящи примери, демонстриращи практическото им приложение.

Темите[1], засегнати в упражнението, са:

1. Управление на правата за достъп до файлове;
2. Пренасочване на входа и изхода на командите;

Като краен резултат от изпълнение на упражнението се очаква студентите да придобият базови знания и умения относно работа с команди без графична среда и по-конкретно, управление на достъпа до файлове и директории.

[1] ***Забележка**: Някои от поставените теми могат да се разглеждат в теоретичната част на упражнението, а други – в практическата.*

II. ТЕОРЕТИЧНА ПОДГОТОВКА

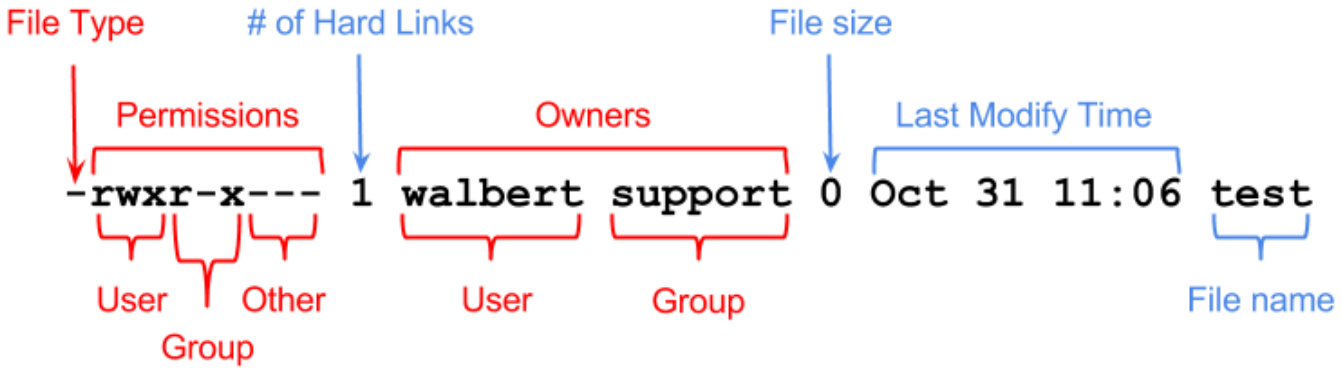
Забележка: *При проблеми* с изпълнение на команда, натиснете *Ctrl + Z* или *Ctrl + C* за прекъсване на изпълнението й.

1. Управление на правата за достъп до файлове

Командата "ls -l" показва правата за достъп до съдържащите се файлове и поддиректории в дадена директория. С **„ls –l fileName“** се извеждат правата за достъп до конкретен файл.

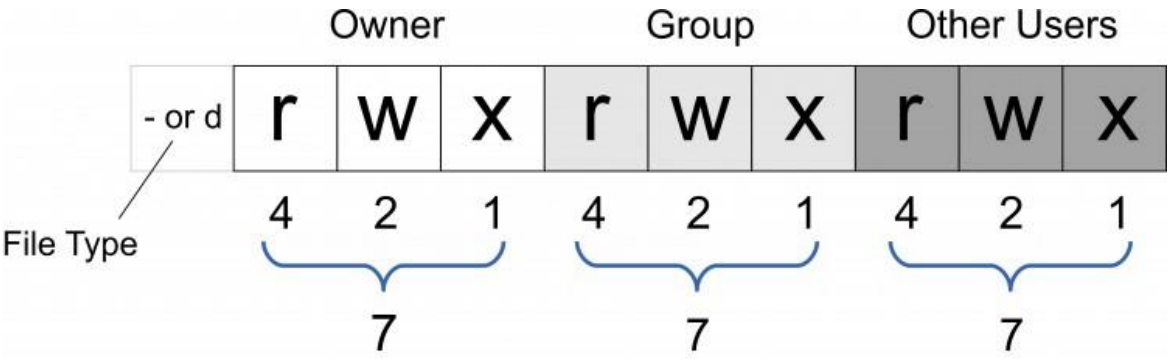
Командата "ls -al" извежда същото като **"ls -l"**, но включително и скритите файлове и директории.

Фиг. 1. показва начина на образуване на правата за конкретен файл или директория.



Фиг. 1. Права за достъп - 1

Фиг. 2 обяснява подробно формирането на цифровите еквиваленти на правата.



Фиг. 2. Права за достъп - 2

Вж. Таблица 1 може да се проследи значението на всяка една от стойностите, зададени в правата за достъп.

Таблица 1

Описание на стойностите в правата за достъп

Стойност	Значение
d	Директория
- (на първа позиция)	Файл
-	Специфичното право за достъп не е зададено.
r	Достъп за четене. r=4
w	Достъп за запис. w=2
x	Достъп за изпълнение, ако файлът е програма. x=1

В таблица 2 са описани различни представяния на правата за достъп – осмични, двоични и символни.

Таблица 2

Представяне на правата за достъп

Осмично	Двоично	Символно	Съответствие
0	000	---	Няма право
1	001	--x	Само право за изпълнение
2	010	-w-	Само право на запис
3	011	-wx	Изпълнение и запис: 1 (execute) + 2 (write) = 3
4	100	r--	Само право за четене
5	101	r-x	Четене и изпълнение: 4 (read) + 1 (execute) = 5
6	110	rw-	Четене и запис: 4 (read) + 2 (write) = 6
7	111	rwx	Пълни права: 4 (read) + 2 (write) + 1 (execute) = 7

В Таблица 3 са отразени основните команди, които се използват при управление на правата за достъп до файлове и директории.

Таблица 3

Основни команди за управление на правата за достъп до файлове и директории в Unix-базираните ОС

Команда	Предназначение
chmod (change mode)	Променя правата за достъп до файл или директория. Използва се в символен и абсолютен режим. Синтаксис: *chmod [-R] [classes][operator][modes] file*
chown (change owner)	Променя собственика на файл или директория. Основен синтаксис: *chown user filelist*, където filelist може да е един файл / директория или списък от файлове / директории. Например, *chown manager file1 dir1*. Командата не променя автоматично собственика на съдържащите се в дадена директория файлове. За да го направи, трябва да се изпълни *chown -R user dir*.
chgrp (change group)	Променя групата на даден файл. Основен синтаксис: *chgrp group filelist*, където filelist може да е един файл / директория или списък от файлове / директории. Например, *chgrp admin file1*.

chmod в символен режим се използва заедно с оператори за добавяне и премахване на право за достъп или набор от правила за достъп. Отразени са в Таблица 4.

Таблица 4

Оператори на chmod

Оператор	Предназначение
+	Добавя право/а за достъп.
-	Премахва право/а за достъп.
=	Правата за достъп стават единствени.

Когато се присвояват права за достъп на отделен потребител, група или други потребители, се използват символни атрибути. Отразени са в Таблица 5.

Таблица 5

Символни атрибути на chmod

Символ	Значение
u	Собственикът на файла/директорията
g	Групата собственик
o	Всички останали
a	Всички

Случаи на използване на символните атрибути в комбинация с оператори:

- g+w — добавя право на запис (write) към група.
- o-rwx — премахва всички права за достъп на други потребители;
- u+x — дава право на изпълнение (execute) на собственика на файла;
- a+rw — дава право за четене и запис (read and write) на всички потребители;
- ug+r — дава право за четене (read) на файл на собственик и група;

- g=rx — позволява само на група права за четене и изпълнение (read and execute), без запис.

Команда umask

Връща или задава стойността на маската, използвана при създаване на даден файл. Новите файлове се създават с права да достъп по подразбиране. Разрешенията до новия файл могат да бъдат ограничени чрез прилагане на "маска", наречена Umask. Командата umask се използва за задаване на тази маска или за показване на текущата й стойност.

Синтаксис: **umask [-S] [mask]**, където:

- -S – приема символно представяне на маската или връща такова;
- mask – ако е зададена валидна маска, то я задава за конкретния файл / директория. Ако не, то извежда текущата маска.

В Таблица 6 са отразени видове маски и съответстващите им права за достъп.

Таблица 6

Видове маски

Umask	Created Files	Created Directories
000	666 (rw-rw-rw-)	777 (rwxrwxrwx)
002	664 (rw-rw-r--)	775 (rwxrwxr-x)
022	644 (rw-r--r--)	755 (rwxr-xr-x)
027	640 (rw-r-----)	750 (rwxr-x---)
077	600 (rw-------)	700 (rwx------)
277	400 (r--------)	500 (r-x------)

За удобство, може да се използва калкулатора, предложен на следния адрес: https://goo.gl/hWu2H5.

Подробна информация за командата можете да прочетете на https://en.wikipedia.org/wiki/Umask.

SUID и SGID права за достъп

Представляват допълнителни права за достъп, задавани на програми. Това е т. нар. механизъм „Set User ID (SUID) и Set Group ID (SGID)“.

При изпълнение на програма, потребителят наследява правата за достъп до нея от собственика й чрез разрешаване на т. нар. SUID бит. Програми, които нямат подобен SUID бит, се стартират с правата на потребителя, който я изпълнява в момента. Обикновено програмите се изпълняват с правата на групата на дадения потребител. SUID и SGID битовете се появяват с буква "s", ако има такива специални права за достъп. SUID "s" бита се появява в първата група от правата (групата на собственика), на мястото на правото за изпълнение. Тези специални права за достъп могат да бъдат управляване чрез използване на символните оператори и атрибути на chmod.

Например, командата **ls -l /usr/bin/passwd** проверява правата за достъп до програмата passwd, която се грижи за управлението на паролите на потребителите. Резултатът, който извежда, е *-rwsr-xr-x 1 root root 54256 May 17 2017 /usr/bin/passwd*.

Командата **chmod ug+s dirname** добавя специалните права за достъп.

2. Пренасочване на входа и изхода на командите

Входът на много UNIX команди е от стандартния вход (standard input - stdin), изходът им се извежда в стандартния изход (standard output - stdout), а съобщенията за грешки се извеждат в стандартния изход за целта (standard error - stderr). По подразбиране, стандартният вход е свързан към клавиатурата на терминала, а стандартния изход и грешките към екрана на терминала. Чрез натискане на <Ctrl-D> се достига до края на файла.

Възможно е да се извърши пренасочване на входа и изхода на командите чрез определяне на дестинацията в командния ред. За целта се прилагат специални метасимволи. Описани са в Таблица 7.

Таблица 7

Метасимволи за пренасочване на входа и изхода на командите

Символ	Действие
От шела на C	
>	Пренасочва стандартен изход
>&	Пренасочва stdout и stderr
<	Пренасочва стандартен вход
>!	Пренасочва стандартен изход; презаписва файл, ако съществува
>&!	Пренасочва stdout и stderr; презаписва файл, ако съществува
|	Пренасочва стандартен изход към друга команда (pipe)
>>	Добавя към стандартен изход
>>&	Добавя към stdout и stderr
От шела на Bourne	
>	Пренасочва стандартен изход
2>	Пренасочва стандартна грешка
2>&1	Пренасочва стандартна грешка към стандартен изход
<	Пренасочва стандартен вход
|	Пренасочва стандартен изход към друга команда (pipe)
>>	Добавя към стандартен изход
2>&1|	Пренасочва stdout и stderr към друга команда (pipe)

Общ вид на команда с пренасочване на стандартен вход и изход:

command -[*options*] [*arguments*] < *input file* > *output file*

3. Пример за използване на команди по управление на правата за достъп

```
id
cd ~
id
touch octal
ls -l octal
chmod 600 octal
ls -l octal
chmod u+x octal
chmod u-x octal
chmod g+x octal
chmod 777 octal
chmod o-wx octal
chmod go=rw octal
chmod u+x,go=rx octal
chmod o+wx,u-x,g=rx octal
ls -l octal
```

Описание на примера: Проверява uid / gid на директорията. Премества се в /home директорията на текущия потребител и отново проверява uid / gid. Създава нов файл и проверява правата за достъп до него. Променя правата за достъп до файла на 600. Отново проверява правата за достъп до файла. Неколкократно извършва промяна на правата за достъп до файла. Накрая отново прави проверка на правата.

Когато тествате примера, можете да проверявате правата за достъп след всяко изпълнение на chmod.

III. ВЪПРОСИ ЗА САМОПРОВЕРКА

1. Кое е съответствието на пълни права за достъп до файл или директория в символен вид? А в абсолютен (с числа)?

2. Как следва да обозначите в символен вид правото 640?

3. Коя команда се използва за промяна на правата за достъп до файл или директория?

4. Коя команда се използва за промяна на собственика на файл или директория?

IV. ДОПЪЛНИТЕЛНА ЛИТЕРАТУРА

1. Managing Group Access
2. Permissions
3. Unix Permissions Calculator