

Politechnika Śląska w Gliwicach
Wydział Automatyki, Elektroniki i Informatyki



Tłumacz

Autor : Paweł Wieczorek

Prowadzący : dr inż. Adam Gudys

Rok akademicki : 2016/2017

Kierunek : Informatyka

Rodzaj studiów : SSI

Semestr : 1

Termin laboratorium : czwartek, 13:00 – 14:30

Grupa : 3

Termin oddania sprawozdania : 2017-01-25

Data oddania sprawozdania : 2017-01-23

1 Treść zadania

Napisać program, który tłumaczy tekst z języka polskiego na angielski i z powrotem. Tłumaczenie polega na zastępowaniu słów języka wyjściowego słowami języka docelowego. Jest to zatem sposób dość niedoskonały. Przy implementacji należy skorzystać z drzewa binarnego. Drzewo binarne zbudowane dla języka polskiego zbudowane jest w ten sposób, że wyrazy, które stoją wcześniej w porządku alfabetycznym są po lewej stronie węzła, a te, które stoją później – po prawej stronie. Te same elementy, które należą do drzewa dla języka polskiego, są także elementami drzewa dla języka angielskiego budowanego według porządku alfabetycznego wyrazów angielskich.

2 Analiza zadania

Zagadnienie przedstawia problem przetłumaczenia danego w zadaniu tekstu.

2.1 Struktury danych

W programie wykorzystano drzewo binarne do przechowywania wyrazów polskich i angielskich. Drzewo osobno wskazuje kolejne drzewa dla wyrazów angielskich i polskich.

2.2 Algorytmy

Program sortuje wyrazy alfabetycznie poprzez umieszczenie je w drzewie binarnym. Po lewej stronie znajdują się wyrazy późniejsze w alfabecie, a po prawej wcześniejsze.

3 Specyfikacja zewnętrzna

Program jest uruchamiany z linii poleceń. Należy przekazać do programu nazwy plików: wejściowego, wyjściowego i słownikowego po odpowiednich przełącznikach (odpowiednio: -i dla pliku wejściowego, -o dla pliku wyjściowego i -s dla pliku słownikowego), a także kierunek tłumaczenia po przełączniku -k. Przełączniki mogą być podane w dowolnej kolejności. Uruchomienie programu z nieprawidłowymi parametrami spowoduje wyświetlenie odpowiedniego komunikatu, a także krótkiej pomocy.

4 Specyfikacja wewnętrzna

Program został zrealizowany zgodnie z paradygmatem strukturalnym. W programie rozdzielono interfejs (komunikację z użytkownikiem) od logiki aplikacji (sortowania liczb).

4.1 Typy zdefiniowane w programie

W programie zdefiniowano następujący typ:

```

struct drzewo
{
    std::string pol;
    std::string ang;
    drzewo *pp;
    drzewo *pl;
    drzewo *ap;
    drzewo *al;
};

```

Ten typ służy do zbudowania drzewa.

4.2 Ogólna struktura programu

W programie głównym program sprawdza czy został on wywołany w prawidłowy sposób. Gdy program nie został prawidłowo wywołany wypisuje komunikat i program się kończy. Po sprawdzeniu parametrów w zmiennych si, so, ss, sk przechowywane są odpowiednio nazwy pliku wejściowego, pliku wyjściowego, pliku słownikowego i kierunku tłumaczenia. W dalszej części program wywołuje funkcje

void tłumacz(string si, string so, string ss, string sk);

Funkcja tłumaczy tekst z pliku wejściowego do pliku wyjściowego. Po wykonaniu funkcji program się kończy.

4.3 Szczegółowy opis implementacji funkcji

void tłumacz(string si, string so, string ss, string sk);

Funkcja tłumaczy plik wejściowy si i zapisuje go do pliku wyjściowego so. Kierunek tłumaczenia pokazuje zmienna sk. Plik słownikowy ss jest przepisywany do drzewa binarnego.

string plen(drzewo * g, string pl)

Funkcja wyszukuje w drzewie binarnym o głowie g tłumaczenia słowa pl.

string enpl(drzewo * g, string en)

Funkcja wyszukuje w drzewie binarnym o głowie g tłumaczenia słowa en.

void UsunDrzewoEng(drzewo * g)

Funkcja usuwa drzewo binarne o głowie g.

drzewo* Dodajwyraz(drzewo * g, string pol, string ang)

Funkcja dodaje na drzewo binarne wyrazy pol i ang. Zwraca wskazanie na nowo utworzoną część drzewa.

void Dodajang (drzewo * g, drzewo * v)

Funkcja do drzewa binarnego o głowie g dołącza idąc angielskimi wskaźnikami na wyraz który jest wskazywany przez v.

void help()

Funkcja wypisuje krótką pomoc.

5 Testowanie

Program został przetestowany na różnego rodzaju plikach. Pliki niepoprawne powodują zgłoszenie błędu.

6 Wnioski

Program do tłumaczenia tekstu jest programem prostym, chociaż wymaga samodzielnego zarządzania pamięcią.