SE375 - Laboratory Assignment 03

Part 1: Word Count with Threads and Shared Data Structure

You have developed the Word Count application in the previous laboratory assignment using threads and an array of data structures. Your task now is to write the same Java application with the threads, this time sharing the same instance of a data structure instead of working on separate instances.

You should follow these steps in your Java applications: (1) For each file passed as a parameter, you will need a thread to parse the file and a shared data structure passed to the threads to save the results. (2) Each file reads the file assigned to it, and uses the shared data structure to add or update the words. (3) Once all threads are complete, ask the user for a word they are looking for, and will print the number of appearances of that word.

Consider the flow diagram given in Figure 1. Although the figure is for n number of files, assume that three files have been passed as arguments ($n = 3$). In the main method, a shared data structure is created and its reference is passed to each thread along with the file names from the *args* parameter. Each thread works on the shared data structure and the file they are assigned to. Once they are all complete, you let the user query the data structure for specific words.

Sample Run

```
C:\java ThreadedWordCount file1 file2 file3
Thread parsing file2...
Thread parsing file3...
Thread parsing file1...
Threads are complete.
Enter word: programming
The word programming appears 19 times.
```

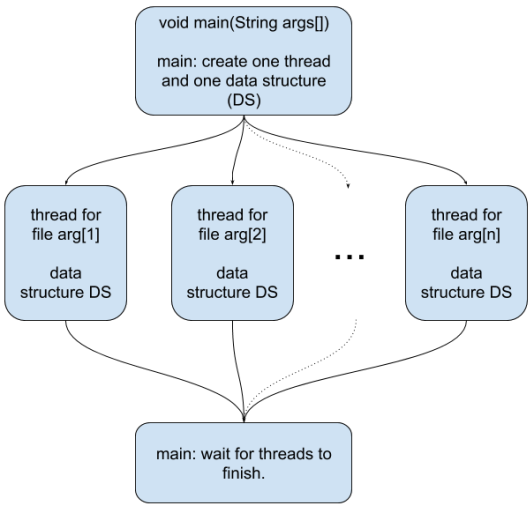

Figure 1: The flow diagram for both Word Count and Inverted Index.

Part 2: Inverted Index with Threads and Shared Data Structure

Repeat the same approach for the Inverted Index. This time, you are asked to find the files the word appears in. Use one thread per file with a shared data structure passed as argument to each thread. Once the threads are complete, let the user query the data structure.

```
C:\java WordCount file1 file2 file3
Thread parsing file1...
Thread parsing file2...
Thread parsing file3...
Threads are complete.
Enter word: programming
The word programming appears in: file1, file3
```