

Programowanie Komputerów 2

Długości wyrazów

| | |
|-----------------------------|-----------------------|
| autor | Kuba Domerecki |
| prowadzący | Piotr Pecka |
| rok akademicki | 2019/2020 |
| kierunek | Informatyka |
| rodzaj studiów | stacjonarne (dzienne) |
| semestr | 2 |
| termin laboratorium | Środa 15:30 |
| termin oddania sprawozdania | 18.09.2020 |

1 Zadanie

Napisać program, który wczytuje plik tekstowy o dowolnym rozmiarze i wypisuje w pliku wynikowym statystykę długości wyrazów występujących w pliku oraz zadaną ilość najczęściej występujących słów o danych długościach. Spacje, przecinki, kropki itp. to wyrazy o długości 0.

Program uruchamiany jest z linii poleceń z wykorzystaniem następujących przełączników:

- i plik wejściowy
- n liczba najczęściej występujących słów
- o plik wyjściowy
- p drukuj wyjście na ekran

2 Analiza zadania

Zadanie polega na podaniu statystyki długości wyrazów.

2.1 Struktury

2.1.1 Struktura “List”

Struktura o nazwie “List” jest strukturą listy jednokierunkowej. Zadaniem tej listy jest przechowywanie każdego wyrazu z pliku. Dzięki zastosowaniu struktury listy jednokierunkowej nie trzeba znać wielkości pliku przed jego otwarciem, a każdy wyraz z pliku jest dodawany dynamicznie do listy. Lista zawiera wskaźnik do następnego elementu w liście, łańcuch znaków zawierający słowo listy, długość wyrazu oraz częstość pojawiania się tego wyrazu w liście.

2.1.2 Struktura “BinarySearchTree”

Struktura o nazwie “BinarySearchTree” jest strukturą drzewa poszukiwań binarnych zawierającą jeden dodatkowy element, który jest w wskaźniku na element tej samej wartości. Drzewo te można stworzyć na dwa sposoby. Pierwszym sposobem tworzenia tego drzewa jest częstość występowania danego wyrazu, drzewo te odpowiada za wyświetlanie najczęściej występujących słów. Drugim sposobem tworzenia tego drzewa jest długość wyrazu, dzięki temu możemy mieć statystykę długości wyrazów. Po stworzeniu drzewa na podstawie długości wyrazów drzewo jest skracane o potomka o tej samej długości, a częstość występowania jest dodawana do pierwszego potomka z tą samą długością. Dzięki tej strukturze można z łatwością wyświetlać posortowane dane.

2.2 Algorytmy

W programie wszystkie algorytmy dotyczące struktury drzewa są realizowane rekurencyjnie. Algorytmy dotyczące listy są realizowane głównie rekurencyjnie, przy dodawaniu nowego elementu do listy sprawdzane jest czy słowo które chcemy dodać do listy nie występuje już w niej, jeśli nie tworzony jest nowy element, jeśli tak częstość występowania danego wyrazu jest zwiększana o jeden. Każdy wyraz który ma pierwszą wielką literę, zamieniana jest ona na małą, aby słowa które są na początku zdania nie tworzyły nowego elementu. Algorytm usuwania listy jest realizowany rekurencyjnie.

3 Specyfikacja zewnętrzna

Program jest uruchamiany z linii poleceń należy przekazać do programu nazwy plików, liczbę najczęściej występujących słów i czy wyjście ma być drukowane na ekran (odpowiednio -i dla pliku wejściowego, -o dla pliku wyjściowego, -n dla liczby i -p dla drukowania na ekran), np.

```
program -i input.txt -o output.tx  
program -i input.txt -p -n 12  
program -p -n 2 -i input.txt -o output.txt
```

Przełączniki mogą być podane w odpowiedniej kolejności. Program można uruchomić używając: wyłącznie dwóch przełączników dla pliku wejściowego. Kiedy nie ma podanej liczby najczęściej występujących słów, program zignoruje tą część zadania i zapisze tylko statystykę wyrazów. Program musi mieć podany przełącznik dla pliku wejściowego oraz przełącznik pliku wyjściowego lub przełącznik do drukowania wyjścia na ekran. Uruchomienie programu bez parametru pliku wejściowego lub kiedy taki plik nie istnieje powoduje pojawienie się komunikatu:

error, file not exists or is not given

Natomiast podanie tylko parametru pliku wyjściowego powoduje pojawienie się komunikatu:

error, program will not run without the appropriate parameters

4 Specyfikacja wewnętrzna

Program został zrealizowany zgodnie z paradygmatem strukturalnym. W programie rozdzielono interfejs od logiki aplikacji.

4.1 Ogólna struktura programu

Po uruchomieniu programu, program na podstawie parametrów otwiera dane pliki i przekazuje je wraz z pobraną zmienną dotyczącą ilości najczęściej występujących słów do funkcji `run`. Następnie dodaje słowa do listy funkcją `getWords` i zamyka plik wejściowy, potem tworzy drzewo długości wyrazów i w zależności od parametru drzewo częstotliwości występowania wyrazów. Później zapisuje drzewa do pliku i w zależności od parametru wypisuje na ekranie wyniki oraz usuwa drzewa i zamyka plik.

5 Testowanie

Program został stworzony w środowisku programistycznym Visual Studio 2019. Program można otworzyć na każdym rodzaju pliku. Program został sprawdzony pod kątem wycieków pamięci. Program działa dla mniej niż 240 kB'ego pliku. Program źle określa długość polskich wyrazów.

Program wykonywał się poprawnie na komputerach z Windows 10.

6 Wnioski

Projekt w języku C, był trudniejszy do napisania niż projekt w języku C++ w poprzednim semestrze. Dzięki napisaniu tego projektu utrwaliłem sobie tworzenie struktur danych i przetwarzanie ich w tym języku. Najtrudniejszą rzeczą było przekazywanie parametrów funkcji poprzez referencje, ponieważ różni się to w języku C. Muszę przyznać że mimo kilku problemów zmuszających do dłuższego zastanowienia się, projekt pisało się bardzo przyjemnie.