

Klasa: II TI Technikum Kształtowania Środowiska - Technik Informatyk

SYSTEMY OPERACYJNE

Temat: Skrypty powłoki.

Wykonałam zrzuty z Podręcznika: WSiP „Przygotowanie stanowiska komputerowego. Część2” T.Marciniuk, K.Pytel, S.Osetek. Proszę zapoznać się z treścią tego podręcznika.

Przeczytaj rozdział dotyczący dzisiejszej lekcji.

18.2. Skrypty powłoki

Skrypty są zwykłymi plikami tekstowymi, w których są zapisane polecenia zrozumiałe dla powłoki. Zadaniem powłoki jest przetłumaczenie ich na polecenia systemu. Aby przygotować skrypt, należy:

- stworzyć plik, w którym trzeba umieścić kod, np.:

```
touch naszskrypt.bat
```

- za pomocą dowolnego edytora tekstu wpisać do pliku kolejno wykonywane polecenia;
- nadać uprawnienia do wykonywania pliku dla uprawnionych użytkowników;
- uruchomić skrypt:

```
./naszskrypt.bat
```

gdzie symbol ./ oznacza, że skrypt znajduje się w bieżącym katalogu.

Przykładowy, bardzo prosty skrypt może się składać z następujących poleceń:

```
#!/bin/bash
#Tu jest komentarz.
echo „Hello World”
```

Pierwsza linia skryptu zaczynająca się od znaków **#!** ma szczególne znaczenie – wskazuje na rodzaj powłoki, w której skrypt ma być wykonany. Tutaj skrypt zawsze będzie wykonywany przez interpreter poleceń **/bin/bash**, niezależnie od tego, jakiego rodzaju powłoki używamy w danej chwili.

Znak **#** (hasz) oznacza komentarz – wszystko, co znajduje się za nim w tej samej linii, jest pomijane przez interpreter.

Trzecia linia spowoduje wydrukowanie na standardowym wyjściu (*stdout*), czyli na ekranie, napisu: „Hello World”.

Między nawiasami a treścią warunku muszą być spacje, tak jak powyżej.
Polecenie **test** zwraca wartość **0** (*true*), jeśli warunek jest spełniony, i wartość **1** (*false*), jeśli warunek nie jest spełniony, np. **test -e plik**

Wybrane przykłady operatorów polecenia **test**:

```
-e plik istnieje;  
= sprawdza, czy wyrażenia są równe;  
!= sprawdza, czy wyrażenia są różne;  
-d wyrażenie istnieje i jest katalogiem;  
-r można czytać plik;  
-w można zapisywać do pliku;  
-x można wykonać plik;  
-lt mniejsze niż;  
-gt większe niż;  
-ge większe lub równe;  
-le mniejsze lub równe.
```

instrukcja **case** – pozwala na dokonanie wyboru spośród kilku wzorców.

Sprawdzana jest wartość zmiennej po słowie kluczowym **case** i porównywana ze wszystkimi wariantami po kolei. Jeśli dopasowanie zakończy się sukcesem, wykonane zostanie polecenie lub polecenia przypisane do danego wzorca. W przeciwnym wypadku zostanie użyte polecenie domyślne oznaczone gwiazdką „*”. Składnia polecenia:

```
case zmienna in  
„wzorzec1”) polecenie1 ;;  
„wzorzec2”) polecenie2 ;;  
„wzorzec3”) polecenie3 ;;  
*) polecenie_domyślne  
esac
```

PRZYKŁAD 18.1

```
#!/bin/bash  
echo „Podaj cyfrę dnia tygodnia”  
read d  
case „$d” in  
„1”) echo „Poniedziałek” ;;  
„2”) echo „Wtorek” ;;  
„3”) echo „Środa” ;;  
„4”) echo „Czwartek” ;;  
„5”) echo „Piątek” ;;  
„6”) echo „Sobota” ;;  
„7”) echo „Niedziela” ;;  
*) echo „Nic nie wybrałeś”  
esac
```

- pętla **for** – wykonuje polecenia zawarte wewnątrz pętli, na każdym składniku listy.
Składnia polecenia:

```
for zmienna in lista  
do  
polecenie  
done
```

Pozdrawiam 2TI

Bogusława Kocałek