

Bash

Linuxowy command line



**Fundusze
Europejskie**
Program Regionalny



URZĄD MARSZAŁKOWSKI
WOJEWÓDZTWA POMORSKIEGO

Unia Europejska
Europejski Fundusz Społeczny



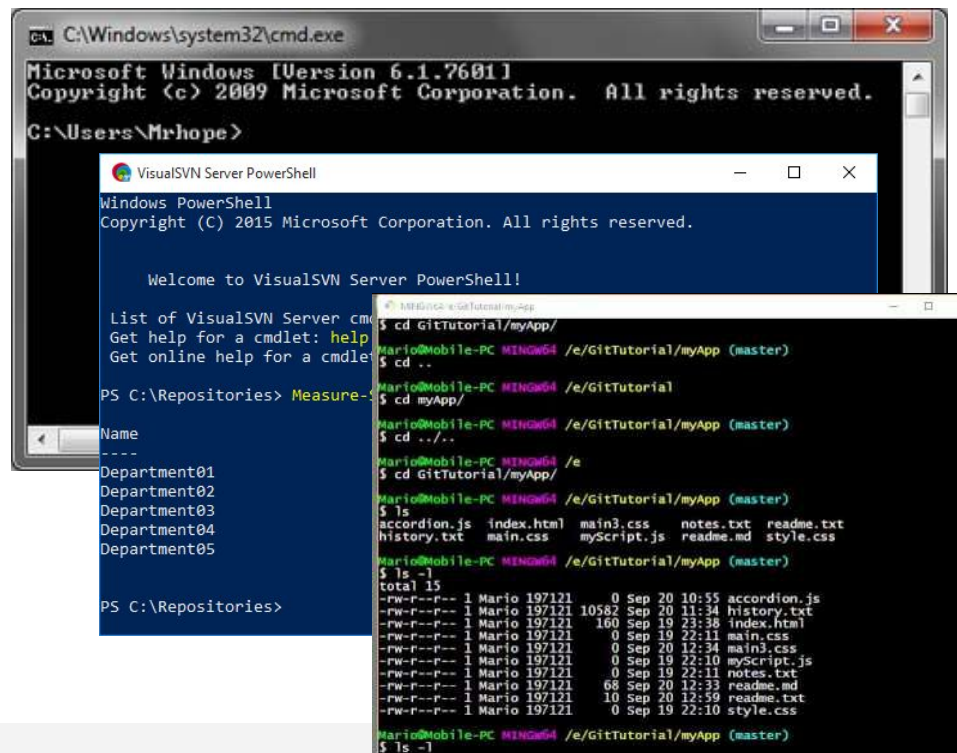
Shell, Bash - co to takiego?

Shell - aplikacja umożliwiająca wykonywanie poleceń command line, czyli instrukcji wpisywanych z klawiatury.

W Windows mamy kilka możliwości skorzystania z command line:

- Windows Command Line - cmd.exe
- PowerShell
- Zewnętrzne aplikacje, jak np. GitBash

Bash - Bourne-Again Shell - specyficzna odmiana shella, wywodząca się z systemu Linux



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Mrhope>

VisualSVN Server PowerShell
Windows PowerShell
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

Welcome to VisualSVN Server PowerShell!

List of VisualSVN Server cmdlets:
Get help for a cmdlet: help
Get online help for a cmdlet: help -online

PS C:\Repositories> Measure-Command { cd C:\Repositories\myApp }
Name
-----
Department01
Department02
Department03
Department04
Department05

PS C:\Repositories>

C:\Repositories\myApp> cd ..
C:\Repositories> cd myApp/
C:\Repositories\myApp> ls
accordions.js  index.html  main3.css  notes.txt  readme.txt
history.txt    main.css   myScript.js  readme.md  style.css

C:\Repositories\myApp> ls -l
total 15
-rw-r--r-- 1 Mario 197121 0 Sep 20 10:55 accordions.js
-rw-r--r-- 1 Mario 197121 10582 Sep 20 11:34 history.txt
-rw-r--r-- 1 Mario 197121 160 Sep 19 23:38 index.html
-rw-r--r-- 1 Mario 197121 0 Sep 19 22:11 main.css
-rw-r--r-- 1 Mario 197121 0 Sep 20 12:34 main3.css
-rw-r--r-- 1 Mario 197121 0 Sep 19 22:10 myScript.js
-rw-r--r-- 1 Mario 197121 0 Sep 19 22:11 notes.txt
-rw-r--r-- 1 Mario 197121 68 Sep 20 12:33 readme.md
-rw-r--r-- 1 Mario 197121 10 Sep 20 12:59 readme.txt
-rw-r--r-- 1 Mario 197121 0 Sep 19 22:10 style.css
```

Bash

W tym tutorialu zajmiemy się shellem Bash. Przykłady będą podawane z systemu operacyjnego Linux, natomiast wszystkie ćwiczenia można wykonać również w systemie Windows, korzystając np. z GitBash.

Na początek proste polecenia:

ls - wyświetla zawartość katalogu

cal - wyświetla kalendarz

date - wyświetla aktualną datę

pwd - wyświetla bieżącą ścieżkę w systemie plików

```
marek@marek-ThinkPad-T480: ~  
File Edit View Search Terminal Help  
marek@marek-ThinkPad-T480:~$ ls  
Desktop  dotnet  Music    readme.vbox  snippet  VMs  
devtools Downloads Pictures  sds         Templates  Workspace  
Documents java     Public   sign-vbox-modules  Videos  
marek@marek-ThinkPad-T480:~$ cal  
      June 2019  
Su Mo Tu We Th Fr Sa  
                1  
 2  3  4  5  6  7  8  
 9 10 11 12 13 14 15  
16 17 18 19 20 21 22  
23 24 25 26 27 28 29  
30  
marek@marek-ThinkPad-T480:~$ date  
Sun Jun 30 09:24:22 CEST 2019  
marek@marek-ThinkPad-T480:~$ pwd  
/home/marek  
marek@marek-ThinkPad-T480:~$
```

Podstawowe komendy

Operacje na plikach - ls

Polecenie ls powoduje wyświetlenie zawartości katalogu. Możemy je wykonać bez żadnych parametrów, wtedy po prostu zostaną wyświetlone nazwy plików i katalogów. Dodatkowe parametry powodują wyświetlenie dodatkowych informacji lub wyświetlenie katalogu w innym formacie.

Spróbuj wykonać poszczególne polecenia:

ls -l - wyświetlenie katalogu w postaci listy

ls -a - wyświetlenie plików i katalogów ukrytych (w systemach unix/linux zaczynających się od .)

Parametry można łączyć, np. połączenie **ls -al**

Pełną listę parametrów można zobaczyć wpisując

ls --help

```
marek@marek-ThinkPad-T480:~$ ls -l
total 68
drwxr-xr-x 2 marek marek 4096 Jun 21 19:37 Desktop
drwxrwxr-x 3 marek marek 4096 Jun 21 20:49 devtools
drwxr-xr-x 4 marek marek 4096 Jun 28 21:36 Documents
drwxrwxr-x 3 marek marek 4096 Jun 22 17:00 dotnet
drwxr-xr-x 2 marek marek 4096 Jun 30 08:55 Downloads
```

Przechodzenie między katalogami

Polecenie, które służy do przechodzenia między katalogami to

cd

Np.

cd /etc **cd ~** **cd /home/user** **cd ..** **cd ../../..**

Samo polecenie **cd** powoduje przejście do katalogu domowego użytkownika i jest równoważne z poleceniem **cd ~**

Katalogi specjalne

~ - katalog domowy bieżącego użytkownika (równoważny ścieżce `/home/<user_name>`).

/ - root - główny katalog systemowy, najwyższego poziomu

. - bieżący katalog

.. - katalog wyższego poziomu

W Linuxie nie ma oznaczenia dysków takich jak `c:\` czy `d:\`. Zamiast tego dyski mogą być zamontowane w jednym z katalogów. W GitBashu pod Windows są zamontowane w katalogach **/c/** **/d/**



Operacje na plikach - atrybuty plików

```
marek@marek-ThinkPad-T480:~/Workspace/trainingta$ ls -al
total 64
drwxr-xr-x 8 marek marek 4096 Jun 30 07:08 .
drwxr-xr-x 4 marek marek 4096 Jun 22 21:52 ..
drwxr-xr-x 7 marek marek 4096 Jun 24 07:27 build
-rw-r--r-- 1 marek marek 2162 Jun 30 07:08 build.gradle
drwxr-xr-x 8 marek marek 4096 Jun 30 07:41 .git
-rw-r--r-- 1 marek marek 1534 Jun 21 21:30 .gitignore
-rw-r--r-- 1 marek marek 199 Jun 21 21:30 .gitlab-ci.yml
drwxr-xr-x 3 marek marek 4096 Jun 21 21:30 gradle
drwxr-xr-x 5 marek marek 4096 Jun 21 21:30 .gradle
-rwxr-xr-x 1 marek marek 5296 Jun 21 21:30 gradlew
```

Prawa do pliku:

r - odczyt (read)

w - zapis (write)

x - uruchomienie (execute)

Właściciel	Grupa	Publiczne
rwx	rwx	rwx

chmod - polecenie zmiany uprawnień

chown - polecenie zmiany właściciela

Po wydaniu komendy `ls -al` pokazuje nam się zawartość katalogu z dodatkowymi informacjami:

- atrybuty pliku / katalogu
 - pierwsza litera
 - - zwykły plik
 - **d** - katalog
 - **c**, **b** - pliki specjalne
 - **l** - link symboliczny (skrót do pliku)
 - kolejne bloki po 3 litery:
 - prawa do pliku odpowiednio dla właściciela, grupy, publiczne
- użytkownik
- grupa (czasami taka sama jak użytkownik)
- wielkość w bajtach
- data modyfikacji pliku
- nazwa pliku.

Operacje na plikach - zmiana atrybutów

1. **touch** - Utworzenie (pustego) pliku lub uaktualnienie daty modyfikacji

Np. **touch dummy** - pierwsze wykonanie tworzy plik dummy w bieżącym katalogu

touch dummy - ponowne wykonanie aktualizuje datę modyfikacji pliku

```
$ touch dummy
$ ls -al                -   został utworzony plik (o zerowej długości) o nazwie dummy
-rw-r--r-- 1 marek marek   0 Jun 30 20:18 dummy
```

2. **chmod** - Zmiana atrybutów

Wykorzystując utworzony plik z poprzedniego ćwiczenia możemy zmienić atrybuty tak utworzonego pliku.

chmod 777 dummy - wszystkie możliwe uprawnienia

```
-rwxrwxrwx 1 marek marek   0 Jun 30 20:18 dummy
```

chmod 600 dummy - uprawnienia tylko do odczytu i

```
-rw----- 1 marek marek   0 Jun 30 20:18 dummy
```

zapisu tylko dla właściciela pliku

Inne przykłady użycia: **chmod o+x**, **chmod g=wr**

Dokładniejszy opis polecenia i wyjaśnienie wartości liczbowych: <https://pl.wikipedia.org/wiki/Chmod>

Operacje na plikach

3. chmod - zmiana właściciela pliku

Sposób użycia:

```
chown marek:marek dummy
```

```
chown -R marek:marek sample_folder/
```

4. Tworzenie katalogu

```
mkdir sample_folder
```

5. Kopiowanie plików

```
cp dummy sample_folder
```

```
cp -R sample_folder copy_of_sample_folder
```

6. Przenoszenie plików / folderów, zmiana nazwy

```
mv dummy dummy1
```

7. Usuwanie plików / katalogów

```
rm dummy
```

```
rm -f dummy
```

```
rm -rf sample_folder
```

```
rmdir sample_folder
```

8. file - Określenie typu pliku

```
marek@marek-ThinkPad-T480:~/Workspace/trainingta$ file .gitignore
.gitignore: ASCII text
```

```
marek@marek-ThinkPad-T480:~/Workspace/trainingta$ file dummy
dummy: empty
```

```
marek@marek-ThinkPad-T480:~/Pictures$ file Selection_007.png
Selection_007.png: PNG image data, 433 x 125, 8-bit/color RGB, non-interlaced
```


Wyświetlanie zawartości plików

Na początek ciekawostka: za pomocą operatora `>` możemy przekierować wyjście z dowolnego polecenia z konsoli do wskazanego pliku. Spróbujmy wykonać:

```
ls -al > dummy
```

Wyświetlanie zawartości pliku

1. `cat`

`cat dummy` - spowoduje wyświetlenie zawartości pliku `dummy`. I jak, mamy zawartość katalogu?

2. `more`

Teraz poznamy jeszcze jedną ciekawostkę basha:

`for i in {1..10}; do ls -al; done > dummy` spowoduje wykonanie w pętli 10 razy polecenia `ls -al` i przekierowanie do pliku `dummy`. Dzięki temu mamy całkiem długą zawartość tego pliku. Teraz wykonaj:

`more dummy` - spowoduje wyświetlenie pliku z podziałem na stronę.

spacją przechodzimy na kolejną stronę, po naciśnięciu `q` kończymy podgląd.

3. `less`

`less dummy` - działa podobnie, ale możemy używać strzałek w dół/górę do nawigacji

```
-rw-r--r-- 1 marek marek 40 Jun 30 07:08 settings.gradle
drwxr-xr-x 4 marek marek 4096 Jun 21 21:30 src
total 72
drwxr-xr-x 8 marek marek 4096 Jun 30 20:18 .
--More-- (80%)
```

Wyszukiwanie tekstu - grep

Bardzo przydatnym poleceniem jest **grep**. Służy ono do wyszukiwania wystąpienia wskazanego tekstu w plikach lub przekierowanym tekście.

Przykłady użycia:

grep dummy * - wyszukuje pliki zawierające tekst dummy

Operator **|** spowoduje jednoczesne wykonanie dwóch poleceń rozdzielonych tym znakiem

ls -al | grep -i Dummy - wyszukuje wiersze zawierające dummy (ignorując wielkość znaków)

cat dummy | grep "marek marek"

cat dummy | grep '\-rw\-\-' - jeżeli chcemy wyszukać - musimy znieść jego specjalne znaczenie za pomocą ****

cat dummy | grep -n -E "total [0-9]+" - **-E** powoduje wyszukanie wyrażenia regularnego **-n** podaje nr linii

```
marek@marek-ThinkPad-T480:~/Workspace/trainingta$ grep dummy *
grep: build: Is a directory
dummy: -rw----- 1 marek marek 0 Jul 1 06:15 dummy
dummy: -rw----- 1 marek marek 829 Jul 1 06:15 dummy
```

```
marek@marek-ThinkPad-T480:~/Workspace/trainingta$ ls -al | grep dummy
-rw----- 1 marek marek 8290 Jul 1 06:15 dummy
```

Edytor tekstu Vim

Edytor Vim (a właściwie Vi - Vim to nowsza, udoskonalona wersja) wzbudza sporo emocji, zwłaszcza wśród początkujących. Dzieje się tak z powodu mało intuicyjnego interfejsu użytkownika. W edytorze Vi mamy 2 podstawowe tryby pracy: **tryb komend** i **tryb wprowadzania tekstu**.

Absolutnym minimum, które trzeba znać z Vima, to:

i - przejście do trybu edycji i możliwość pisania tekstu

Esc - wyjście z trybu edycji

:w - zapis pliku **:q** - wyjście z vima **:wq** - jednocześnie zapis i wyjście z zapisem

:q! - wyjście z edytora z niezapisanymi zmianami (gdy wyedytowaliśmy tekst, ale nie chcemy go zapisać)

Tryb komend jest aktywny zaraz po włączeniu edytora.

Przykładowe komendy:

i - przejście do trybu edycji w bieżącym miejscu

a, A, o, O - przejście do trybu edycji odpowiednio za bieżącym znakiem, na końcu linii, w linii poniżej, linii powyżej

Esc - wyjście z trybu edycji

Edytor tekstu Vim (2)

x - usunięcie bieżącego znaku

dd, 5dd - usunięcie bieżącej linii, 5 kolejnych linii

u - undo, cofnięcie ostatniej operacji

yy - skopiowanie bieżącej linii do schowka

5yy - skopiowanie 5 linii do schowka

p - wstawienie zawartości schowka w linii poniżej

v - tryb wizualny, możemy za pomocą strzałek zaznaczać tekst, który np. skopiujemy do schowka.

Nawigacja w dokumencie (w trybie komend):

:n (zamiast n liczba - **:50**) - przejście do linii nr n

/szukany tekst - przejście do pierwszego wystąpienia szukanego tekstu, n - przejście do następnego wystąpienia

^ - przejście na początek wiersza, **\$** - przejście na koniec wiersza

gg - przejście na początek dokumentu

G - przejście na koniec dokumentu

