

Polecenia systemu Unix/Linux

Polecenia 3

1. vi - edytor tekstowy pełnoekranowy, lista dostępnych opcji znajduje się w pliku "vi_shorcuts.txt",
2. nano - edytor tekstowy pełnoekranowy, lista dostępnych opcji znajduje się w pliku "nano_shortcuts.txt",
3. more - wyświetla zawartość pliku tekstowego w terminalu. Pozwala na przeglądanie strona po stronie,
4. file - zwraca informację o typie pliku,
5. head - wyświetla początkowe 10 linie tekstu,
6. tail - wyświetla ostatnie 10 linii tekstu,
7. operator && - wykonuje następne polecenie, jeśli poprzednie zakończyło się sukcesem,
8. operator ; - wykonuje następnie polecenie, niezależnie od tego, czy poprzednie polecenie zakończyło się sukcesem,
9. operator | - wykonuje następne polecenie, jeśli pierwsze nie zakończy się sukcesem
10. przekierowanie strumieni we/wy
 - > - przekierowanie domyślnego strumienia (ekranu) do pliku, poprzednia zawartość pliku jest zastępowana nową,
 - >> - przekierowanie domyślnego strumienia (ekranu) do pliku w trybie dopisywania treści,
 - < - przekierowanie domyślnego strumienia wejściowego (klawiatura) na plik, i przekazanie jest zawartości do polecenia
 - 2> - przekierowanie domyślnego strumienia błędów (ekran) do pliku
 - 2>&1 - przekierowanie domyślnego strumienia błędów do domyślnego strumienia wyjściowego
11. Potoki | - mechanizm pozwalający na przekazywanie wartości między poleceniami, wartość wyjściowa polecenia z lewej strony potoku, jest przekazywana jak wejście polecenia z prawej strony potoku,
12. przerwanie programu - CTRL+C , przerwanie wykonywania bieżącego programu
13. znaki specjalne - wzorce stosowane do przetwarzania tekstów
 - * - dopasowanie do dowolnej ilości znaków
 - ? - dopasowanie do dokładnie jednego znaku
 - {} - dopasowanie do przedziału wartości
 - [] - dopasowanie do list znaków
14. zmienne specjalne
 - \$0 - nazwa bieżącego skryptu lub powłoki
 - \$1..\$9 - parametry przekazane do skryptu lub powłoki
 - \$@ - zwraca wszystkie parametry przekazane do skryptu
 - \$? - kod powrotu ostatnio wykonanego polecenia
 - \$\$ - PID procesu bieżącej powłoki

Przykłady zastosować przedstawionych poleceń:

```
# Utworzenie nowego pliku tekstowego i zapisanie do niego wybranego tekstu

vi file.txt
i
Ala ma kota
ESC
:wq

# Sprawdzenie typu pliku "file.txt"
file file.txt

# Dublowanie zawartości plik
cat file.txt file.txt > file2.txt

# Zapisanie komunikatu błędu do pliku
ls /dev/one 2> blad.txt

# Zliczenie ilości wierszy w pliku tekstowym
wc -l < file2.txt

# przekierowanie komunikatu błędu do pliku
echo Hello ; cat /dev/two > output.txt 2>&1

# wyświetlenie ostatnich 10 linii pliku syslog
cat /var/log/syslog | tail

# wyświetlenie pierwszych 10 linii pliku auth.log
head < /var/log/auth

# jeśli mamy następujące pliki tekstowe:
ls *.txt

> 001.txt  01.txt  02.txt  03.txt  04.txt  11.txt  12.txt  13.txt  14.txt
# to wykonanie poleceń ze znakami specjalnymi powoduje uzyskanie
następujących wartości:

ls 0*.txt
> 001.txt  01.txt  02.txt  03.txt  04.txt

ls *.txt
> 001.txt  01.txt  02.txt  03.txt  04.txt  11.txt  12.txt  13.txt  14.txt

ls 0?.txt
> 01.txt  02.txt  03.txt  04.txt

ls *1.txt
> 001.txt  01.txt  11.txt

ls ?1.txt
> 01.txt  11.txt

ls ?0?.txt
> 001.txt

ls ?{2,3}.txt
> 02.txt  03.txt  12.txt  13.txt

ls *{1,2,3}.txt
```

001.txt 01.txt 02.txt 03.txt 11.txt 12.txt 13.txt

```
ls 0{1,4}.txt  
> 01.txt 04.txt
```

```
ls 0[1,4].txt  
> 01.txt 04.txt
```

```
ls 0{1..4}.txt  
> 01.txt 02.txt 03.txt 04.txt
```

```
echo {1..3}{a..c}  
> 1a 1b 1c 2a 2b 2c 3a 3b 3c
```