

BASH ♡

Adrianna Zelek

19 stycznia 2016

Spis treści

1	Wiadomości podstawowe	3
1.1	Co to jest skrypt?	3
1.2	Hello world	3
2	Zmienne	4
2.1	Zmienne programowe	4
2.2	Zmienne systemowe	4
2.3	Zmienne tablicowe	4
2.4	Zmienne specjalne	5
3	Operacje Arytmetyczne	5
3.1	Dodawanie i odejmowanie	5
3.2	Potęgowanie i pierwiastkowanie	6
4	Dostęp do plików	6
5	Trudne wzory	6

1 Wiadomości podstawowe

BASH - powłoka systemowa UNIX napisana dla projektu GNU. Program jest rozprowadzany na licencji GPL.

Bash to jedna z najpopularniejszych powłok systemów uniksowych. Jest domyślną powłoką w większości dystrybucji systemu GNU/Linux oraz w systemie OS X od wersji 10.3, istnieją także wersje dla większości systemów uniksowych. Bash jest także domyślną powłoką w środowisku Cygwin dla systemów Win32.

Bash pozwala na pracę w trybie konwersacyjnym i w trybie wsadowym. Język basha umożliwia definiowanie aliasów, funkcji, zawiera konstrukcje sterujące przepływem (if, while, for, ...). Powłoka systemowa zachowuje historię wykonywanych poleceń i zapisuje ją domyślnie w pliku .bash-history w katalogu domowym użytkownika.

Nazwa jest akronimem od Bourne-Again Shell (angielska gra słów: fonetycznie brzmi tak samo, jak born again shell, czyli odrodzona powłoka). Wywodzi się od powłoki Bourne'a sh, która była jedną z pierwszych i najważniejszych powłok systemu UNIX oraz zawiera pomysły zawarte w powłokach Korn i csh. Bash był pisany głównie przez Briana Foksa i Cheta Rameya w 1987. [1]

1.1 Co to jest skrypt?

Skrypt to niekompilowany tekstowy plik wykonywalny, zawierający jakieś polecenia systemowe oraz polecenia sterujące jego wykonaniem (instrukcje, pętle itp.). Skrypt wykonywany jest tylko i wyłącznie przez *interpreter* (tutaj /bin/bash), który tłumaczy polecenia zawarte w skrypcie na język zrozumiały dla procesora.

1.2 Hello world

Nasz pierwszy skrypt:

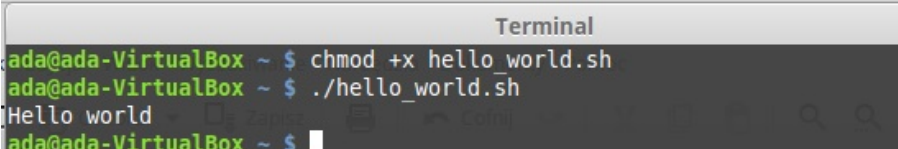
```
#!/bin/bash
```

```
# Tu jest komentarz.
```

```
echo "Hello world"
```

- oznacza komentarz, wszystko co znajduje się za nim w tej samej linii, jest pomijane przez interpreter.

Pierwsza linia skryptu zaczynająca się od znaków: # ! ma szczególne znaczenie - wskazuje na rodzaj shella w jakim skrypt ma być wykonany, tutaj skrypt zawsze będzie wykonywany przez interpreter poleceń /bin/bash, niezależnie od tego jakiego rodzaju powłoki w danej chwili używamy. 1



```
Terminal
ada@ada-VirtualBox ~ $ chmod +x hello_world.sh
ada@ada-VirtualBox ~ $ ./hello_world.sh
Hello world
ada@ada-VirtualBox ~ $
```

Rysunek 1: Wykonanie skryptu hello_world.sh

2 Zmienne

2.1 Zmienne programowe

Zmienne programowe (użytkownika) (ang. program variables) - są to zmienne po prostu wykorzystywane wewnątrz skryptu, które definiuje użytkownik. Nie są dostępne dla innych programów, skryptów i powłok potomnych. Ich nazwy nie mogą się zaczynać od cyfry i pisane są zwyczajowo małymi literami. Czasami stosuje się podkreślenia.

Zmienne tego typu inicjujemy w następujący sposób:

```
nazwa_zmiennej="wartość"
```

a korzystamy z nich, poprzedzając ich wartość znakiem dolara (\$).

```
echo $nazwa_zmiennej
```

2.2 Zmienne systemowe

Zmienne systemowe, często nazwane także zmiennymi środowiskowymi (ang. environment variables)- są zdefiniowane i przechowywane w środowisku nadrzędnym (powłoce z której uruchamiane są Twoje skrypty). Ich nazwy są zwyczajowo tworzone z dużych liter i mogą być testowane poleceniem set. Definiują środowisko użytkownika, dostępne dla wszystkich procesów potomnych. [2]

Przykłady:

- \$HOME ścieżka do twojego katalogu domowego
- \$USER twój login
- \$HOSTNAME nazwa twojego hosta
- \$OSTYPE rodzaj systemu operacyjnego
- \$UID Unique User ID - unikalny numer użytkownika
- \$TERM Typ terminala z jakiego korzystamy ...

2.3 Zmienne tablicowe

BASH pozwala na stosowanie zmiennych tablicowych jednowymiarowych. Tablica to zmienna która przechowuje listę jakichś wartości (rozdzielonych spacjami). W BASH'u nie ma maksymalnego rozmiaru tablic. Kolejne wartości zmiennej tablicowej indeksowane są przy pomocy liczb całkowitych, zaczynając od 0. [3]

Składnia

```
zmienna=(wartość1 wartość2 wartość3 wartośn)
```

Przykład:

```
#!/bin/bash
```

```
tablica=(element1 element2 element3)
```

```
echo $tablica[0]
```

```
echo $tablica[1]
```

```
echo $tablica[2]
```

2.4 Zmienne specjalne

\$0 nazwa bieżącego skryptu lub powłoki

Przykład:

```
#!/bin/bash
```

```
echo "$0"
```

Pokaże nazwę naszego skryptu.

\$1..\$9 Parametry przekazywane do skryptu (wyjątek, użytkownik może modyfikować ten rodzaj \$-ych specjalnych.2

```
#!/bin/bash echo "$1 $2 $3"
```

Jeśli wywołamy skrypt z parametrami to przypisane zostaną zmiennym: od \$1 do \$9.

\$@

Pokaże wszystkie parametry przekazywane do skryptu (też wyjątek), równoważne \$1 \$2 \$3..., jeśli nie podane są żadne parametry \$@ interpretowana jest jako nic.

Przykład:

```
#!/bin/bash
```

```
echo "Skrypt uruchomiono z parametrami: $@"
```

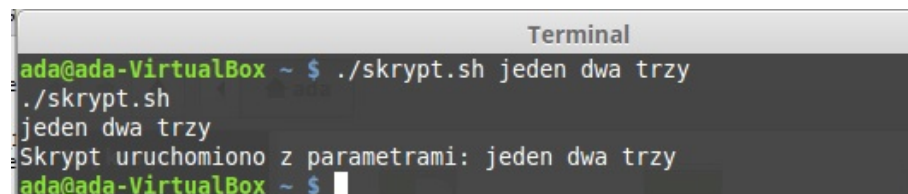
skrypt.sh obrazek2

```
#!/bin/bash
```

```
echo "$0"
```

```
echo "$1 $2 $3"
```

```
echo "Skrypt uruchomiono z parametrami: $@"
```



Rysunek 2: Wykonanie skryptu skrypt.sh

3 Operacje Arytmetyczne

3.1 Dodawanie i odejmowanie

1. Polecenie "let"

```
let result=123+4
```

```
echo $result
```

2. Operator []

```
result=$(( 5 + 5 )
```

```
echo $resultt
```

3. Polecenie "expr"

```
result=`expr 5 + 5 `
```

```
echo $result
```

3.2 Potęgowanie i pierwiastkowanie

Pierwiastkowanie \sqrt{x}

echo "sqrt(100)" | bc

Potęgowanie a^n

echo "sqrt(2 ^12)" | bc

4 Dostęp do plików

chmod (ang. change mode – zmiana atrybutu) – polecenie zmiany zezwoleń dostępu do plików w systemach unixowych. [4] 1

Cyfra	Prawa	Litera
0	Brak praw	
1	Wykonywanie	x
2	Pisanie	w
4	Czytanie	r

Tabela 1: Tabela z interpretacją kodów ósemkowych

Przykłady użycia:

- \$ chmod a+w plik.txt — nadaje wszystkim uprawnienia do zmiany 'plik.txt',
- \$ chmod o-x plik.txt — usuwa możliwość wykonywania 'plik.txt' przez pozostałych użytkowników,
- \$ chmod go=rx plik.txt — grupa oraz pozostali użytkownicy będą mogli tylko czytać i wykonywać.
- \$ chmod -R 777 /home/user — wszyscy będą mogli zmieniać zawartość katalogu /home/user oraz jego podkatalogów, jak też czytać go i wykonywać zawarte w nim pliki

5 Trudne wzory

Całka 5

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$$

$$\sum_{i=1}^n \int_0^{\frac{\pi}{2}}$$

$$y = \begin{cases} a & \text{gdy } d > c \\ b + x & \text{gdy } d = c \\ l & \text{gdy } d < c \end{cases}$$

Literatura

- [1] Wikipedia bash
- [2] www.bash.0x1fff.com
- [3] prezentacja - Środowisko programisty - BASH Emilia Lubecka
- [4] Wikipedia Chmod