

10. Работа с файлове и директории (навигиране), създаване, копиране, изтриване, преименуване, местене на файл

Командата find намира приложение при търсене в указаното дърво, като извършва проверка за име на файла, дата на създаване, права на достъп, размер, собственик и т.н., за да намери файла, удовлетворяващ даденото условие. Командата има следният общ вид

find [PATH...] [EXPRESSION...]

Под **expression** се има предвид критерият или критериите, по които се извършва търсене. Използва се шаблон на търсене. Примери за приложение:

find \$HOME -print - извеждане на съдържанието на личната директория на текущия потребител, включително поддиректории и файлове;

find /home -name file1 - търсене на файл в директория / home;

find /home -name 'memo*' -user unix -print - търсене на всички файлове, съдържащи „memo“ в името си, създадени от потребител unix;

find / -mtime -2 -print - търсене на всички файлове, модифицирани преди два дни.

Командата **locate** се използва за търсене на файлове по име. Връща имената на всички файлове, които съдържат указания низ. Поддържа и използва собствена база от данни, в която извършва търсенето. Тя се обновява периодично. Може да не намери имената на новосъздадени файлове. Общият ѝ вид е

locate [OPTIONS] PATTERN

Примери за приложението ѝ:

locate -c file - преброяване на файлове;

locate -A file-* - извеждане на всички файлове, които съдържат „file-“ в имената си.

Търсене може да се извършва и в съдържанието на файловете. За целта се използва командата **grep**. Синтаксисът ѝ е

grep [OPTIONS] PATTERN [FILES]

Някои полезни опции:

- c - извежда броя на редовете;
- f file - търсеното условие се получава от файл, а не от командния ред;
- i - не се прави разлика между малки и големи букви;
- r - извършва се рекурсивно търсене в поддиректориите на зададената директория.

Примери за приложение на командата са:

grep adm /etc/group - претърсване на файл по ключова дума (в случая ще изведе членове на група adm);

grep -e rn* /etc/group - претърсване на съдържанието на файл по шаблон;

grep -c admin /etc/group - извеждане на броя редове, в които се съдържа ключовата дума.

Една от командите с многостранно приложение е cat. Тя се използва за извеждане на съдържание, обединяване и създаване на файлове. Общият ѝ вид е

cat [OPTION]... [FILE]...

Опцията -n показва **номерацията на редовете във файла**, а -T -табулацията.

Примери за практическото приложение на командата:

cat -n \$HOME/f1 - показва номерация на редовете на файла f1;

cat > f1 - добавя със заместване, Ctrl+C на нов ред за запазване;

cat >> f1 - добавя в края на файла;

cat f1 > f2 - копира съдържанието на f1 в f2;

cat f1 >> f2 - добавя съдържанието на f1 в f2;

cat f1 f2 > f3 - обединява съдържанието на f1 и f2 в f3;

cat f* - извежда съдържанието на всички файлове, отговарящи на шаблона;

cat /proc/cpuinfo | grep model \equiv **grep model /proc/cpuinfo** - претърсва съдържанието на файл по ключова дума;

grep root /etc/passwd >> f1 - добавя резултатите от команда grep към файл;
cat f1 | grep hello - търси по ключова дума в съдържанието на файл;
cat -E f1 - извежда съдържанието на файл с обозначаване на край на редовете.

Командите, които извеждат част от съдържанието на файлове, са **head** и **tail**. Те показват първите или последните редове, блокове или символи от файла. Общият им вид е

head [OPTIONS] <FILE>

tail [OPTIONS] <FILE>

Могат да се посочат следните примери:

tail -20 f1 - извежда последните 20 реда от файл;

head -15 f1 - извежда първите 15 реда от файл;

head -20 /etc/passwd | tail - 5 - извежда последните 5 реда от първите 20;

tail -20 /etc/passwd | head -5 - извежда първите 5 реда от последните 20.

За преброяване на редовете, думите и символите в текстови файлове се използва командата **wc**. Общият ѝ вид е

wc [OPTIONS] <FILE>

Примери за приложението ѝ:

wc f1 - извежда едновременно броят на редовете, думите и символите на файл;

wc -l f1 - за преброяване на редовете;

wc -w f1 - за преброяване на думите;

wc -c f1 - за извеждане на байтовете на файла.

Командата **ls** се използва за извеждане на списък на отворените файлове. Общият ѝ вид е

ls [OPTIONS]

Някои примери за приложението ѝ:

ls -u root - извеждане на отворените файлове от потребител root;

ls -i -u^root - извеждане на отворените файлове с изключение на потребител root;

ls -i 4 - извеждане на мрежовите файлове IPv4;

ls -i TCP:1-1024 - извеждане на файловете, които използват TCP порт;

ls -p 1 - извеждане на файловете, отворени от процес с идентификатор 1.

Една от най-лесните за приложение команди е **pwd**, която извежда пътя спрямо корена към текущата директория. Тя обикновено се използва без опции.

За да се промени текущата директория, се използва **cd**. Общият ѝ вид е

cd [OPTIONS] DIRECTORY

Някои примери за приложение ѝ:

cd . . - преминава се в родителската директория;

cd /home - преминава се в директория home;

cd ~ \equiv **cd** - преминава се в личната директория на текущия потребител, която по подразбиране е поддиректория на home;

cd ./dir1 -преминава се в директория dir1, която е наедно ниво с текущата;

cd ~/dir1 -преминава се в директория dir1, която се намира в личната директория на текущия потребител.

Създаването на нова директория се извършва чрез **mkdir**. Опцията **-p** се използва, за да се създадат необходимите родителски директории, ако не съществуват. Общият ѝ вид е

mkdir [OPTIONS] DIRECTORY

Примери за приложението ѝ:

mkdir myDir - създава една директория;

mkdir d1 d2 - създава две директории едновременно;

mkdir -p ./dir1/dir2/dir3 - създава дърво от директории.

Изтриването само на празна директория може да стане чрез командата **rmdir**. Опцията

-- ignore-fail-on-nonempty

се използва, за да не се изведе съобщение за грешка при опит за изтриване на директория, в която има съдържание. Като резултат - директорията не се изтрива. **Опцията -p** намира приложение при изтриване на зададено като аргумент дърво от директории.

Примери: **rmdir - - ignore-fail-on-non-empty d2**

или **rmdir -p one/two/three.**

Командата **ls** извежда **съдържанието на директория**. Ако се изпише без опции, показва **съдържанието на текущата директория**. Някои полезни опции на командата са следните:

-a - извежда имената на скритите файлове;

-f - извежда съдържанието на директорията, без да сортира;

-h - извежда размера в килобайти и мегабайти;

-l - извежда списък за вида на файловете, правата за използването им, собствениците им, времето на последното модифициране;

-t - сортира файловете според времето на модифициране;

-u - сортира файловете според времето на достъп;

-R - извежда рекурсивно съдържанието на директорииите и техните поддиректории.

Примери за приложението й:

ls -l /home - извежда съдържанието на директория home с правата за достъп до файлове и поддиректории;

ls -al \equiv **ls -la** - извежда съдържанието на текущата директория, включително скритите файлове и правата за достъп;

ls -l - извежда съдържанието на текущата директория заедно с индексния номер на всеки елемент;

ls -u -l - извежда съдържанието на текущата директория, като подрежда елементите по име и дата на последен достъп.

Аналогична на ls е командата **dir**, но без допълнителното оцветяване на файлове и директории. Опциите на dir са подобни на тези на ls.

Пример за съвместно използване на част от разгледаните в тази глава команди, включително с пренасочване на входа и изхода. Всяка команда се изписва на нов ред след натискане на Enter.

```
cd ~  
mkdir myDir  
sudo cp -r ~/myDir /home/test  
cd ../test && sudo touch f1  
sudo mv f1 ~/testfile  
echo "new_text" | sudo tee -/testfile  
sudo rmdir /home/test  
pwd && ls -l  
ls -l testfile  
who > testfile  
(pwd; ls -l) > names  
who >& /dev/null  
touch myfile1 myfile2 myfile3  
(cat myfile1 > myfile2) >& myfile3
```

Описание на примера. Премества се в личната директория на текущия потребител. Създаваме директория myDir. Копираме myDir в директорията home под наименование test. Премества се в test и в нея създаваме файл f1. Премества файла f1 в личната директория на текущия потребител под името testfile. Въвежда текст new_text в testfile. Изтрива директорията test. Проверява коя е текущата директория и извежда списък със съдържащите се в нея файлове и директории с правата им за достъп. Извежда правата за достъп до файла testfile. Пренасочва стандартния изход към файл testfile - първо на една команда, а после на списък от команди. Пренасочва стандартния изход и грешка на команда who към устройство null. Създава три файла едновременно, които използва за пренасочване на стандартния изход и стандартната грешка на команда cat.