

Project #1

Description: This program will read a text file and count the number of occurrences of each word. It saves this data using three different data structures (parallel arrays, array of objects and a sequential container object). The user can decide what data structure he wants.

Reason for using the vector: One data structure in this project is the sequential container object. In C++ there are different kind of sequential containers, like lists or vectors. The sequential container is implemented with a vector because of the dynamically increased size what makes it easier to implement a solution.

Explanation of the logic at based on the parallel array: The application will go through each word in the file and compares this word with every element in the array. If there is already a word that matches the word the application is looking at, the program will increment the count of the occurrences. If the word is not already in the array, the program will store the word at the first element which is empty if there is still room to store it (capacity is set to 10). There is also a variable called 'found' which is set to true if the word is found in the first for-loop. With this variable it is possible to know if the second for-loop (for adding the word) is needed.

Possible improvements: It could be possible to search for a word in an array without the use of a 'found'-variable. Although the array or the vector is empty, the program will still go through every element. That could be a possible improvement in the future as well.

Content of the application:

- p1.cpp
- p1
- WordData.cpp
- WordData.h
- Makefile.txt
- words.txt
- words2.txt
- words3.txt