Projekt 3: Program usuwający wszystkie nadmiarowe spacje, tabulatory oraz znaki nowej linii z pliku.

Autor: *Bartosz Ruta*Informatyka I stopnia III sem., 2019/20

Programowanie Niskopoziomowe

**1. Cel projektu**

Celem projektu było napisanie prostego programu w języku assemblera usuwającego wszystkie nadmiarowe spacje, tabulatory oraz znaki nowej linii z pliku podanego przez użytkownika.

**2. Założenia**

Program okienkowy usuwający nadmiarowe spacje, tabulatory oraz znaki nowej linii z pliku o adresie podanym przez użytkownika. Aplikacja wyświetla rozmiar pliku źródłowego, zlicza oraz usuwa powtarzające się znaki: nowej linii oraz 2 znaki (znaki ASCII - domyślnie spacja oraz poziomy tab) podane przez użytkownika. Następnie wyświetla całkowitą ilość usuniętych znaków i rozmiar pliku po wykonaniu operacji.

**3. Ograniczenia**

W celu ograniczenia ewentualnego zużycia pamięci operacyjnej dla dużych plików program zamiast umieszczać poprawiony tekst w pamięci RAM, zapisuje docelowy tekst do nowego pliku tekstowego, który znajduje się w lokalizacji programu, a następnie przenoszony jest w miejsce pliku źródłowego.

Znakami, które obsługuje program są znak nowej linii(CR,LF), a także 2 dowolne znaki ASCII - o numerach od 0 do 127, podanie numerów spoza zakresu skutkuje wyświetleniem odpowiedniego komunikatu.

Podanie nieprawidłowej lub niepełnej ścieżki do pliku spowoduje wyświetlenie komunikatu o błędzie.

Bufory zapisujące poprzedni i aktualny znak mają rozmiar 1 bajta – zawartość pliku zapisywana w pamięci operacyjnej ogranicza się tylko do tych zmiennych.

**4. Wstęp**

Algorytm usuwania powtórzeń wymaga pobrania i zapamiętania jednego znaku z pliku. Następnie pobiera z pliku następny znak i porównuje go z poprzednim. Jeśli jest to znak, którego powtórzeń mamy szukać, to porównujemy go ze znakiem poprzednim. Gdy są takie same zwiększamy licznik usuniętych znaków, jeśli inne zapisujemy znak do pliku pomocniczego.

W przypadku znaku nowej linii także bierzemy pod uwagę poprzedni znak, ale gdy aktualnie pobrany znak ma kod 13, a poprzedni ma kod różny od 10 to zapisujemy go do pliku, a gdy poprzednim znakiem był kod 10, to czytamy następny znak i jeśli ma on kod równy 10, to zwiększamy tylko licznik usuniętych znaków, jeśli jest tam coś innego, to zapisujemy taki znak do pliku.

**5. Implementacja**

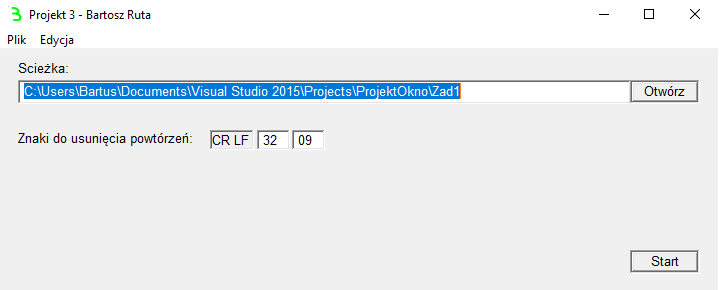
Do realizacji zadania potrzebna była obsługa plików – skorzystano z procedur: CreateFileA, WriteFile, ReadFile, DeleteFileA, MoveFileA, CloseHandle, GetCurrentDirectoryA. Do obsługiwania aplikacji okienkowej korzystałem z metod SetDlgItemTextA, SendDlgItemMessageA, GetLastError.

Zasada działania programu:

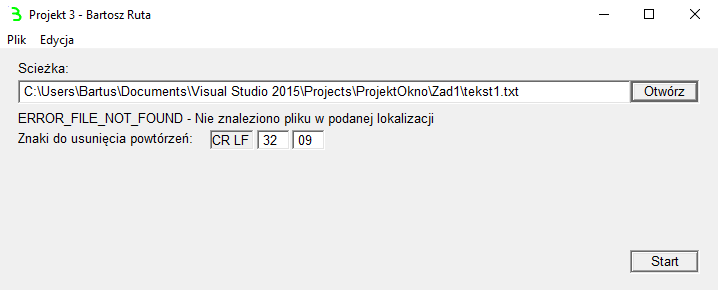
1. W pole edycyjne wpisujemy ścieżkę do pliku źródłowego. Domyślnie znajduję się tam lokalizacja programu. Po kliknięciu przycisku otwórz wyświetla się komunikat o powodzeniu bądź niepowodzeniu otwarcia pliku i jego rozmiar.
2. W kolejne 2 pola edycyjne wpisujemy liczby reprezentujące kody znaków ASCII, których nadmiarowe wystąpienia chcemy usuwać.
3. Po kliknięciu przycisku Start, za pomocą pętli WHILE algorytm sczytuje kolejne znaki pliku źródłowego, porównuje aktualnie sczytywany znak z poprzednim i gdy się różnią dopisuje go do pliku tymczasowego. W przeciwnym wypadku zwiększany jest licznik usuniętych znaków.
4. Wyświetlenie informacji o rozmiarze pliku docelowego i ilości usuniętych znaków, usunięcie pliku źródłowego i przeniesienie pliku docelowego w jego miejsce.
5. Zamknięcie uchwytów do plików i czekanie na dalsze kroki użytkownika.

**5. Przykłady działania programu**

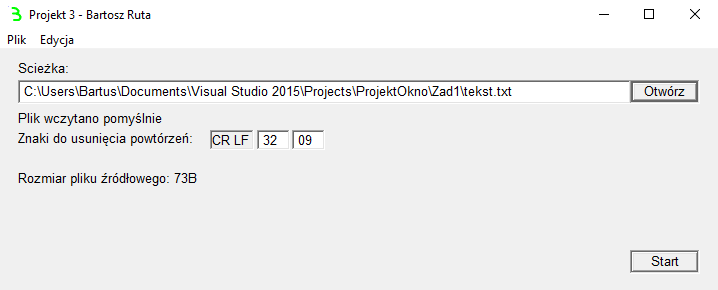
1. Okno programu bezpośrednio po uruchomieniu lub po kliknięciu przycisku Edycja->Wyczyść:



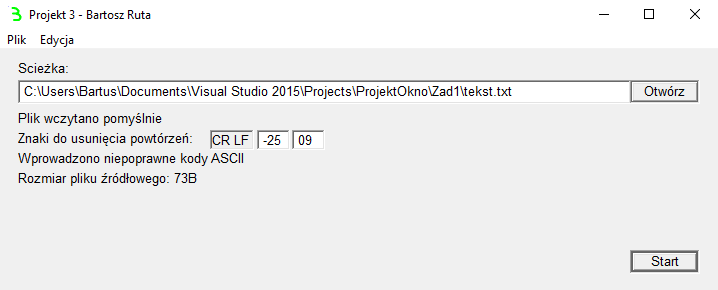
1. Okno programu po naciśnięciu przycisku Otwórz podając niepoprawną ścieżkę do pliku:



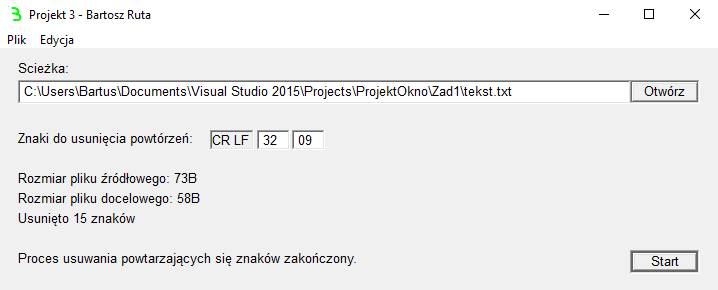
1. Okno programu po pomyślnym wczytaniu pliku:



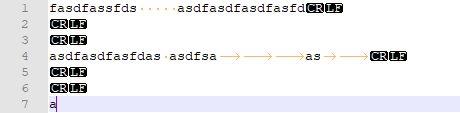
1. Okno programu po podaniu niepoprawnego kodu znaku ASCII



1. Okno programu po wprowadzeniu poprawnych kodów ASCII i kliknięciu przycisku Start:



Plik źródłowy:



Plik docelowy:

