Още за файлове и права

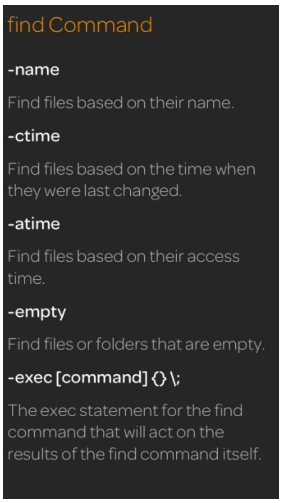
08 октомври 2020 г. 13:14

Основни команди с цел привилегий

- Всеки файл е собственост на определен потребител и група
- chown – промяна на собствеността на файл (потребител и група)
- chgrp – промяна само на груповата собственост на файл
- newgrp – временен login към друга първична група
- Правата на файловете могат да бъдат представени символично или цифрово
- В Линукс се дефинират права за четене, запис и изпълнение за потребителя собственик, за групата собственик и за всички останали потребители
- umask – настройва правата по подразбиране на файлове и директории
- chmod – промяна на правата на файлове и директории

Командата find:

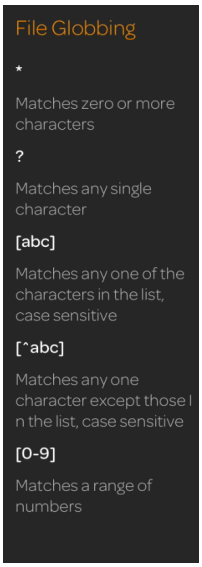
- find . намира всичко в директорията и под нея
- find име на директория намирате всички директории свързани с нея
- find . –type d ще намери всички директории без файлове
- find . –type f ще намери всички файлове
- find . –type f –name “и
- sudo find -type f -name "test*” астерикса ще го направи wildcard
- sudo find . -type f -iname "test* ще извади case insensitive
- find -type f -iname "*.txt"
- find . -type f -mmin -10 намира файлове модифицирани в последните 10 мин. (по малко от десет минути + значи повече)
- find -mmtime (опцията посочва дни)
- find -size option find . -size +5M търси файлове с размер по голям от 5 мегабайта, ако се сложи минус по малко
- find . –empty
- find –perm 777 или 666
- find webin -exec chown ivan:sudo {} +
- find webin -type f -exec chmod 660 {} +



Регулярни изрази:
Регулярните изрази позволяват фина спецификация на търсенията и се поддържат от множество популярни команди и езици за програмиране.

- grep g..m /etc/passwd
- Точката обозначава един знак.
- ^ търси в началото на линията
- $ търси в края на линията
- Може определени комбинации
- grep –i прави grep не чуствителен към знаците
- grep ^[Aa].[Aa] /etc/passwd
- man 7 regex

- Wildcards (file globbing) - предлагат лесен начин за подаване на множество имена на файлове като аргументи на команди (програми)
- Wildcards се указват със специални символи:
- ? – единичен символ
- * - нула или повече последователни символа
- […] – класове и обхвати {…} – списък символни низове



Пример:

1. ls *.txt
2. ls test*
3. ls ?.txt
4. ls ?????.txt
5. ls [Tt]*.txt
6. ls [t].txt
7. ls [Ww]eather[Re]eport199[11[0-9]?2017*
8. ls folder/*/ outputs whatever is inside

Всички UNIX базирани системи имат като "кофа" за всичкият изход от комади.

"Кофата" се нарича още standard output.

Абривиацията е 'stdout' ===> Стандартен изход.

Пример:

Когато напишете командата ls тя генерира stdout, той отива в „кофата“ и после от кофата се появя на екрана.

Какво можем да правим с него:

Можем даго насочваме по наше желание

ls –lah >> test.txt

Примера от горе един вид прескача кофата и го слага във файла.

Piping – позволява STDOUT (изход) на една програма (лявата на pipe) да стане STDIN (вход) на друга (дясната на pipe) .

Символът за pipe е |

tee – позволява едновременно запис към STDOUT и към файл

xargs – позволява предаването на списък с аргументи към други програми
find /etc -name “*.conf” | xargs cat

Стандартен вход:

Стандартиният вход (standard input) е най често зададен от нас използвайки клавиатурата – пишейки командите.

Но може да бъде и от приложение/deamon.

Файлове и стандартен вход могат да се ползват в стандартният вход на други команди:

- със <, |
- Абривиация – ‘stdin’
- Примерно wc < test.sh.
- cat /etc/passwd | less

Стандартна Грешка:

Стандартна грешка ‘Standard Error’

Абривиация ‘stderr’

Попринцип е излиза на екрана като стандартен изход

Има ‘файлова дръжка’ – цифра с която е свързана – 2

stdin е 0

stdout е 1

./script.sh == виждаме грешката директно на екрана

./script.sh 2> error.log пращаме грешката във файл

./script.sh 2>&1 | less == отваряме грешката дирекнто на екрана с less

Тее командата:

Командата ‘tee’ чете данните от ‘stdin’ и записва данните в ‘stdout’. Командата е много полезна за връзване на много команди поглеждайки ‘stdout’ на различни етапи.

find . -name 'test*.t*' | tee found.log

Направете ваша папка в която създайте 20/30 файла.

touch {file1..30}

След това намерете с find всички файлове по име и използвайте tee за да пренасочите stout във файл.

find . -name 'test*.t*' | tee found.log | sort -n | tee sorted.log

tr командата която конвертира, извлича и изтрива символи:

- cat tr_test.txt | tr “[a-z]” “[A-Z]”
- cat tr_test.txt | tr "[:lower:]" "[:upper:]"
- echo "WELCOME TO THE TEST FILE" | tr [:space:] ‘\t’
- echo "my ID is 12432" | tr -cd [:digit:]
-
- Можем да превеждаме от и към файл:
- tr '{}' '()' < inputfile > outputfile
- Да махнем всички не принтиращи се знаци:
- tr -cd [:print:] < file.txt

Без опции - конвертиране от един набор символи към друг (ива (извлича) повтарящи се поредни символи и оставя самобикновено се използва за конвертиране на малки към големи букви и обратно):

tr a-z A-Z - опция -s - изтро един от тях (обикновено се използва за сливане на поредни празни редове):

tr -s ‘\n’ - опция -d - изтрива набор от символи (обикновено се използва за премахване на специални символи - например carriage return): tr -d ‘\r’

sort подрежда текст по определени критерии (по подразбиране на лексикален принцип)

- sort filename.txt

- sort > filename.txt
- Syntax :
- $ sort inputfile.txt > filename.txt
- $ sort -o filename.txt inputfile.txt
- Reverse Order – sort –R
- sort -n filename.txt /// sort -nr filename.txt{random numbers}
- sort -k filename.txt в колони
- sort –M по месеци

Uniq

- uniq - премахва дублираните редове в сортиран файл
- Опции:
- -i - игнориране на разликата между големи и малки букви
- -d - показване само на дублираните редове
- -u - показване само на уникалните редове
- -c - показване колко пъти се среща даден ред

cut

- Извлича избрани полета от редове текст (използва табулация за разделител по подразбиране). Работи най-добре със структуриран текст (текст с колони).
- Опции: -d - избор на разделител
- -c - извличане на полета в определен обхват символи
- -f - извличане на полета в определен обхват колони
- $ cut -d: -f1,3 /etc/passwd root:0 test:1000

Други полезни команди:

- expand - превръща табулациите (TAB) в празни места (SPACE)
- unexpand - превръща празни места (SPACE) в табулации (TAB)
- fmt - форматира текстови файл по определни критерии
- join - съединява редове от два файла на база общо поле
- nl - номерира редовете от файл
- od - показва съдържанието на файл в различни формати (осмичен по подразбиране)
- pr - страницира или подрежда в колони файл като подготовка за принтиране
- split - разделя текст на равни части и ги записва във файлове с определен префикс