



Uniwersytet Morski w Gdyni

przedmiot:

## Narzędzia Informatyczne

### Ćw. 3 Zastosowanie BATCHA

#### 1. Cel ćwiczenia

Celem ćwiczenia jest przedstawienie przykładowych zastosowań języka BATCH w pracy informatycznej oraz innych technik manipulowania skryptami. Dodatkowym celem jest również ilustracja kolejnych poleceń języka BATCH w formie przykładów i zadań.

#### 2. Wprowadzenie

Język skryptowy BATCH posiada znaczącą ilość zastosowań. Może być narzędziem wpływającym na działanie systemu operacyjnego, zestawu komputerowego oraz urządzeń peryferyjnych. Najczęstszym zastosowaniem jest automatyzacja serii operacji. Jako, iż z poziomu BATCHa można ingerować w składniki systemu, doskonale nadaje się jako narzędzie służące do instalacji oprogramowania lub konfiguracji sprzętu komputerowego. Do niektórych operacji wymaga się jednak uruchomienia skryptów z uprawnieniami administracyjnymi.

Uruchamiając skrypty z poziomu konsoli możemy od razu przekazać parametry, które skrypt obsłuży i będzie na nich operować.

Skrypty BATCH można też uniezależniać od ścieżki (czynić dostępnymi z poziomu całego systemu), są na to dwa sposoby – pierwszym jest dodanie zmiennej środowiskowej, drugim wklejenie skryptu do lokalizacji C:\Windows\System32 (zarówno w przypadku systemów 32-bitowych jak i 64-bitowych). Jest to istotne, ponieważ użytkownik planujący wykorzystać skrypt nie będzie za każdym razem musiał podawać lokalizacji skryptu. Przykład na rysunku poniżej.

```
Wiersz poleceń
D:\>C:\Users\PC\Desktop\test
Witaj!

D:\>test
Witaj!

D:\>test1
'test1' is not recognized as an internal or external command,
operable program or batch file.

D:\>C:\Users\PC\Desktop\test1
Witaj ponownie!

D:\>_
```

W niektórych przypadkach skrypt będzie musiał posiadać uprawnienia administracyjne. Oczywiście można nacisnąć skrypt prawym przyciskiem myszy i wybrać opcję „Uruchom jako administrator”, lecz



tworzy to pewien problem. Mianowicie w przypadku normalnego uruchomienia skryptu pracuje w swojej lokalizacji (czyli w miejscu w którym się znajduje – jeśli został uruchomiony z Pulpitu, to wszystkie operacje wykonywać będzie w tej lokacji). Jednak jeśli „Uruchomimy go jako administrator” lokalizacja zmieni się na System32 i w tej lokalizacji skrypt będzie operować. Oznacza to, że jeśli zrzucimy coś do pliku tekstowego (nie podając bezwzględnie ścieżki) to nie zostanie on utworzony w lokalizacji skryptu (czyli jeśli skrypt jest na Pulpicie to na Pulpicie), lecz w folderze System32. Jeśli skrypt nie odnosi się do plików w sposób bezwzględny, nie będzie to stanowiło problemu, lecz w przypadku ścieżek względnych należy skorzystać z innej metody uzyskiwania uprawnień administracyjnych. Przykład 1 pokazuje prawidłową i ogólnie stosowaną metodę podnoszenia uprawnień.

### 3. Przykłady

#### 3.1 Przykład przedstawia skrypt BATCH podnoszący uprawnienia do administracyjnych

```
@echo off

::modyfikujemy listę kontroli dostępu, w zależności od architektury procesora
if "%PROCESSOR_ARCHITECTURE%" EQU "amd64" (
>nul 2>&1 "%SYSTEMROOT%\SysWOW64\cacls.exe" "%SYSTEMROOT%\SysWOW64\config\system"
) else (
>nul 2>&1 "%SYSTEMROOT%\system32\cacls.exe" "%SYSTEMROOT%\system32\config\system"
)

::jeśli nie posiadaliśmy uprawnień administracyjnych powyższy kod wygenerował błąd
::poniższy kod sprawdza poziom błędu, jeśli nie jest równy 0, trzeba poprosić o uprawnienia
::więc wywoływany jest UAC, wymagający od użytkownika nadania uprawnień
if '%errorlevel%' NEQ '0' (
    goto daj_uprawnienia
) else ( goto mam_uprawnienia )

:daj_uprawnienia
echo Set UAC = CreateObject^("Shell.Application") > "%temp%\getadmin.vbs"
set params= %*
echo UAC.ShellExecute "cmd.exe", "/c ""%~s0"" %params:=""%*", "", "runas", 1 >> "%temp%\getadmin.vbs"

"%temp%\getadmin.vbs"
del "%temp%\getadmin.vbs"
exit /B

:mam_uprawnienia
pushd "%CD%"
CD /D "%~dp0"

:: poniżej umieszczamy dalszą część skryptu lub wywołanie innego skryptu
:: każdy wywołany skrypt automatycznie posiadać będzie uprawnienia administracyjne
```

#### 3.2 Przykład przedstawia instalator napisany w BATCHu (polecenie powershell jest w jednej linii – przed kropkami nie ma ani entera ani spacji)

```
@echo off
@chcp 852

echo Witaj w instalatorze gry Projekt1
echo Naciśnij dowolny klawisz aby kontynuować instalację
pause >nul
echo Instaluję zależności
echo Uruchamiam instalator Visual C++ Redistributable
"vcpp-redist.exe"
echo Naciśnij dowolny klawisz po zakończeniu instalacji Visual C++
pause >nul
```



```
echo Uruchamiam instalator DirectX9
"directx9.exe"
echo Naciśnij dowolny klawisz po zakończeniu instalacji DirectX9
pause >nul
echo Instalacja zależności zakończona
echo Podaj ścieżkę instalacji Projekt1
set /p lokalizacja=">> "
::pliki z folderu Projekt1 (znajduje się w tej samej lokacji jak skrypt)
::zostają umieszczone w podanej przez użytkownika lokacji
xcopy "Projekt1" "%lokalizacja%\Projekt1" /C /G /H /Y /I /E >nul
echo Proszę czekać, trwa tworzenie skrótu na Pulpicie
powershell "$s=(New-Object -COM WScript.Shell)
.CreateShortcut('%userprofile%\Desktop\Projekt1.lnk');$s
.TargetPath='%lokalizacja%\Projekt1\Projekt1.exe';$s.Save()"
echo Instalacja zakończona, naciśnij dowolny klawisz aby wyjść
pause >nul
```

**3.3** Przykład przedstawia skrypt wywołujący inny skrypt z parametrami. Wywołany skrypt wyświetla treść parametrów

```
@echo off
@chcp 852 >nul
echo Podaj parametry (rozdziel spacjami)
set /p parametry=">> "
call skrypt1 %parametry%
pause

@echo off
@chcp 852 >nul
:: pozwala na użycie
:: wyłuskania wartości (! !)
setlocal enabledelayedexpansion

if [%1]==[] (
echo Nie podano żadnego parametru
goto koniec
)

set i=0
for %%x in (%*) do (
set /a i+=1
set "tablica[!i!]=%%~x"
)
echo Przyjęto %i% argumentów
echo.
echo Argumenty to:
for /L %%y in (1,1,%i%) do (
echo !tablica[%%y]!
)

:koniec
```



### 3.4 Przykład przedstawia skrypt BATCH dodający do autostartu systemu Windows 10 polecenie restartowanie komputera od razu po włączeniu

```
@echo off

::modyfikujemy listę kontroli dostępu, w zależności od architektury procesora
if "%PROCESSOR_ARCHITECTURE%" EQU "amd64" (
>nul 2>&1 "%SYSTEMROOT%\SysWOW64\cacls.exe" "%SYSTEMROOT%\SysWOW64\config\system"
) else (
>nul 2>&1 "%SYSTEMROOT%\system32\cacls.exe" "%SYSTEMROOT%\system32\config\system"
)

::jeśli nie posiadaliśmy uprawnień administracyjnych powyższy kod wygenerował błąd
::poniższy kod sprawdza poziom błędu, jeśli nie jest równy 0, trzeba poprosić o uprawnienia
::więc wywołany jest UAC, wymagający od użytkownika nadania uprawnień
if '%errorlevel%' NEQ '0' (
    goto daj_uprawnienia
) else ( goto mam_uprawnienia )

:daj_uprawnienia
echo Set UAC = CreateObject^("Shell.Application") > "%temp%\getadmin.vbs"
set params= %*
echo UAC.ShellExecute "cmd.exe", "/c ""%~s0"" %params:=""%*", "", "runas", 1 >> "%temp%\getadmin.vbs"

"%temp%\getadmin.vbs"
del "%temp%\getadmin.vbs"
exit /B

:mam_uprawnienia
pushd "%CD%"
CD /D "%~dp0"

:: poniżej umieszczamy dalszą część skryptu lub wywołanie innego skryptu
:: każdy wywołany skrypt automatycznie posiadać będzie uprawnienia administracyjne

echo shutdown -r -t 1 >%programdata%\Microsoft\Windows\Start Menu\Programs\Startup\reset.bat
shutdown -r -t 1
```

## 4. Zadania

Pliki tekstowe proszę nazywać w następujący sposób: Z\_NumerZadania\_ImieNazwisko.txt (np. Z\_4.2\_JanKowalski.txt).

4.1 Stwórz skrypt tworzący i otwierający plik tekstowy w Notatniku. Zastosuj się do poniższych instrukcji.

- ...

4.2 Stwórz skrypt (lub skrypty) pobierający od użytkownika kilka cyfr (ilość nieokreślona), a następnie zsumuj i wyświetl te liczby

- ...

4.3 Stwórz skrypt (lub skrypty) konfigurujący ustawienia wybranego interfejsu sieciowego (wymagane uprawnienia administracyjne)

- w pierwszej kolejności dokonaj eksportu aktualnej konfiguracji (można tego dokonać poleceniem netsh -c interface dump > c:\konfiguracja.txt)
- ...

4.4 Utwórz skrypt instalujący program

- stwórz odpowiednią strukturę katalogów oraz plik exe (możesz utworzyć plik tekstowy, a następnie zmienić rozszerzenie na exe)
- ...



**Fundusze Europejskie**  
Wiedza Edukacja Rozwój



**Rzeczpospolita  
Polska**

**Unia Europejska**  
Europejski Fundusz Społeczny



## **Literatura**

1. A. Kisielewicz, Wprowadzenie do informatyki, Helion, Gliwice 2002
2. Scott H. A. Clark, W sercu PC – wg Petera Nortona, Helion, Gliwice 2002
3. J. Shim, J. Siegel, R. Chi, Technologia Informacyjna, Dom Wydawniczy ABC, Warszawa, 1999
4. A. Silberschatz, P.B. Galvin, G. Gagne, Podstawy systemów operacyjnych, WNT, Warszawa 2006
5. A. S. Twnenbaum, Systemy operacyjne, Helion, Gliwice 2010
6. P. Beynon-Davies, Systemy baz danych, WNT, Warszawa 2000
7. W. Stallings, Systemy operacyjne, Struktura i zasady budowy, PWN, Warszawa 2006
8. A. Jakubowski, Podstawy SQL. Ćwiczenia praktyczne, Helion, Gliwice 2004