

Gliwice, 15 kwietnia 2017

Programowanie Komputerów

Temat: Przedrostki

Autor: Dariusz Opitek
Semestr: drugi
Grupa: I
Prowadzący: Tomasz Drosik

1. Analiza rozwiązania

Głównym zadaniem programu jest znalezienie wszystkich słów o zadanym przedrostu. Słowa zostają wyszukane w podanym pliku wejściowym i następnie w kolejności alfabetycznej wpisane do pliku wyjściowego.

Aby nie wczytywać całego pliku od razu, by zapobiec używaniu zbyt dużej ilości pamięci operacyjnej, program będzie wczytywał po jednej linii a następnie dokona wyszukiwania pasujących słów i wpisze je do listy wynikowej.

W programie zostaną użyte dwie struktury dynamiczne. Do przechowywania wszystkich słów znalezionych w linii i następnie do przechowywania pasujących słów zostanie użyty stos. Dla końcowej listy wynikowej zawierającej pasujące słowa z całego pliku zostanie użyta lista samo sortująca się.

2. Specyfikacja zewnętrzna

Program należy uruchomić z linii poleceń w następujący sposób:

Windows: `przedrostki.exe -i <plik wejściowy> -o <plik wyjściowy> -p <przedrostek>`

Linux: `./przedrostki -i <plik wejściowy> -o <plik wyjściowy> -p <przedrostek>`

Jeżeli program nie jest uruchamiany z katalogu, w którym się znajduje należy podać do niego pełną ścieżkę, tak samo należy postępować z plikiem do przetworzenia. W przypadku, gdy ścieżka zawiera nazwy katalogów ze spacjami należy wziąć ją w cudzysłów.

W programie jest dostępna pomoc, którą można uzyskać za pomocą przełącznika `-h`.

Po uruchomieniu programu, jeżeli wszystko przebiegło pomyślnie, program zakończy swoje działanie bez żadnego komunikatu.

Jeżeli program napotka na problem poinformuje o tym użytkownika za pomocą następujących komunikatów:

- Nie podano przedrostka. Podaj przedrostek z opcją `-p`. Np. `-p pre`
- Nie podano pliku wejściowego. Podaj plik wejściowy z opcją `-i`. Np. `-i plik_wejscowy.txt`
- Plik wejściowy nie istnieje. Sprawdź nazwę pliku
- Nie można otworzyć pliku wejściowego do odczytu
- Nie podano pliku wyjściowego. Podaj plik wyjściowy z opcją `-o`. Np. `-o plik_wyjsciowy.txt`
- Nie można otworzyć pliku wyjściowego do zapisu
- Brak dostępnej pamięci

3. Specyfikacja wewnętrzna

- `core/core.h`

```
void process(model model_t);
```

Główna funkcja programu. Funkcja wczytuje plik wejściowy, analizuje tekst i zapisuje wyniki w pliku wyjściowym.

```
void process_line(char *line, char* prefix, sorted_list* results);
```

Analiza pojedynczej linii wczytanej z pliku. Wyszukane słowa z przedrostkami zostają wpisane do samo sortującej się listy.

- `core/menu.h`

```
model menu(int argc, char ** argv);
```

Główna funkcja odpowiedzialna za parsowanie argumentów przekazanych do programu i wyświetlenie odpowiedniego komunikatu w razie błędu.

```
int validate_input(model model_t);
```

Walidacja wszystkich parametrów wczytanych do struktury model.

- core/prefix.h

```
int has_prefix_ignore_case(char const *text, char const *prefix);
```

Funkcja sprawdzająca czy podane słowo zawiera podany przedrostek.

```
stack* select_many_with_prefix(stack ** matches, char *prefix);
```

Wyszukanie z podanej listy słów, słów zawierających podany przedrostek.

- core/common.h

```
struct model
```

Struktura używana do przechowywania parametrów podanych z wywołaniem programu.

- collections/sortedlist.h

```
struct sorted_list
```

Struktura używana do przechowywania posortowanych ciągów znaków (słów).

```
sorted_list* initialize_list();
```

Utworzenie samo sortującej się listy.

```
void add_element(sorted_list* head, char* text, int (*comparer)(const char*, const char*));
```

Dodanie elementu do listy. Element jest od razu wstawiany w odpowiednie miejsce w posortowanej liście. Do funkcji jako ostatni parametr należy podać funkcję porównującą.

```
void destroy_list(sorted_list* head);
```

Dealloc pamięci dla listy.

```
void save_list(sorted_list* head, char *file_name);
```

Zapis listy do pliku.

- collections/stack.h

```
struct stack
```

Struktura używana do przechowywania ciągów znaków na stosie.

```
stack* initialize_stack();
```

Utworzenie stosu.

```
void push(stack** head, char* object);
```

Wrzucenie element na stos.

```
char* pop(stack** head);
```

Pobranie element ze stosu i jednocześnie dealokacja pamięci.

```
void dealloc_stack(stack* head);
```

Dealokacja pamięci dla stosu.

- extensions/fileext.h

```
char* read_line(FILE* file);
```

Wczytanie linii z pliku. Jeśli linia jest dłuższa od zaalokowanej pamięci to następuje realokacja.

```
int file_exists(const char *fileName);
```

Sprawdzenie czy plik istnieje.

- extensions/memoryext.h

```
void realloc_if_exceeds(char** buffer, size_t current_position, size_t* size);
```

Realokacja pamięci w przypadku dotarcia do końca zaalokowanej pamięci.

- extensions/stringext.h

```
int strcmp_ignore_case(char const *a, char const *b);
```

Porównanie dwóch ciągów znaków bez względu na wielkość poszczególnych znaków.

```
stack* split(char *line, char *delimiter);
```

Podział linii na słowa.

4. Testowanie

Program został przetestowany pod dwoma względami.

- 1) Sposób pierwszy to błędne wprowadzenie parametrów, bądź brak pliku do odczytu, lub brak praw do zapisu.
- 2) Program został przetestowany pod względem poprawności otrzymanych wyników.

Ad. 1)

Przykładowe wywołania program i otrzymane wyniki

- Wywołanie: przedrostki.exe -i nie_ma_takiego_pliku.txt -o out.txt -p pr
Wynik: Plik wejściowy nie istnieje. Sprawdź nazwę pliku
- Wywołanie: przedrostki.exe -i in.txt -o plik_tylko_do_odczytu.txt -p pr
Wynik: Nie można otworzyć pliku wyjściowego do zapisu
- Wywołanie: przedrostki.exe -i in.txt -o out.txt -p
Wynik: Nie podano przedrostka. Podaj przedrostek z opcją -p. Np. -p pr
- Wywołanie: przedrostki.exe -i -o out.txt -p pr
Wynik: Nie podano pliku wejściowego. Podaj plik wejściowy z opcją -i. Np. -i plik_wejsciowy.txt

- Wywołanie: przedrostki.exe -i -p pr
Wynik: Nie podano pliku wejściowego. Podaj plik wejściowy z opcją -i. Np. -i plik_wejsciowy.txt"

Ad. 2)

- Wygenerowany został plik lorem_ipsum.txt z 10 tysiącami słów. Wyszukiwano przedrostek „co”.

Następujące słowa zostały znalezione.

commodo
condimentum
congue
consectetur
consequat
conubia
convallis

- Przetestowano program na książce Pana Tadeusza. Wyszukiwano przedrostek „świę”

Następujące słowa zostały znalezione:

święci
święconą
świętą
święta
świętego
świętej
świętem
świętościami
święto
święty
świętych
świętym

Pliki lorem_ipsum.txt i pan_tadeusz.txt są dostępne w folderze z programem.