

**POLITECHNIKA KRAKOWSKA**  
**IM. TADEUSZA KOŚCIUSZKI**

**Wydział Inżynierii Elektrycznej i Komputerowej**



**Politechnika Krakowska**  
**im. Tadeusza Kościuszki**

**Systemy Operacyjne**

**Sprawozdanie z laboratorium**

*Laboratorium nr 5*

<i>Autor:</i>	Jarosław Sipika
<i>Kierunek:</i>	Informatyka
<i>Semestr:</i>	III
<i>Grupa:</i>	L2
<i>Data:</i>	23.12.2018

## Spis treści

1. Cel i zakres sprawozdania .....	3
2. Laboratorium nr 5.....	3
2.1 Wstęp .....	3
2.2 Opis zadań .....	4
2.2.1 Zadanie 01 .....	4
2.2.2 Zadanie 02 .....	4
2.2.3 Zadanie 03 .....	5
2.2.4 Zadanie 04 .....	5
2.2.5 Zadanie 05 .....	6
3. Wnioski.....	6

## 1. Cel i zakres sprawozdania

Celem sprawozdania jest omówienie zadań oraz informacji które zostały przedstawione podczas zajęć z przedmiotu „Systemy operacyjne”. Sprawozdanie swym zakresem obejmować będzie zadania, które zostały wykonane podczas laboratorium nr 5

Omówione zostaną kolejne ćwiczenia wraz z wnioskami, które można było dostrzec po wykonaniu odpowiedniego zadania.

Laboratorium nr 5 miało na celu zapoznanie się z używaniem funkcji oraz ich tworzeniem

Na końcu sprawozdania znajdują się załączniki do wykonanych zadań w postaci wydruków z notatnika oraz zrzuty ekranu przedstawiające pracę programu.

## 2. Laboratorium nr 5

### 2.1 Wstęp

Laboratorium miało na celu zapoznanie się z tworzeniem funkcji.

**Funkcje** – w każdym języku programowania lub języku skryptowym można organizować sobie kod za pomocą funkcji. Funkcje są to tzw. podprogramy, które rozwiązują jakieś problemy.

#### Przykład

```
Function nazwa_funkcji {  
    Polecenie  
    Polecenie  
    Polecenie  
}
```

```
Function hello {  
    echo „Hello world!”  
}
```

#### Wywołanie funkcji:

```
hello
```

Funkcje mogą zostać dodane w bardziej wygodny sposób. Można utworzyć osobny plik z funkcjami i dodać go.

## Przykład

```
./plik_z_funkcjami
```

**lub**

```
source ~/plik_z_funkcjami
```

## Łączenie instrukcji w bloki

Można utworzyć blok funkcyjny dzięki, któremu można rozwiązywać niektóre problemy ułatwiając sobie tworzenie skryptu oraz poprawiając czytelność kodu.

### Przykład

```
{  
    Polecenia  
    ...  
}
```

## 2.2 Opis zadań

### 2.2.1 Zadanie 01

Celem zadania było napisanie funkcji maksimum, która wyświetli największy element

W zadaniu użyto instrukcji warunkowych **while** oraz **if**. Wpisywanie liczb odbywa się przy pomocy pętli **while**, dzięki której dopóki nie wstawimy znaku e oznaczającego exit możemy wpisywać dowolną ilość liczb, które następnie zapisane zostają w tablicy.

Na końcu skryptu uruchamiana jest nasza funkcja **MAX**, która porównuje każdą liczbę z liczbą aktualnie maksymalną, czy jest ona od niej większa, jeżeli tak, następuje przypisanie jej do wartości **MAX**.

W celu zademonstrowania jak działa nasz skrypt zostały wpisane z klawiatury liczby 21, 441, 6 i e. Maksymalna znaleziona liczba to 441.

### 2.2.2 Zadanie 02

Celem zadania było napisanie funkcji sortujących od najmniejszej do największej oraz sortującej od największej do najmniejszej.

Wypisanie liczb następuje z klawiatury przy pomocy funkcji **while**, aż do wpisania znaku e oznaczającego exit, który powoduje zakończenie wpisywania liczb.

Funkcja sortowania jest to typowe sortowanie bąbelkowe.

Algorytm działa następująco:

w każdym przejściu pętli wewnętrznej porównywane są ze sobą dwie kolejne wartości i w razie potrzeby są zamieniane miejscami. W jednym cyklu pętli wewnętrznej, największa liczba (tak jak bąbelki w napoju gazowanym) w zbiorze będzie się przemieszczała na ostatnią pozycję. W ten sposób otrzymujemy podzbiór częściowo już posortowany. Czynności te powtarzamy dla zbioru pominiętego o elementy już poukładane.

Sortowanie odbywa się dopóki liczby nie zostaną ułożone w odpowiedniej kolejności.

### 2.2.3 Zadanie 03

Celem zadania było przy pomocy funkcji obliczenie ilości słów w pliku tekstowym. Ponadto skrypt powinien mieć możliwość podania ścieżki do pliku z terminala.

Aby wykonać to ćwiczenie należało utworzyć plik tekstowy z dowolną ilością słów.

Komenda odpowiedzialna za zliczanie liczby słów to **wc**.

**wc -w** – komenda odpowiedzialna za zliczanie liczby słów w pliku.

Przy pomocy komendy **read** możemy wczytać lokalizację pliku, który potrzebujemy uruchomić.

### 2.2.4 Zadanie 04

Celem zadania było usunięcie części tekstu zapisanego z klawiatury.

Aby wykonać powyższe zadanie należało na początku utworzyć plik tekstowy z dowolnym tekstem w tym przypadku „Informatyka Studia Zaoczne”

Tworzenie pliku tekstowego wykonuje się poprzez utworzenie pliku za pomocą funkcji **touch**.

Zapisywanie tekstu tworzy się poprzez skierowanie naszej wiadomości do pliku za pomocą strzałki > skierowanej na docelowy plik.

Funkcja **cat** powoduje dołączenie pliku i wypisanie jego zawartości, aby zobaczyć pierwotny tekst.

Za pomocą funkcji **sed -e** zostaje usunięty tekst „Zaoczne”

Wynikiem programu jest tekst przed i po usunięciu:

- Przed: Informatyka Studia Zaoczne
- Po: Informatyka Studia

## 2.2.5 Zadanie 05

Celem zadania było wykorzystanie pliku z poprzedniego zadania i dodanie nowych funkcjonalności które mamy do wyboru:

- Zmiana tekstu na drukowane litery
- Usunięcie tekstu
- Zamiana liter

Wybór funkcjonalności został stworzony przy pomocy instrukcji case.

Tak jak w poprzednim zadaniu tworzenie pliku tekstowego odbywa się ten sam sposób. Funkcjonalność usunięcia tekstu została przedstawiona w zadaniu04.

W celu wyświetlenia tekstu drukowanymi literami należy użyć polecenia **tr**, która powoduje zamianę wszystkich liter małych od a do z na duże od A do Z (komenda **a-z A-z**).

W celu zamiany wszystkich liter „a” na dużą literę „B” należy również użyć komendy **tr** i zastosować podobny zapis jak dla formy wyżej (komenda **a B**)

## 3. Wnioski

Laboratorium nr 5 pozwoliło nam:

- Zapoznać się z funkcjami w bashu
- Tworzyć funkcje
- Poznać ich funkcjonalność

Zadania pozwoliły nam przypomnieć sobie wiedzę z zakresu tworzenia plików tekstowych, dołączania plików i wyświetlaniu ich na ekranie terminala, usuwaniu części tekstu, zamiany liter czy obliczaniu ilości słów w tekście.

Wiedza dotycząca funkcji na pewno ułatwi nam przeprowadzanie obliczeń oraz pisanie skryptów, a także pozwoli nam w efektywny sposób korzystać z Basha. Ponadto programowanie przy pomocy funkcji poprawia czytelność kodu.