Wrocław University of Technology

Fundusze Europejskie Wiedza Edukacja Rozwój
Politechnika Wrocławska
Unia Europejska Europejski Fundusz Społeczny
„ZPR PWr – Zintegrowany Program Rozwoju Politechniki Wrocławskiej”

Języki Skryptowe
Skrypty Windows

Andrzej Siemiński

Wydział Informatyki i Zarządzania
Politechnika Wrocławska
Andrzej.Sieminski@pwr.edu.pl
203C A1

Agenda

- Składnia poleceń: znaki specjalne
- Przełączniki
- Parametry skryptów
- Proste funkcje
- Zmienne systemowe
- Polecenie for
- Tablice
- Pobieranie danych od użytkownika
- Polecenie if
- Przykłady

Składnia poleceń: znaki specjalne

- Znak | używany do tworzenia potoku, standardowe wyjście z procesu lewostronnego jest traktowane jak wejście do procesu prawostronnego
- Znak & bezwarunkowe and. Kolejny proces jest wykowywany zawsze
- Znaki && warunkowe and. Kolejny proces wykonywany gdy poprzedni kończy się 0
- Znaki || podobnie ale poprzedni musi się skończyć z kodem <> 0
- Znaki nawiasów (oraz) powodują traktowanie komend w środku jako pojedynczej komendy
  - (dir * & dir * /D) | sort
  - Linie z katalogami powielone

Przełączniki - switches

- Są w większości jednoliterowe
- Są zaczynają się od /
- Nie rozróżniają małych i dużych liter
- Zapożyczone z innych systemów zachowują ich konwencje

Ciągi znaków

- Składnia: %ciąg:~start,długość%
  - Konwencja powielana w innych zastosowaniach
  - Koniecznie rozdzielać przez ','
  - :~ oznacza wybór ciągu
  - Pierwszy indeks to 0, można przekraczać długość
  - Wartości ujemne odejmują od długości,
  - Nie podanie drugiego parametru to do końca
  - Nie można przejść na początek

```
@set ciag=0123456789
@echo %ciag:~0,0%   ECHO is on, pusty argument daje stan
@echo off
set ciag=0123456789
echo %ciag:~0,1%     0
echo %ciag:~1,-1%    12345678
echo %ciag:~-1,1%    9
echo %ciag:~-3,1%    7
```

Parametry skryptów

- Przekazywanie, zasady dość dziwne, separatory to : spacja, przecinek, średnik, tabulacja, znak równości
- Takie same znaczenie:
  - test.bat a b c d
  - test.bat a,b,c,d
  - test.bat a, b, c, d
  - test.bat a;b;c;d
  - test.bat a=b=c=d
  - test.bat a b,c;,;=d
- Maksymalna liczba to 4000
- Odwołania:
  - %0% nazwa programu lub funkcji
  - %1%, … %9%
  - Kolejne dostępne przez SHIFT

Parametry przykład na shift

Skrypt: liczy.bat

```
@ECHO off
SET liczy=0
CALL :start %*
ECHO %liczy%
ECHO Arg one: %1 & echo %1, %2, etc. bz
EXIT /b

:start
IF -%1-==-- GOTO koniec
ECHO %1 & REM lub cos innego
SET /A liczy+=%1
REM set /a liczy=%liczy%+1
REM SET liczy=%liczy%%1
SHIFT
GOTO start
:koniec
EXIT /b
```

Prosta funkcja (1)

```
call :start %*
```

- Przekazanie wszystkich parametrów

```
if -%1-==-- goto koniec
```

- Sprawdzenie czy parametr istnieje, brak :

```
set /a liczy+=1
REM set /a liczy=%liczy%+1
```

- Alternatywne zwiększanie zmiennej globalnej liczy

```
SET liczy=%liczy%%1
```

- konkatenacja ciągów
- EXIT kończy sesję
- EXIT /B kończy wykonanie skryptu
- EXIT /B kod j.w. ale ustawia ERRORCODE
- Rvalue uzyskane przez ujęcie w % %

```
c:\VC>liczy 1 dwa 29
1
dwa
29
30
Arg one: 1
1, dwa, bz.
```

Prosta funkcja ()

- Zamiast przez zmienną globalną lepiej jest zwracać wartość przez ERRORLEVEL

```
:nowe
echo %0
set kl=10
set /a kl=%kl%*2
set /a kl=%kl%+1
echo %kl%
set wynik=%kl%
exit /b %kl%
```

- Wywołanie

```
call :nowe
echo %ERRORLEVEL%
echo %wynik%
```

- Wynik działania

```
:nowe
21
21
21
```

Zmienne

- Nie ma rozróżnionych dużych i małych liter
- Zastąpienie zmiennej przez wartość (RVALUE) przez ujęcie w % %
- Zmienne systemowe nie powinny być nadpisywane

Zmienne systemowe

%CD%	The current directory, not ending in a slash character if it is not the root directory of the current drive
%TIME%	The system time in HH:MM:SS.mm format.
%DATE%	The system date in a format specific to localization.
%RANDOM%	A generated pseudo-random number between 0 and 32767.
%ERRORLEVEL%	The error level returned by the last executed command, or by the last called batch script.
%CMDEXTVERSION%	The version number of the Command Processor Extensions currently used by cmd.exe.
%CMDCMDLINE%	The content of the command line used when the current cmd.exe was started.

Polecenie FOR (1)

- Służy do warunkowej iteracji po wymienionych elementach

FOR %%parameter IN (set) DO command
np.:
Konsola:
for %i in (1,6,2) do @echo %i
skrypt:
for %%i in (1,6,2) do @echo %%i
źle
- for %%i in (1,6,2) do @echo %i

błąd
for %i in (1,6,2) do @echo %i

- Jest wygodne ale dość specyficzne
- Działanie:
  - Weź sekwencję elementów
  - Przypisz parametrowi jeden z nich (%%parametr)
  - Wykonaj komendę określoną po DO dla tego parametru
  - Powtórz powyższe dla kolejnych elementów sekwencji

Polecenie FOR (2)

- Służy do iteracji, wygodne ale dość specyficzne
- Wersja lista: przechodzenie po wymienionych elementach
  - for %i in (1,6,2) do @echo %i
    - 1
    - 6
    - 2
  - Przy wykonaniu w skrypcie podwoić trzeba %%
  - for %%i in (1,2,3) do echo %%i
- Wersja arytmetyczna (start, krok, granica)
    - for /L %i in (1,6,2) do @echo %i
      - 1
    - for /L %i in (1,2,6) do @echo %i
      - 1
      - 3
      - 5
- Wyliczenie nie tylko liczbowe ale i po ciągach
    - for %f in (ala ma kota) do @echo %f
      - ala
      - ma
      - kota

Polecenie FOR: przetwarzanie tekstu

- For może dzielić tekst i selekcjonować fragmentu tekstu lub zawartość pliku
- Badanie tekstu:
- FOR /F ["options"] %%parameter IN ("Text string to process") DO command
- Wybrane opcje:
- delims=xxx // ogranicznik(i), domyślnie spacja
- skip=n // liczba linii do pominięcia z początku pliku, domyślnie 0
- tokens=n //numery wybieranych tokenów, poczynając od 1, gdy brak to pusty ciąg, można je rozdzielać , 0 powoduje błąd
- parametr // %% w skryptach, % dla konsoli, nazwy jednoliterowe, podaje się tylko pierwszy, nieistniejące literalnie wypisywane

```
c:\VC>FOR /F "tokens= 1,2" %a IN ("Text string to process") DO @ECHO %b
%a
string Text
c:\VC>FOR /F "tokens= 1,8" %a IN ("Text string to process") DO @ECHO %b
%a
 Text
c:\VC>FOR /F "tokens= 1,2" %a IN ("Text string to process") DO @ECHO %b
%a %b %c %d
string Text string %c %d
```

Polecenie FOR - pliki

- Stosować przełącznik /b dla dir – daje listę nazw plików:

```
c:\VC>dir /B *.txt
dane.txt
dane2.txt

for /F %i in ('dir /b *.log') do notepad %i
```

- Selekcja kolumn może być użyta dla zawartości pliku:

```
 for /F "tokens=2,4" %i in (test.txt) do @echo %i %j
c:\VC>FOR /F  "tokens=1" %k in (dane.txt) DO @Echo %k
to
testowe
dla
a
```

- Można podać więcej plików po ,
- czy też:

```
for /F "tokens=2,4 delims=," %i in (test.txt) do @echo %i,%j
```

Warianty polecenie FOR

syntax-FOR-Files
FOR %%parameter IN (set) DO command

syntax-FOR-Files-Rooted at Path
FOR /R [[drive:]path] %%parameter IN (set) DO command

syntax-FOR-Folders
FOR /D %%parameter IN (folder_set) DO command

syntax-FOR-List of numbers
FOR /L %%parameter IN (start,step,end) DO command

syntax-FOR-File contents
FOR /F ["options"] %%parameter IN (filenameset) DO command

FOR /F ["options"] %%parameter IN ("Text string to process") DO command

syntax-FOR-Command Results
FOR /F ["options"] %%parameter IN ('command to process') DO command

Polecenie FOR - przykłady

- Iteracja po plikach
  - for %f in (pliki *.bat) do @echo %f
    - Wszystkie pliki o podanym rozszerzeniu nazwy
    - Możliwość stosowania ? Dla maskowania jednego znaku
  - for %i in (1,2,a*, 1) do @echo %i
    - Gdy nie ma takiego pliku to ignoruje
      - 1
      - 2
      - Ala.txt
      - 1
  - for %i in (1,2,a*, brak) do @echo %i
    - Gdy nie ma takiego pliku to ignoruje
      - 1
      - 2
      - Ala.txt // plikowo
      - Ala1.txt
      - Brak //literalnie
  - for /L %i in (1,2,4, y*, brak) do @echo %i
    - 1
    - 3
  - for %f in (pliki *.bat) do echo %f
  - Specjalne traktowanie systemu plików:
    - for /r /d %i in (*) do @echo %i
      - /r rekursywnie /d tylko foldery
  - for /f "tokens=*" %i in (list.txt) do @echo %i
    - Wypisuje każdą linię z podanego pliku

Polecenie FOR - przykłady

- FOR /F domyślnie przetwarza plik tekstowy linia po linii i rozbija każdą z nich na segmenty (tokeny) rozdzielane domyślnie spacjami. Komenda po DO jest wykonywana dla wybranych z nich.
- Specjalne traktowanie systemu plików:
  - for /r /d %i in (*) do @echo %i
    - /r rekursywnie /d tylko foldery
- for /f "tokens=*" %i in (list.txt) do @echo %i
  - Wypisuje każdą linię z podanego pliku
- Inne wersje tokens:
  - tokens=2,4,6 - przetwarzanie tylko drugiego, czwartego i szóstego elementu.
  - tokens=2-6 – przetwarzanie elementów od drugiego do szóstego.
  - tokens=* will cause all items on each line to be processed.

- FOR /F "tokens=4 delims=," %G IN ("depozyt,4500,123.4,12-AUG-09") DO @echo Zapłacono %G
- Zapłacono 12-AUG-09

Tablice

- Symulowanie przez znaki % i !

```
@echo off
setlocal EnableDelayedExpansion
for /l %%i in (1, 1, 10) do (
  set array_%%i=!random!
)
for /l %%i in (1, 1, 10) do (
  echo !array_%%i!
)
:: For each item in the array, not knowing the length
set i=1
:startloop
if not defined array_%i% goto endloop
set array_%i%=!array_%i%!_dummy_suffix
echo A%i%: !array_%i%!
set /a i+=1
goto startloop
:endloop
```

Po zmianie na 5 stare wartości są zachowane

```
17470
14593
9717
23316
28335
A1: 17470_dummy_suffix
A2: 14593_dummy_suffix
A3: 9717_dummy_suffix
A4: 23316_dummy_suffix
A5: 28335_dummy_suffix
A6: !array_6!_dummy_suffix_dummy_suffix
A7: !array_7!_dummy_suffix_dummy_suffix
A8: !array_8!_dummy_suffix_dummy_suffix
A9: !array_9!_dummy_suffix_dummy_suffix
A10: !array_10!_dummy_suffix_dummy_suffix
```

Pobieranie danych od użytkownika

- Pobranie dowolnego ciągu:
  - >set /p daj= podaj dane spacje na końcu
    - podaj dane to są dane wejściowe
  - >echo %daj%
    - to są dane wejściowe polskie znaki są obsługiwane
- Choice: Wybór jednego z podanych ciągów
  - Zwracany jest numer wybranego ciągu poczynając od 1 wpisując go to zmiennej ERRORLEVEL
  - CTRL+C zwraca 0
  - Nieprawidłowy wybór jest odrzucany, sygnał dźwiękowy
  - /C z czego wybieramy
  - /M tekst do wyświetlenia
    - >CHOICE /C YNC /M "Press Y for Yes, N for No or C for Cancel."
    - Press Y for Yes, N for No or C for Cancel. [Y,N,C]?C
    - echo %ERRORLEVEL%
    - 3
- Pobieranie tekstu z wielu liniami
  - Copy con: > nazwaPliku
  - Kończymy przez CTRL+Z

IF warunkowe wykonywanie poleceń

- Przy wielu poleceniach wykonywanych warunkowo ująć je w ()
- Dostępne testy:
  - exist <filename>
  - <string>==<string>
  - <expression1> equ <expression2> -- equals
  - <expression1> neq <expression2> -- not equal
  - <expression1> lss <expression2> -- less than
  - <expression1> leq <expression2> -- less than or equal
  - <expression1> gtr <expression2> -- greater than
  - <expression1> geq <expression2> -- greater than or equal
  - defined <variable>
  - errorlevel <number>
  - cmdextversion <number>
- Każdy test można poprzedzić przez To each elementary test, "not"
- Nie ma operatorów takich jak AND, OR, etc. By tworzyć warunki złożone.
- Przełącznik /I switch sprawia, że == oraz equ testu nie uwzględniają wielkości liter

Inne polecenia

- FC – file compare nie za bardzo użyteczna
  - fc bada1.txt bada1.txt >NUL && Echo Same || echo Different or error
  - Same
- PAUSE
  - Oczekiwanie na wprowadzenie linii przez użytkownika
- START
  - Uruchamia program w nowym oknie, praca asynchroniczna
- CALL
  - Uruchamia skrypt z wewnątrz skryptu, praca synchroniczna
  - Można też stosować do etykiet, substytut funkcji
- TREE
  - Pokazuje (semigrafika, lub znakowo /a) drzewo katalogów począwszy od bieżącego, z przełącznikiem /f także pliki
- XCOPY src dst
  - Bardziej zaawansowana wersja copy
  - Obecnie zastępowane przez jeszcze bardziej rozbudowane robocopy
  - Możliwość kopiowania plików i katalogów spełniających różne warunki
  - xcopy /s /i /d:09-01-2020 C:\Windows\system C:\Windows-2\system
    - Kopiowanie plików i katalogów na wszystkich poziomach /s
    - Jeżeli kopiowane jest więcej niż 1 plik a cel nie istnieje to zakłada się, że ma to być katalog
    - Kopiowane są tylko pliku zmienione począwszy od pierwszego września 2020, konieczność stosowania konwencji zapisu dat stosowanej w US

Funkcje dokładniej

- Nie ma jawnie definiowanych
- Implementacja przez:
  - Call :etykieta
    - Lepiej niż goto bo sterowanie wraca do miejsca wywołania
  - setlocal
    - Uniknięcie kolizji ze zmiennymi już zdefiniowanymi
  - endlocal
    - J.w. przywraca globalne zmienne

Funkcja do policzania potęgi

```
@echo off
call :power %1 %2
echo %result%
goto :eof

rem __Function power_____________________________
rem Arguments: %1 and %2
:power
setlocal
set counter=%2
set interim_product=%1
:power_loop
if %counter% gtr 1 (
  set /a interim_product=interim_product * %1
  set /a counter=counter - 1
  goto :power_loop
)
endlocal & set result=%interim_product%
goto :eof
```

Funkcje uwagi

- Parametry funkcji mogą być rozdzielane przez spacje = , ;
- Zamiast
  - goto :eof można użyć
  - exit /b
- Ostanie goto :eof nie jest konieczne, potrzeba gdy jest w skrypcie więcej funkcji
- Użyte exit poza funkcją kończy cmd
- Można dodać kod powrotu
- Zwraca to też wartość funkcji:
  - exit /b %interim_product%
  - Odbiór
  - echo %ERRORLEVEL%
  - Powrót do systemu operacyjnego:
    - Wartość 0 poprawne zakończenie
    - Dopuszczalne są wartości ujemne

Set /a

- Działania arytmetyczne na liczbach całkowitych, ze znakiem 32 bitowych
- Domyślnie dziesiętne ale też ósemkowe i szestastkowe (0 0x0)
- Operatory znane z CPP
- Specjalne znaczenie operatorów np. ^ usunięte przez " " lub podwojenie
  - set /a num="255^127"
  - set /a num=255^^127

Liczby pierwsze

wypisz wszystkie liczby pierwsze do wartości parametru

```
@echo off
setlocal
set n=1
:print_primes_loop
set /a n=n+1
if %n% gtr %1 goto :eof
set cand_divisor=1
:print_primes_loop2
set /a cand_divisor=cand_divisor+1
set /a cand_divisor_squared=cand_divisor*cand_divisor
if %cand_divisor_squared% gtr %n% echo Prime %n% & goto :print_primes_loop
set /a modulo=n%%cand_divisor
if %modulo% equ 0 goto :print_primes_loop & REM Not a prime
goto :print_primes_loop2
```

```
c:\VC>primes.bat 10
Prime 2
Prime 3
Prime 5
Prime 7
```

A

- Koniec

Funkcja power

```
@echo off
set result=12
call :power %1 %2
echo %1 do potegi %2 to errollevel %ERRORLEVEL%
echo   result nie jest zmieniony (local) %result%
goto :eof

rem __Function power_____________________
rem Arguments: %1 and %2
:power
setlocal
set counter=%2
set interim_product=%1
:power_loop
if %counter% gtr 1 (
  set /a interim_product=interim_product * %1
  set /a counter=counter - 1
  goto :power_loop
)
:endlocal & set /a rel=%interim_product%
set /a result=%interim_product%
exit /b %interim_product%
REM goto :eof
```

- Wyniki:

```
>power 2 7
2 do potegi 7 to errollevel 128
  result nie jest zmieniony (local) 12
```