Podstawy programowania w języku skryptowym Bash

Karol Wszelaki, Rafał Wojtyna

Nokia

Wrocław, November 3, 2016

NOKIA

Agenda

NOKIA

Agenda

NOKIA

bash

bash, powłoka systemowa UNIX napisana dla projektu GNU.

NOKIA

Podstawowe informacje

- dostarcza język skryptowy
- jest rozszerzeniem powłoki **sh**,
- bazuje na powłoce **Korna** i **csh**,
- posiada historię poleceń, stos katalogów, wiele użytecznych zmiennych środowiskowych,
- w trybie interaktywnym zapewnia uzupełnianie nazw plików, poleceń wbudowanych i zewnętrznych (klawisz TAB)
- umożliwia wykonywanie operacji arytmetycznych na liczbach całkowitych
- pozwala na przekierowanie wejścia/wyjścia
- wspiera obsługę wyrażeń regularnych, pętli, itp.

NOKIA

Bash jako język spryptowy

- zapewnia interfejs do systemu operacyjnego
- służy do pracy w konkretnym środowisku
- wykorzystywany jest do automatyzacji zadań
- opiera się na użyciu narzędzi dostępnych w systemie
- nie jest językiem ogólnego zastosowania jak np. Python, Perl

NOKIA

Czemu Bash? (zalety)

- zapewnia integrację z systemem
- ułatwia korzystanie z gotowych narzędzi dostarczanych przez system
- zapewnia dużą szybkość działania, narzędzia systemowe są przeważnie napisane w C, dobrze zoptymalizowane i dojrzałe
- umożliwia pisanie potężnych jednolinijkowców
- pozwala oszczędzić czas wprawnemu programiście

NOKIA

Czemu Bash? (zalety cd.)

- proste zadania są krótkie i proste
- powszechność na platformie linuxowej
- duża wsteczna kompatybilność zapewniająca przenośność pomiędzy systemami i ich różnymi wersjami
- wciąż duża popularność, pomimo obecności nowych rozwiązań

NOKIA

Kiedy Bash się nie sprawdzi? (wady)

- realizacja bardziej złożonych zadań (xml, json, ...)
- nieumiejętność pisania czytelnego kodu (trudna analiza, odpluskwianie)

NOKIA

Agenda

NOKIA

Najczęstsze operacje i narzędzia

zmienne środowiskowe

- $HOME, $SHELL, $PATH, $RANDOM, $PWD, $OLDPWD, $PS1

polecenia powłoki

- cd, pwd, cat, mkdir, touch, ls, rm, mv, ps, top, history, source, export, tail, head, grep, sed, fg

skróty klawiszowe używane w konsoli

- ctrl+c, ctrl+z, ctrl+l, ctrl+d, esc+.

istotne pliki czytane przez powłokę Bash

- .bashrc, .bash_history, .bash_profile, .bash_logout

NOKIA

Przypisanie zmiennej i operacje arytmetyczne

```
1 var=255                # komentarz: Przypisz liczbę 255 do var.
    Ważne: bez znaku spacje przed i po znaku '='
2 echo $var              # Wypisz wartosc var na ekran
3 ((var = var + 255))    # Dodaj 1
4 echo $((var + 1))      # Wypisz wartosc var+1 na ekran
5 ((++var))              # inkrementacja jak w języku C
6 ((var++))              # postinkrementacja
7 echo $[var=var*100]    # składnia z możliwością przypisania i
    wypisanie wyniku na ekran
8 echo $((var * 20))     # To samo, bez zapisania do zmiennej
9 echo $((var>100))      # Sprawdzenie czy var jest wieksze od 100.
    Zero oznacza prawdę. Jeden - fałsz.
```

NOKIA

Agenda

NOKIA

shebang

Plik skrypt.sh:

```
1    #!/usr/bin/env bash
```

Linia poleceń:

```
$ echo '#!/bin/bash' > skrypt.sh
$ ls -l skrypt.sh
-rw-rw-r-- 1 user user 12 Oct 28 15:41 skrypt.sh
$ chmod +x skrypt.sh
$ ls -l skrypt.sh
-rwxrwxr-x 1 user user 12 Oct 28 15:41 skrypt.sh*
$ ./skrypt.sh # uruchamianie
```

NOKIA

wyrażenie warunkowe

```
1 if warunek
2 then
3   polecenie
4 fi
```

NOKIA

wyrażenie warunkowe

```
1 if warunek
2 then
3   polecenie1
4 else
5   polecenie2
6 fi
```

NOKIA

wyrażenie warunkowe

```
1 if warunek
2 then
3   polecenie1
4 elif warunek
5 then
6   polecenie2
7 fi
```

wyrażenie warunkowe

```
1 #!/bin/bash
2 if [ -e ~/.bashrc ]
3 then
4   echo "Istnieje plik .bashrc"
5 else
6   echo "Nie istnieje plik .bashrc"
7 fi
```

NOKIA

wyrażenie warunkowe - operatory test

- -f plik istnieje i jest zwykłym plikiem
- -b plik istnieje i jest blokowym plikiem specjalnym
- - plik istnieje i jest plikiem znakowym
- -e plik istnieje
- -h plik istnieje i jest linkiem symbolicznym
- = sprawdza czy wyrażenia są równe
- != sprawdza czy wyrażenia są różne
- -n wyrażenie ma długość większą niż 0
- -d wyrażenie istnieje i jest katalogiem
- -z wyrażenie ma zerową długość
- -r można czytać plik
- ...

Więcej informacji: `$ man test`

wyrażenie warunkowe - operatory test

- -f plik istnieje i jest zwykłym plikiem
- -b plik istnieje i jest blokowym plikiem specjalnym
- - plik istnieje i jest plikiem znakowym
- -e plik istnieje
- -h plik istnieje i jest linkiem symbolicznym
- = sprawdza czy wyrażenia są równe
- != sprawdza czy wyrażenia są różne
- -n wyrażenie ma długość większą niż 0
- -d wyrażenie istnieje i jest katalogiem
- -z wyrażenie ma zerową długość
- -r można czytać plik
- ...

Więcej informacji: `$ man test`

wyrażenie warunkowe - operatory test

- -f plik istnieje i jest zwykłym plikiem
- -b plik istnieje i jest blokowym plikiem specjalnym
- - plik istnieje i jest plikiem znakowym
- -e plik istnieje
- -h plik istnieje i jest linkiem symbolicznym
- = sprawdza czy wyrażenia są równe
- != sprawdza czy wyrażenia są różne
- -n wyrażenie ma długość większą niż 0
- -d wyrażenie istnieje i jest katalogiem
- -z wyrażenie ma zerową długość
- -r można czytać plik
- ...

Więcej informacji: `$ man test`

wyrażenie warunkowe - operatory test

- -f plik istnieje i jest zwykłym plikiem
- -b plik istnieje i jest blokowym plikiem specjalnym
- - plik istnieje i jest plikiem znakowym
- -e plik istnieje
- -h plik istnieje i jest linkiem symbolicznym
- = sprawdza czy wyrażenia są równe
- != sprawdza czy wyrażenia są różne
- -n wyrażenie ma długość większą niż 0
- -d wyrażenie istnieje i jest katalogiem
- -z wyrażenie ma zerową długość
- -r można czytać plik
- ...

Więcej informacji: `$ man test`

wyrażenie warunkowe - operatory test

- -f plik istnieje i jest zwykłym plikiem
- -b plik istnieje i jest blokowym plikiem specjalnym
- - plik istnieje i jest plikiem znakowym
- -e plik istnieje
- -h plik istnieje i jest linkiem symbolicznym
- = sprawdza czy wyrażenia są równe
- != sprawdza czy wyrażenia są różne
- -n wyrażenie ma długość większą niż 0
- -d wyrażenie istnieje i jest katalogiem
- -z wyrażenie ma zerową długość
- -r można czytać plik
- ...

Więcej informacji: `$ man test`

wyrażenie warunkowe - operatory test

- -f plik istnieje i jest zwykłym plikiem
- -b plik istnieje i jest blokowym plikiem specjalnym
- - plik istnieje i jest plikiem znakowym
- -e plik istnieje
- -h plik istnieje i jest linkiem symbolicznym
- = sprawdza czy wyrażenia są równe
- != sprawdza czy wyrażenia są różne
- -n wyrażenie ma długość większą niż 0
- -d wyrażenie istnieje i jest katalogiem
- -z wyrażenie ma zerową długość
- -r można czytać plik
- ...

Więcej informacji: `$ man test`

wyrażenie warunkowe - operatory test

- -f plik istnieje i jest zwykłym plikiem
- -b plik istnieje i jest blokowym plikiem specjalnym
- - plik istnieje i jest plikiem znakowym
- -e plik istnieje
- -h plik istnieje i jest linkiem symbolicznym
- = sprawdza czy wyrażenia są równe
- != sprawdza czy wyrażenia są różne
- -n wyrażenie ma długość większą niż 0
- -d wyrażenie istnieje i jest katalogiem
- -z wyrażenie ma zerową długość
- -r można czytać plik
- ...

Więcej informacji: `$ man test`

wyrażenie warunkowe - operatory test

- -f plik istnieje i jest zwykłym plikiem
- -b plik istnieje i jest blokowym plikiem specjalnym
- - plik istnieje i jest plikiem znakowym
- -e plik istnieje
- -h plik istnieje i jest linkiem symbolicznym
- = sprawdza czy wyrażenia są równe
- != sprawdza czy wyrażenia są różne
- -n wyrażenie ma długość większą niż 0
- -d wyrażenie istnieje i jest katalogiem
- -z wyrażenie ma zerową długość
- -r można czytać plik
- ...

Więcej informacji: `$ man test`

wyrażenie warunkowe - operatory test

- -f plik istnieje i jest zwykłym plikiem
- -b plik istnieje i jest blokowym plikiem specjalnym
- - plik istnieje i jest plikiem znakowym
- -e plik istnieje
- -h plik istnieje i jest linkiem symbolicznym
- = sprawdza czy wyrażenia są równe
- != sprawdza czy wyrażenia są różne
- -n wyrażenie ma długość większą niż 0
- -d wyrażenie istnieje i jest katalogiem
- -z wyrażenie ma zerową długość
- -r można czytać plik
- ...

Więcej informacji: `$ man test`

wyrażenie warunkowe - operatory test

- -f plik istnieje i jest zwykłym plikiem
- -b plik istnieje i jest blokowym plikiem specjalnym
- - plik istnieje i jest plikiem znakowym
- -e plik istnieje
- -h plik istnieje i jest linkiem symbolicznym
- = sprawdza czy wyrażenia są równe
- != sprawdza czy wyrażenia są różne
- -n wyrażenie ma długość większą niż 0
- -d wyrażenie istnieje i jest katalogiem
- -z wyrażenie ma zerową długość
- -r można czytać plik
- ...

Więcej informacji: `$ man test`

wyrażenie warunkowe - operatory test

- -f plik istnieje i jest zwykłym plikiem
- -b plik istnieje i jest blokowym plikiem specjalnym
- - plik istnieje i jest plikiem znakowym
- -e plik istnieje
- -h plik istnieje i jest linkiem symbolicznym
- = sprawdza czy wyrażenia są równe
- != sprawdza czy wyrażenia są różne
- -n wyrażenie ma długość większą niż 0
- -d wyrażenie istnieje i jest katalogiem
- -z wyrażenie ma zerową długość
- -r można czytać plik
- ...

Więcej informacji: `$ man test`

wyrażenie warunkowe - operatory test

- -f plik istnieje i jest zwykłym plikiem
- -b plik istnieje i jest blokowym plikiem specjalnym
- - plik istnieje i jest plikiem znakowym
- -e plik istnieje
- -h plik istnieje i jest linkiem symbolicznym
- = sprawdza czy wyrażenia są równe
- != sprawdza czy wyrażenia są różne
- -n wyrażenie ma długość większą niż 0
- -d wyrażenie istnieje i jest katalogiem
- -z wyrażenie ma zerową długość
- -r można czytać plik
- ...

Więcej informacji: `$ man test`

Agenda

NOKIA

Pętla for

```
1 for i in lista; do
2     polecenie
3     if warunek; then
4         continue
5     else
6         break
7     fi
8 done
```

Generowanie sekwencji: {0..10..2}

NOKIA

Pętla while

```
while warunek; do
    polecenie
    if warunek; then
        continue
    else
        break
    fi
done
```

NOKIA

Pętla until

```
until warunek; do
    polecenie
    if warunek; then
        continue
    else
        break
    fi
done
```

NOKIA

Przerywanie i kontynuacja

- break [N] - przerwanie N (domyślnie 1) pętli
- continue [N] - przejście do następnej iteracji N-tej pętli

NOKIA

Przerywanie i kontynuacja

- break [N] - przerwanie N (domyślnie 1) pętli
- continue [N] - przejście do następnej iteracji N-tej pętli

NOKIA

Zasięg zmiennych

- export - zmienna zostanie przekazana do potomków,
- local - zmienna będzie widoczna/zmieniana tylko w obrębie funkcji

NOKIA

Zasięg zmiennych

- export - zmienna zostanie przekazana do potomków,
- local - zmienna będzie widoczna/zmieniana tylko w obrębie funkcji

NOKIA

Zmienne specjalne

- `$0` - nazwa skryptu/programu
- `$1 $2 $3` - odpowiednio pierwszy, drugi, trzeci argument
- `$#` - liczba argumentów
- `$@` - wszystkie argumenty
- `$*` - wszystkie argumenty (inne zachowanie)
- `$?` - wynik wykonania ostatniej komendy
- `$$` - `PID` procesu aktualnie wykonywanego shella

Więcej informacji: `$ man bash`, sekcja **Special Parameters**

NOKIA

Zmienne specjalne

- `$0` - nazwa skryptu/programu
- `$1 $2 $3` - odpowiednio pierwszy, drugi, trzeci argument
- `$#` - liczba argumentów
- `$@` - wszystkie argumenty
- `$*` - wszystkie argumenty (inne zachowanie)
- `$?` - wynik wykonania ostatniej komendy
- `$$` - `PID` procesu aktualnie wykonywanego shella

Więcej informacji: `$ man bash`, sekcja **Special Parameters**

NOKIA

Zmienne specjalne

- `$0` - nazwa skryptu/programu
- `$1 $2 $3` - odpowiednio pierwszy, drugi, trzeci argument
- `$#` - liczba argumentów
- `$@` - wszystkie argumenty
- `$*` - wszystkie argumenty (inne zachowanie)
- `$?` - wynik wykonania ostatniej komendy
- `$$` - `PID` procesu aktualnie wykonywanego shella

Więcej informacji: `$ man bash`, sekcja **Special Parameters**

NOKIA

Zmienne specjalne

- `$0` - nazwa skryptu/programu
- `$1 $2 $3` - odpowiednio pierwszy, drugi, trzeci argument
- `$#` - liczba argumentów
- `$@` - wszystkie argumenty
- `$*` - wszystkie argumenty (inne zachowanie)
- `$?` - wynik wykonania ostatniej komendy
- `$$` - `PID` procesu aktualnie wykonywanego shella

Więcej informacji: `$ man bash`, sekcja **Special Parameters**

NOKIA

Zmienne specjalne

- `$0` - nazwa skryptu/programu
- `$1 $2 $3` - odpowiednio pierwszy, drugi, trzeci argument
- `$#` - liczba argumentów
- `$@` - wszystkie argumenty
- `$*` - wszystkie argumenty (inne zachowanie)
- `$?` - wynik wykonania ostatniej komendy
- `$$` - `PID` procesu aktualnie wykonywanego shella

Więcej informacji: `$ man bash`, sekcja **Special Parameters**

NOKIA

Zmienne specjalne

- `$0` - nazwa skryptu/programu
- `$1 $2 $3` - odpowiednio pierwszy, drugi, trzeci argument
- `$#` - liczba argumentów
- `$@` - wszystkie argumenty
- `$*` - wszystkie argumenty (inne zachowanie)
- `$?` - wynik wykonania ostatniej komendy
- `$$` - `PID` procesu aktualnie wykonywanego shella

Więcej informacji: `$ man bash`, sekcja **Special Parameters**

NOKIA

Zmienne specjalne

- `$0` - nazwa skryptu/programu
- `$1 $2 $3` - odpowiednio pierwszy, drugi, trzeci argument
- `$#` - liczba argumentów
- `$@` - wszystkie argumenty
- `$*` - wszystkie argumenty (inne zachowanie)
- `$?` - wynik wykonania ostatniej komendy
- `$$` - `PID` procesu aktualnie wykonywanego shella

Więcej informacji: `$ man bash`, sekcja **Special Parameters**

NOKIA

funkcje

```
1 #!/bin/bash
2
3 myfunc() {
4   do_something
5 }
6
7 function myfunc() { # nie polecane
8   do_something
9 }
```

NOKIA

funkcje

```
1 #!/bin/bash
2 echo "W glownym programie"
3
4 echo "\$*: $*"
5 echo "\$@: $@"
6 echo "\$1: $1"
7
8 func() {
9   echo "W ciele funkcji:"
10  echo "\$*: $*"
11  echo "\$@: $@"
12  echo "\$1: $1"
13 }
14
15 func "$@"
16 func "$*"
```

```
W glownym programie
$*: 1 2 3
$@: 1 2 3
$1: 1
W ciele funkcji:
$*: 1 2 3
$@: 1 2 3
$1: 1
W ciele funkcji:
$*: 1 2 3
$@: 1 2 3
$1: 1 2 3
```

NOKIA

wielowybór `case`

```
case "$1" in
    build)
        show_config
        build_image
    ;;
    bundle)
        show_config
        bundle_image
    ;;
    upload)
        show_config
        upload_image
    ;;
esac
```

NOKIA

select

```
1PS3='terminal? '   # by default PS3="$@"
2select term in gl35a t2000 s531 vt99; do
3    if [[ -n $term ]]; then
4        TERM=$term
5        print TERM is $TERM
6        break
7    else
8        print 'invalid.'
9    fi
10done
```

Output:

```
1) gl35a
2) t2000
3) s531
4) vt99
terminal?
```

NOKIA

Tematy na drugą część wykładu

- wyrażenia regularne =~, `$BASH_REMATCH`
- tablice (arrays)
- podstawienia ciagów (substring substitution)
- programy wykorzystywane w skryptach `sed`, `awk`, `grep`, `find`

NOKIA

Tematy na drugą część wykładu

- wyrażenia regularne =~, `$BASH_REMATCH`
- tablice (arrays)
- podstawienia ciagów (substring substitution)
- programy wykorzystywane w skryptach `sed`, `awk`, `grep`, `find`

NOKIA

Tematy na drugą część wykładu

- wyrażenia regularne =~, `$BASH_REMATCH`
- tablice (arrays)
- podstawienia ciagów (substring substitution)
- programy wykorzystywane w skryptach `sed`, `awk`, `grep`, `find`

NOKIA

Tematy na drugą część wykładu

- wyrażenia regularne `=~`, `$BASH_REMATCH`
- tablice (arrays)
- podstawienia ciagów (substring substitution)
- programy wykorzystywane w skryptach `sed`, `awk`, `grep`, `find`

NOKIA

Literatura

- http://tldp.org/LDP/Bash-Beginners-Guide/html/
- http://tldp.org/LDP/abs/html/
- http://wiki.bash-hackers.org/doku.php

NOKIA