

Narzędzia w branży IT

Laboratorium 2

„Podstawy pisania algorytmów w języku BASH”

1. Cele ćwiczenia:

- Zapoznanie się ze składnią i podstawowymi wyrażeniami w języku bash
- Napisanie prostych algorytmów w języku bash

2. Baza wiedzy:

- udostępnione wykłady

3. Wstęp teoretyczny

a) Zmienne:

Wiek=22 # zmienna **Wiek** przechowuje liczbę 22, brak spacji przed i po '='

powitanie="Dzień dobry" # zmienna **powitanie** przechowuje ciąg znaków ze spacjami

imie=Artur # zmienna **imie** przechowuje ciąg znaków bez spacji

echo Wiek # wyświetli napis „Wiek”

echo \$Wiek # wyświetli wartość zmiennej **Wiek**

echo "\$powitanie \$imie. Ilość Twoich lat to: \$Wiek" # w cudzysłowie można łączyć ciągi znaków i wartości zmiennych

echo '\$Wiek' # wyświetli dosłownie ciąg znaków znajdujący się w “”, czyli \$Wiek

adresUrl=\$(pwd) # wynik komendy **pwd** zapisywany jest do zmiennej **adresUrl**

echo \$adresUrl # wyświetlenie wyniku komendy **pwd** przez zmienną

echo \$(pwd) # bezpośrednie wyświetlenie wyniku komendy **pwd**

b) Zmienne specjalne:

\$0 # nazwa skryptu

\$1 # pierwszy argument

\$2 # drugi argument

\$@ # wszystkie argumenty

\$? # zwrócona wartość przez program echo

\$\$ # PID bieżącej powłoki

\$# # Ilość argumentów

c) Tablice:

```
tablica=(1 3 9 "Ciąg znaków" Ciąg znaków) # spacja oddziela kolejne elementy tablic
echo ${tablica[0]} # tablice w bash indeksowane są od 0
echo ${tablica[1]} # wyświetlenie elementu tablicy o indeksie 1
tablica[0]=99 # przypisanie wartości do elementu tablicy o indeksie 0
echo ${tablica[*]} # drukuje wszystkie elementy
echo ${tablica[@]} # to samo, drukuje wszystkie elementy
echo ${#tablica[3]} # wyświetli ilość znaków elementu pod indeksem 3
echo ${#tablica[@]} # wyświetli ilość elementów tablicy
```

d) Działania matematyczne:

```
a=$(( a+1 ))
a=${a+1}
let a=a+1
let a+=1
```

+ Dodawanie
 - Odejmowanie
 * Mnożenie
 / Dzielenie
 ** Potęgowanie (operand z prawej strony jest wykładnikiem)
 % Reszta z dzielenia

e) Instrukcje warunkowe (if):

-lt Mniejszy niż	-le Mniejszy, lub równy
-gt Większy niż	-ge Większy lub równy
= Równy	!= Różny
-a Plik istnieje	-b Plik istnieje i jest blokowym plikiem specjalnym
- Plik istnieje i jest plikiem znakowym	-e Plik istnieje
-h Plik istnieje i jest linkiem symbolicznym	-n Wyrażenie ma długość większą niż 0
-d Wyrażenie istnieje i jest katalogiem	-z Wyrażenie ma zerową długość
-r Można czytać plik	-w Można zapisywać do pliku
-x Można plik wykonać	-f Plik istnieje i jest plikiem zwykłym
-p Plik jest łączem nazwanym ostatniego użycia	-N Plik istnieje i był zmieniany od czasu jego

```
P1 -nt P2 # P1 jest nowszy od P2
```

```
P1 -ot P2 # P1 jest starszy od P2
```

```
ilosc_arg="$#"
```

```
if [spacja$ilosc_argspacja=spacja"0"spacja]
then
    # kod
elif [ -e "$1" ]
then
    # kod
else
    # kod
fi
```

f) Wczytywanie danych:

read tekst # zapisanie danych z klawiatury do zmiennej tekst

read -p "Wpisz tekst: " tekst # zapisanie danych z klawiatury do zmiennej tekst

g) Instrukcja case:

read liczba

case "\$liczba" **in**

 "1") *echo* "Jeden" ;;

 "2") *echo* "Dwa" ;;

 *) *echo* "Inna liczba"

esac

h) Pętle:

i=0

while [*\$i -lt „10”*]; **do**

echo -n "\$i "

i=\$((i+1))

done

#

i=1

while ((*i <= 10*)); **do**

echo -n "\$i "

let "i++"

done

#

for *i in 1 2 3 4 5 6 7 8 9 10*; **do**

echo -n "\$i "

done

#

for *i in `seq 1 10`*; **do**

echo -n "\$i "

done

#

for ((*i = 1; i <= 10; i++*)); **do**

echo -n "\$i "

done

4:

Zadania do wykonania:

Nazwa „**daneUsera.txt**” oznacza dowolną nazwę pliku wyjściowego, wybraną przez Państwa.

Zad1. utworzyć skrypt wykonujący następujące czynności:

- skrypt przyjmuje argumenty:
./skrypt.sh **imię nazwisko rok_urodzenia daneUsera.txt**
- jeżeli nie podano którejś informacji, skrypt powinien zapytać o nią użytkownika
- skrypt powinien powitać użytkownika jego imieniem i nazwiskiem i wypisać ile lat ma użytkownik.
- napisać funkcję „silnia”, dwie wersje: rekurencyjnie i przez pętle (dowolną)
- skrypt powinien **dopisać** uzyskane informacje do pliku tekstowego (nazwa pliku dowolna, np.: daneUsera.txt) w formacie:
imię, nazwisko, rok urodzenia, aktualny_wiek, wyniki_funkcji_silnia(wiek)

Zad 2. Utworzyć skrypt wykonujący następujące czynności:

- skrypt uruchamiany argumentami ./skrypt.sh **daneUsera.txt**, jeżeli nie podano nazwy pliku, skrypt powinien zapytać o nią użytkownika
- skrypt sprawdza, czy Zad1 istnieje, jeżeli nie, to zgłasza błąd.
- skrypt sprawdza, czy ścieżka do jego skryptu znajduje się w zmiennej systemowej PATH, jeśli nie, to ją dodaje, **należy być ostrożnym aby nie nadpisać/skasować całej zawartości \$PATH**
- skrypt tworzy nowy katalog i przenosi do niego plik: daneUsera/daneUsera.txt, jeżeli taki katalog istniał wcześniej, usuwa go, lub nadpisuje.
- skrypt wyświetla zawartość obecnego folderu i dopisuje wynik do pliku daneUsera/daneUsera.txt

Zad 3. utworzyć skrypt **Kalendarz2022** wykonujący następujące czynności:

- skrypt uruchamiany argumentami ./skrypt.sh **numerDnia miesiąc**
- skrypt dopisuje na ekranie i do pliku1 jaki to dzień tygodnia
- jeżeli użytkownik poda błędne dane, lub wyznaczenie dnia tygodnia się nie powiedzie, informację o tym jako błąd skrypt dopisze do pliku2

Jako podsumowanie ćwiczenia należy przesłać jedynie 3 pliki zawierające skrypty wykonujące poszczególne ćwiczenia.

Pliki należy spakować w dowolnym formacie, np.: zip, rar.