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
Linia komend

Andrzej Siemiński

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

Agenda

- Zalety stosowania linii komend
- Wady używania linii komend
- Kiedy i dlaczego jest w użyciu
- Makra DOSkey
- System plików
- Przeadresowanie standardowego wejścia-wyjścia
- Wybrane polecenia
- Przetwarzanie potokowe

Podstawy

- Dlaczego stosować linię komend
- Interfejs graficzny ułatwia pracę dla prostych przypadków.
- W bardziej skomplikowanych problemem jest znalezienie lokalizacji wywołania.
  - Makra nie wszędzie są możliwe i są zależne od programu (zmiana ułożenia kontrolek)
  - Personalizacja przez pasek narzędzi zwykle jest możliwa tylko lokalnie
- Trudniej w ten sam sposób administrować
  - Automatyzacja pozwala na obsługę masową
  - Język jest bardziej elastyczny
  - Język pozwala na wszystkie zadania
  - Język jest bardziej stabilny
  - Wsteczna kompatybilność
- Ale trzeba go się nauczyć
- Praca w dwu trybach:
  - Interakcyjnym
  - skryptowym

Tryb interakcyjny

- Ikona Windows + R
- Zmiana katalogu na właściwy cd
- Tryb buforowany pobierania poleceń: do czasu naciśnięcia Enter możliwość poprawy
- Pisane jest uciążliwe ale są ułatwienia:
  - UP oraz DOWN ARROWS pokazuje komendy ESC czyści linię komend
  - F7 pokazuje okno z komendami do wyboru
  - ALT+F7 kasuje historię komend
  - Doskey: operowanie na wprowadzonych komendach
    - \history;
    - \makros
    - \listsize=size
  - Tab uzupełnia nazwy plików

Tryb interakcyjny inne skróty

- Esc: kasowanie wprowadzanej komendy
- F1: kolejne znaki poprzednio wykonanej komendy
- F6: zastępuje Ctrl+Z (koniec strumienia danych)
- Nie mylić z Ctrl+C
- F8: dla podanego już ciągu pokazuje polecenie z historii z takim prefiksem
- F9: oddanie mumeny polecenia z histrii do wykonania
- Alt+F7: kasowanie historii poleceń

Makra Doskey

- Są aktywne tylko w ramach sesji ale można je zapisać do pliku
- Bazowe:
  - DOSKEY [macroname=[command]]
    - doskey d=dir /OS
    - d
  - Wprowadzenie nowego makra usuwa stare
  - Nazwa Makra= usuwa z pamięci
  - Można wprowadzać kilka poleceń w linii rozdzielając je przez $T
    - DOSKEY nd=md 1Tcd $1
  - Nazwy makr mogą się pokrywać z nazwami poleceń
  - Zapamiętanie i odtworzenie
    - DOSKEY /makros > filename
    - DOSKEY /macrofile=filename
  - Silny mechanizm, ostrożnie stosować

Makra Doskey- przydatne opcje

Parameter	Opis
/reinstall	Instaluje nową kopię Doskey.exe i czyści bufor historii poleceń.
/listsize=<size>	Określa maksymalną liczbę poleceń w buforze historii.
/macros	Wyświetla wszystkie makra doskey. Możesz użyć symbolu przekierowania (>) z / macros, aby przekierować listę do pliku. Skrót: / m.
/macros:all	Wyświetla makra doskey dla wszystkich plików wykonywalnych
/macros:<exename>	Wyświetla makra doskey dla pliku wykonywalnego określonego przez exename.
/history	Wyświetla wszystkie polecenia przechowywane w pamięci. Możesz użyć symbolu przekierowania (>) z / history, aby przekierować listę do pliku. Możesz skrócić / history jako / h.
/exename=<exename>	Określa plik wykonywalny dla którego działa makro doskey.
/macrofile=<filename>	Określa plik zawierający makra, które chcesz zainstalować.
<macroname>=[<text>]	Tworzy makro, które wykonuje polecenia określone przez Text. MacroName określa nazwę, którą chcesz przypisać do makra. Tekst określa polecenia, które chcesz nagrać. Jeśli tekst pozostanie pusty, parametr MacroName zostanie usunięty ze wszystkich przypisanych poleceń. Komendey rozdzielaj przez $t

Makra Doskey- przykłady

- doskey /macros > macinit zapisanie makr do pliku macint
- doskey /macrofile=macinit otworzenie makr
- doskey /history> tmp.bat
- doskey tx=cd temp$tdir/w $* dwie komendy: przejście do foldera temp i pokazanie jego zawartości w trybie /w, akceptacja dodatkowych parametrów
- doskey mc=md 1tcd $1 parametr z linii polecenia
- mc books przykład użycia
- doskey /exename=ftp.exe go=open 172.27.1.100$tmget *.TXT c:\reports$tbye
- doskey qf=format $1 /q /u
- qf a:

System plików

- Windows przyjął nazewnictwo z DOS
- Różnice w stosunku do UNIXopodobnych systemów operacyjnych
  - / oraz \ (JVM)
  - Nazwy napędów
  - Nierozróżnianie dużych i małych liter
- Nazwa pliku:
  - [ścieżka dostępu]\nazwa.rozszerzenie
  - Ścieżka
  - [napęd] \ …
  - . Bieżący katalog
  - .. Katalog nadrzędny
- Zmiana bieżącego napędu jest wykonywane do bieżącego katalogu
  - Nazwa napędu:
- Polecenia
    - Cd : zmiana bieżącego katalogu

Polecenie dir

- DIR [drive:][path][filename] [/A[[:]attributes]]
- Przydatne atrybuty:
- /A Displays files with specified attributes.
  attributes D Directories R Read-only files
  H Hidden files A Files ready for archiving
  S System files I Not content indexed files
  L Reparse Points O Offline files
  - Prefix meaning not
- sortorder
  - N By name (alphabetic) S By size (smallest first)
  - E By extension (alphabetic) D By date/time (oldest first)
  - G Group directories first
  - Prefix to reverse order
- /T Controls which time field displayed or used for sorting
- /S Displays files in specified directory and all subdirectories.
- timefield
  - C Creation
  - A Last Access
  - W Last Written

Dir przykład

```
C:\Users\siemi\source>dir e:\TestFolder
 Volume in drive E is My Passport
 Volume Serial Number is 74B7-6DCC

 Directory of e:\TestFolder

06.02.2020  11:25    <DIR>          .
06.02.2020  11:25    <DIR>          ..
06.02.2020  11:25                 0 fileLevel1.txt
06.02.2020  11:24    <DIR>          Level1a
06.02.2020  11:24    <DIR>          Level1b
               1 File(s)              0 bytes
               4 Dir(s)  943 176 441 856 bytes free
```

Dir przykład

```
C:\Users\siemi\source>dir e:\TestFolder /s
 Volume in drive E is My Passport
 Volume Serial Number is 74B7-6DCC

 Directory of e:\TestFolder

06.02.2020  11:25    <DIR>          .
06.02.2020  11:25    <DIR>          ..
06.02.2020  11:25                 0 fileLevel1.txt
06.02.2020  11:24    <DIR>          Level1a
06.02.2020  11:24    <DIR>          Level1b
               1 File(s)              0 bytes

 Directory of e:\TestFolder\Level1a

06.02.2020  11:24    <DIR>          .
06.02.2020  11:24    <DIR>          ..
               0 File(s)              0 bytes

 Directory of e:\TestFolder\Level1b

06.02.2020  11:24    <DIR>          .
06.02.2020  11:24    <DIR>          ..
06.02.2020  11:25    <DIR>          Level2
               0 File(s)              0 bytes
....
```

Polecenie tree

- Graficzna reprezentacja struktury katalogu

TREE [drive:][path] [/F] [/A]
  /F Display the names of the files in each folder.
  /A Use ASCII instead of extended characters.

```
C:\Users\siemi\source>tree e:\TestFolder /f
Folder PATH listing for volume My Passport
Volume serial number is 74B7-6DCC
E:\TESTFOLDER
│   fileLevel1.txt
│
├───Level1a
└───Level1b
    └───Level2
            fileLevel2.txt
C:\Users\siemi\source>tree e:\TestFolder /f /a
Folder PATH listing for volume My Passport
Volume serial number is 74B7-6DCC
E:\TESTFOLDER
|   fileLevel1.txt
|
+---Level1a
\---Level1b
    \---Level2
            fileLevel2.txt
```

Przeadresowanie wejścia-wyjścia

- Każdy program dostaje od systemu 3 pliki do obsługi standardowego
  - wejścia (klawiatura)
  - wyjścia (ekran)
  - błędów (ekran)
- Są one automatycznie otwierane i zamykane
- System traktuje je jako pliki o dostępnie sekwencyjnym
- Na niskim przydzielone im są im deskryptory
  - 0 (wejście),
  - 1 (wyjście),
  - 2 (błędy)
- Na wysokim poziomie to w CPP jest to odpowiednio:
  - std:in std:out, std:cerr
- Są one kontrolowane przez system zatem może on przekierować strumień danych do nie standardowego pliku lub urządzenia.

Obsługa plików

- Niski poziom
  - Funkcje open, close
  - Dostęp przez deskryptor pliku (int)
  - Operacje są wykonywane bezpośrednio na danych
- Wysoki poziom
  - Funkcje fopen, fclose …
  - Dostęp poprzez strukturę FILE
  - Operacje wejścia wyjścia są buforowane i konieczność opróżnienia bufora np. przy:
    - Zmianie trybu pracy czytanie na pisanie
    - Zmiana położenia kursora odczytu/zapisu pliku fseek, ftell
  - Każda funkcja wysokiego poziomu jest realizowana za pomocą funkcji niskiego poziomu
  - Możliwość uzyskania deskryptora pliku fileno(FILE *stream)

Przeadresowanie: operatory

- Dostępne:
  - > zapis do pliku z kasowaniem jego zawartości
  - >> dopisanie do pliku nowej zawartości, tworzenie jak jest to konieczne, zapisy logów
  - < przesłanie zawartości pliku na standardowe wejście
- Poziom wysoki: adresowanie prze nazwy plików
- Poziom niski: adresowanie przez deskryptory, systemowe lub własne (int fileno(FILE *stream)
- Odwołania się do err tylko przez numer (2)
- Możliwe jest przekierowanie wielokrotne
  - Polecenie > nazwaPliku
  - Polecenie >> nazwaPliku
  - Polecenie < plikWejsciowe
  - Polecenie <plikWejsciowy >plikWyjsciowy

Wymienność i elastyczność

- Możliwość określenia strumienia na różne sposoby.
- Wymienność określenia przez nazwę i przez przekierowanie
  - sort < a.sort > b.sort
  - sort a.sort /o b.sort
  - polecenie 2>&1

    Przekierowanie standardowego wyjścia błędów, jak i standardowego wyjścia do pliku
  - (polecenie > plikWyjsciowy) 2> plikBledow

    (Przekierowanie standardowego wyjścia błędów do pliku p1, a standardowego wyjścia do pliku do pliku p2)

Sort

- O ile nie podamy inaczej to sortuje:
  - linie ze standardowego wejścia
  - w porządku alfabetycznym
  - Na standardowe wyjście
- Przydatne przełączniki:
  - /R[EVERSE] Reverses the sort order;
  - /O[UTPUT]
    [drive3:][path3]filename3 Specifies the file where the sorted input is to be stored.
  - /+n Specifies the character number, n, to begin each comparison.
  - /REC[ORD_MAXIMUM] characters Specifies the maximum number of characters in a record (default 4096, maximum 65535).

Find

- Podstawowa selekcja linii z pliku/standardowego wejścia

```
FIND [/V] [/C] [/N] [/I] [/OFF[LINE]] "string"
[[drive:][path]filename[ ...]]

  /V         Displays all lines NOT containing the specified string.
  /C         Displays only the count of lines containing the string.
  /N         Displays line numbers with the displayed lines.
  /I         Ignores the case of characters when searching for the
string.
  /OFF[LINE] Do not skip files with offline attribute set.
  "string"   Specifies the text string to find.
  [drive:][path]filename
             Specifies a file or files to search.
```

Findstr – zaawansowana selekcja, przydatne przełączniki

- /B Matches pattern if at the beginning of a line.
- /E Matches pattern if at the end of a line.
- /L Uses search strings literally.
- /R Uses search strings as regular expressions.
- /S Searches for matching files in the current directory and all subdirectories.
- /I Specifies that the search is not to be case-sensitive.
- /X Prints lines that match exactly.
- /V Prints only lines that do not contain a match.
- /N Prints the line number before each line that matches.
- /M Prints only the filename if a file contains a match.
- /F:file Reads file list from the specified file(/ stands for console).

Wyrażenia regularne w findstr

. Wildcard: any character

* Repeat: zero or more occurrences of previous character or class

^ Line position: beginning of line

$ Line position: end of line

[class] Character class: any one character in set

[^class] Inverse class: any one character not in set

[x-y] Range: any characters within the specified range

\x Escape: literal use of metacharacter x

\<xyz Word position: beginning of word

xyz\> Word position: end of word

Łączenie komend

- Operator && łączący polecenia a oraz b
  - Polecenie b będzie wykonane tylko jeśli polecenie a zakończy się sukcesem tzn. kod powrotu == 0
  - DIR myfile.txt >NUL 2>NUL && TYPE myfile.txt
  - **dir myfile.txt**
  - **Volume in drive C is Windows-SSD**
  - **Volume Serial Number is E211-7D12**
  - **Directory of c:\Javy\2019\AntBaza**
  - **File Not Found**
- Operator || łączący polecenia a i b
  - Polecenie b będzie wykonane jeśli polecenie a zakończy się z kodem !=0
  - **DIR myfile.txt >NUL 2>NUL || echo Brak pliku - myfile.txt**
  - **Brak pliku - myfile.txt**

Łączenie komend

- Operator & łączący polecenia a oraz b
  - Polecenie b będzie wykonane niezależnie od kodu powrotu polecenia a
- Operator | powoduje wykonanie zawsze poleceń a oraz b
  - Polecenia a oraz b są wykonanie równocześnie
  - System automatycznie łączy standardowe wyjście polecenia a ze standardowym wejściem polecenia b
  - Synchronizacja pracy wszystkich poleceń następuje automatycznie

Findstr + pipeline

- Siłę widać przy wyszukiwaniu wieloetapowym
- Niestandardowe filtry:
  - Line2Words :
    - zapisuje każdy wyraz jako oddzielną linie
    - pomija wszystko inne
  - LC:
    - wypisuje liczbę linii w pliku
- Odszukać liczbę w pliku Nostromo.txt liczbę wyrazów:
  - Zaczynających się od litery "N"
  - Kończących się na literę "y"
  - Pochodzących z linii:
    - zawierających co najmniej jedno wystąpienie ciągu ui

Łącze: przykład

- Wynikowa polecenie:

findstr "ui" <Nostromo.txt | Line2Words.exe | findstr /B "N" | findstr /E "y"| lc
1

- Inkrementacyjne tworzenie polecenia

findstr "ui" <Nostromo.txt
Linda, disappointed, went out quietly; and Giselle sat on
her eyes. The tranquillity of that girlish figure exasperated the
illness and anguish in her face.

findstr "ui" < Nostromo.txt| Line2Words.exe | Findstr "N" |sort /R
Now
Nothing
Not
Not
Not
Nostromo
Nostromo