

Основни команди в *Linux*

наръчник за потребители

Юри Хоптериев

Този малък наръчник съдържа **често използвани** команди (и **някои** техни опции) в операционната система **Linux**. Предназначен е за потребители (не са включени команди за администриране на системата).

В дефинициите на командите: 1) текст, написан в рамка е **обяснителен**; 2) параметър, написан на сив фон **не е задължителен**. Наръчникът е подходящ за използване като помощно средство при изучаване на ОС **Linux**.

Общ вид на команда в Linux

команда опции параметри

Разделител между командата, опциите и параметрите, между отделните опции и отделните параметри е **интервалът**. Някои команди имат задължителни параметри.

Първи команди

who

Извежда на екрана (в конзолата) списък на включените в момента потребители.

whoami

Кой потребител съм аз?

who am i

Кой потребител съм аз? Не работи в **стари** версии на системата.

help

Извежда на екрана информация (help) за командите в **Linux**.

man команда

Извежда на екрана информация за указаната като параметър команда.

clear

Изчиства екрана.

passwd

Смяна на паролата.

logout

Изход. Комбинация от клавиши **Ctrl+d** също е изход.

date

Извежда дата и време.

cal

Извежда календар (за повече информация: **man cal**).

Работа с файлове и директории

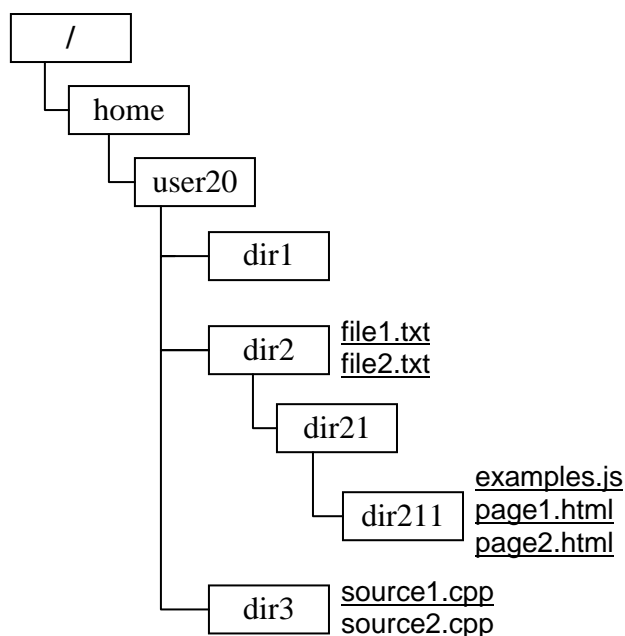
Специални имена на директории:

- . - текуща директория;
- .. - предходна директория;
- ~ - потребителска **home**-директория;
- / - главна директория (корен на дървото).

Път в дърво от директории и файлове (фиг. 1) е последователността от възлите, през които се минава за да се стигне от една точка на дървото до друга. Разделител между отделните имена в пътя е символът “/”.

Път се описва:

- от корена на дървото, започвайки със специално име “/” (**пълен, абсолютен път**);
- от потребителската **home**-директория, започвайки със специално име “~”;
- от текущата директория (**относителен път**). При връщане назад, се използва специално име “..”.



Фигура 1. Дърво от директории и файлове

Примери:

Пълен (абсолютен) път до **file1.txt**, намиращ се в **dir2** (фиг. 1):

/home/user20/dir2/file1.txt

Пълен (абсолютен) път до **page1.html**, намиращ се в **dir211** (фиг. 1):

/home/user20/dir2/dir21/dir211/page1.html

Път от потребителската **home**-директория до **page1.html** (фиг. 1):

~/dir2/dir21/dir211/page1.html

Път от потребителската **home**-директория до **dir1** (фиг. 1):

~/dir1

Относителен път от **dir2** до **page2.html** (фиг. 1):

./dir21/dir211/page2.html

или

dir21/dir211/page2.html

Относителен път от **dir3** до **file2.txt** (фиг. 1):

../dir2/file2.txt

или

../dir2/file2.txt

Относителен път от **dir21** до **source1.cpp** (фиг. 1):

../../dir3/source1.cpp

или

../../dir3/source1.cpp

Команди

pwd

Извежда пътя от корена на дървото до текущата директория (показва текущата директория).

ls

Извежда съдържанието (списък от имена на файлове и директории) на указаната като параметър директория. Ако параметърът е пропуснат, *се подразбира текущата директория* (екран 1).

```
student@e-learning:~/examples$ ls
example1.html  example1.php  example2.html
example2.php   example_.html examples.php
pictures

student@e-learning:~/examples$ _
```

Екран 1. Резултат от изпълнение на командата **ls**

Пътят може да съдържа **специални символи и записи** (таблица 1), което позволява да се зададе изпълнение на команда за множество от файлове, чиито имена съответстват на параметъра.

Таблица 1. Специални символи и записи в пътя

<u>Специален запис</u>	<u>Значение</u>
?	Съответства на един произволен символ.
[множество_от_символи]	Съответства на един символ, принадлежащ на множеството.
[^множество_от_символи]	Съответства на един символ, непринадлежащ на множеството.
*	Съответства на произволна последователност от символи.

В показания по-горе пример (екран 1) е изведен пълният списък от имена на файлове и директории, съдържащи се в текущата директория **examples**. Във всеки от следващите примери (екрани 2-8) командата **ls** извежда списък само на тези файлове от същата директория, *имената на които съответстват на зададения параметър*.

```
student@e-learning:~/examples$ ls example?.html
example1.html  example2.html  example_.html

student@e-learning:~/examples$ _
```

Екран 2. Изпълнение на командата ls с параметър example?.html

```
student@e-learning:~/examples$ ls example[0-9].html
example1.html  example2.html

student@e-learning:~/examples$ _
```

Екран 3. Изпълнение на командата ls с параметър example[0-9].html

```
student@e-learning:~/examples$ ls example[^0-9].html
example_.html

student@e-learning:~/examples$ _
```

Екран 4. Изпълнение на командата ls с параметър example[^0-9].html

```
student@e-learning:~/examples$ ls e*.php
example1.php    example2.php    examples.php

student@e-learning:~/examples$ _
```

Екран 5. Изпълнение на командата **ls** с параметър **e*.php**

```
student@e-learning:~/examples$ ls example[s_].*
example_.html    examples.php

student@e-learning:~/examples$ _
```

Екран 6. Изпълнение на командата **ls** с параметър **example[s_].***

```
student@e-learning:~/examples$ ls example[^s_].*
example1.html    example1.php    example2.html
example2.php

student@e-learning:~/examples$ _
```

Екран 7. Изпълнение на командата **ls** с параметър **example[^s_].***

```
student@e-learning:~/examples$ ls example?.*
example1.html    example1.php    example2.html
example2.php     example_.html    examples.php

student@e-learning:~/examples$ _
```

Екран 8. Изпълнение на командата **ls** с параметър **example?.***

Опции на ls

ls -R име_на_директория

Изпълнява командата **ls** рекурсивно, т.е. изпълнява **ls** за всяка директория от дървото (поддървото), чийто корен е указаната като параметър.

ls -a

Изпълнява командата **ls**, включвайки в списъка и скритите файлове. Скрити файлове в **Linux** са тези, чиито имена започват със символа “.”.

ls -l

Извежда съдържанието на указаната като параметър или подразбиращата се директория в т.н. **дълъг формат** (екран 9).

```
student@e-learning:~/examples$ ls -l
total 28
-rw-r--r-- 1 student student 362 2017-05-09 14:44 example1.html
-rw-r--r-- 1 student student 531 2017-05-09 14:47 example1.php
-rw-r--r-- 1 student student 202 2017-05-09 14:44 example2.html
-rw-r--r-- 1 student student 219 2017-05-09 14:46 example2.php
-rw-r--r-- 1 student student 348 2017-05-09 14:45 example_.html
-rw-r--r-- 1 student student 385 2017-05-09 14:47 examples.php
drwxr-xr-x 2 student student  48 2017-05-09 14:47 pictures

student@e-learning:~/examples$ _
```

Екран 9. Дълъг формат на командата ls

В **дългия формат** на **ls** (екран 9) за всеки от файловете се извежда по един ред информация, включваща:

1) **10 символа**, *първият* от които показва **вида на файла** (*обикновен файл, линк или директория*), а останалите **9** могат да се разглеждат като **3 последователни тройки символи**, всяка от които дават информация за правата на достъп до файла (екран 9) за четене (**r**), писане (**w**) и изпълнение (**x**) на трите типа потребители, съответно собственик на файла (**u**), група (**g**) и всички останали (**o**).

Примери:

– за файла **example1.html** (екран 9) собственикът има права за четене и писане (символът “-” означава липса на съответното право), групата има право за четене и останалите потребители имат право за четене;

– за директорията **pictures** (екран 9) собственикът има права за четене, писане и изпълнение, групата има права за четене и изпълнение и останалите потребители имат права за четене и изпълнение.

Собственикът на файла може да променя првата посредством команда [chmod](#).

2) Брой линкове към файла.

3, 4) Име на собственика и име на групата (екран 9).

5) Дължина на файла в байтове (екран 9).

6, 7) Дата и време на запис на файла (екран 9).

8) Име на файла (екран 9).

dir път

Извежда съдържанието на указаната като параметър директория. Не работи в стари версии на системата.

tree име_на_директория

Извежда (показва) дървото, чийто корен е указаната като параметър директория.

mkdir име

Създава директория с указаното като параметър име.

cd име_на_директория

Сменя текущата директория.

rmdir име

Премахва (изтрива) указаната като параметър директория. Директорията не може да бъде изтрита, ако не е празна или ако е текуща.

cat име_на_текстов_файл

Извежда съдържанието на текстовия файл. Ако липсва параметър, се подразбира стандартното входно устройство (клавиатура).

Използването на командата в комбинация с механизма за [пренасочване](#) дава възможност за създаване на текстови файлове:

cat > име_на_текстов_файл

За край на въвеждането: **Ctrl+d**

cp път_1 път_2

Копира файл(ове). Параметърът **път_1** е път до файла, който ще се копира, а **път_2** е път до мястото, където ще се копира. Ако се копират повече файлове с една команда, **път_2** (последният параметър) **задължително** трябва да е към **директория**.

Примери:

cp ~/dir3/*.cpp ~/dir1 ще копира всички **.cpp**-файлове от **dir3** в **dir1** (фиг. 1);

cp ~/dir2/file1.txt ./file1-copy.txt ще копира **file1.txt** от **dir2** в **текущата директория** под име **file1-copy.txt** (фиг. 1).

mv път_1 път_2

Премества файл(ове). Параметрите са аналогични на тези от **cp**. Използва се и за преименуване на файл.

ln `път_1` `път_2`

Създава твърда връзка (hard link) към файл. Параметрите са аналогични на тези от **cp**.

ln -s `път_1` `път_2`

Създава символична връзка (symbolic link) към файл. Първият параметър става съдържание на символичния линк, т.е. *това е пътят, посредством който символичният линк ще осъществява достъп до файла*. Вторият параметър е аналогичен на този от твърдия линк.

rm `име_на_файл`

Премахва (изтрива) указания като параметър файл (за опции: **man rm**).

Смяна на правата за достъп до файл (команда **chmod**)

Правата се задават като параметър на командата **chmod** посредством *текстов низ* или *трицифрено осмично число*.

chmod `текстов_низ` `име_на_файл`

В текстовия низ се указва на кой(кои) потребител(и) (**u**, **g** и **o**) как се променя(т) (+ или -) едно или повече права (**r**, **w** и **x**). *Например:*

chmod go-r example.sh *отнема на групата и другите потребители (без собственика) правото за четене на файла example.sh без да променя останалите права;*

chmod u+wx example.sh *дава на собственика права за писане и изпълнение на файла example.sh без да променя останалите права.*

chmod `трицифрено_осмично_число` `име_на_файл`

С трицифрено осмично число се задава *цялата конфигурация* от права. Логиката е следната:

Правата **rwX** на *един* тип потребители се представят като трицифрено *двоично* число (1- има право, 0- няма право). Например **r-X** се представя като **101** и означава *права за четене и изпълнение без писане*. Трицифреното двоично число се преобразува в *осмична бройна система* и се получава *една осмична цифра* (в примера това е цифрата **5**), съдържаща *пълна информация* за правата на един тип потребители. *Трите осмични цифри*, записани в параметъра на командата, задават правата съответно на **u**, **g** и **o**. *Например*, след изпълнение на командата **chmod 751 example.sh**, *собственикът* ще има *всички* права, *групата* ще има права за *четене и изпълнение (без писане)*, а *другите потребители* ще имат право *само за изпълнение* на файла **example.sh**.

Пренасочване на входа и изхода

По подразбиране командите получават/извеждат данни от/към стандартните входно-изходни устройства. Има възможност за пренасочване на входа и изхода, която позволява входните/изходните данни за/от изпълнение на команда да бъдат прочетени/записани от/във файл.

команда > **име_на_файл**

Командата се изпълнява и резултатът се *записва* във файл (с указаното име), който системата създава за тази цел.

команда >> **име_на_файл**

Командата се изпълнява и резултатът се *добавя* в края на файла с указаното име. Ако не съществува такъв файл, системата го създава.

команда < **име_на_файл**

Командата се изпълнява, като необходимите входни данни се *получават* от файла с указаното име.

Конвейер от команди

команда | **команда** | **команда**

Командите се изпълняват последователно, като резултатът от изпълнена команда се подава като входни данни за следващата. Последната команда извежда резултата от изпълнението на конвейера.

Процеси

ps

Извежда списък на изпълняващите се в момента процеси.

kill **номер_на_процес**

Прекъсва (прекръпява) изпълнението на указания процеси.

Изпълнение на файлове за пакетна обработка

sh **файл_за_пакетна_обработка** **параметри**

Извиква за изпълнение файл за пакетна обработка (shell-скрипт).

. **файл_за_пакетна_обработка** **параметри**

Извиква за изпълнение файл за пакетна обработка (shell-скрипт).

Работа с отдалечена машина

ping **адрес**

Проверява състоянието на връзката с отдалечена машина.

ssh адрес

Осъществява връзка за работа с отдалечена машина.

Трансфер на файлове

ftp адрес

Осъществява **ftp**-връзка с отдалечена машина.

Команди при работа с FTP:

open адрес

pwd

Изпълнява командата **pwd** на отдалечената машина.

ls

Изпълнява командата **ls** на отдалечената машина.

cd

Изпълнява командата **cd** на отдалечената машина.

!pwd

Изпълнява командата **pwd** на локалната машина.

!ls

Изпълнява командата **ls** на локалната машина.

lcd

Изпълнява командата **cd** на локалната машина.

get име_на_файл

Осъществява трансфер на файл *от отдалечената към локалната* машина.

put име_на_файл

Осъществява трансфер на файл *от локалната към отдалечената* машина.

mget имена_на_файлове

Осъществява трансфер на *няколко* файла *от отдалечената към локалната* машина.

mput имена_на_файлове

Осъществява трансфер на *няколко* файла *от локалната към отдалечената* машина.

close

Край (затваряне) на връзката.

?

Help за FTP.

exit

Изход.

quit

Изход.