**Message Processor**

By: Deepak Swaroop

Email: [deepakswaroop1@hotmail.com](mailto:deepakswaroop1@hotmail.com)

Objective :

Create a Message Processor module that will read data from multiple text file in a directory in particular format and process that data and create a output file based without duplicates and in order.

Also messages should be processed as per the set limit, like X messages in Y seconds. If message limit is exhausted then wait till next time slot. Like 100 messages in 10 seconds. If 100 messages reached before 10 seconds, then wait for 10 seconds to complete and then execute again.

**Tools Used :**

IDE : CodeBlocks

Platforms : Windows10

Compiler : MinGW

Language : C++11/C++14

Compiler command :

**-------------- Build: Release in MessageProcessor (compiler: GNU GCC Compiler)---------------**

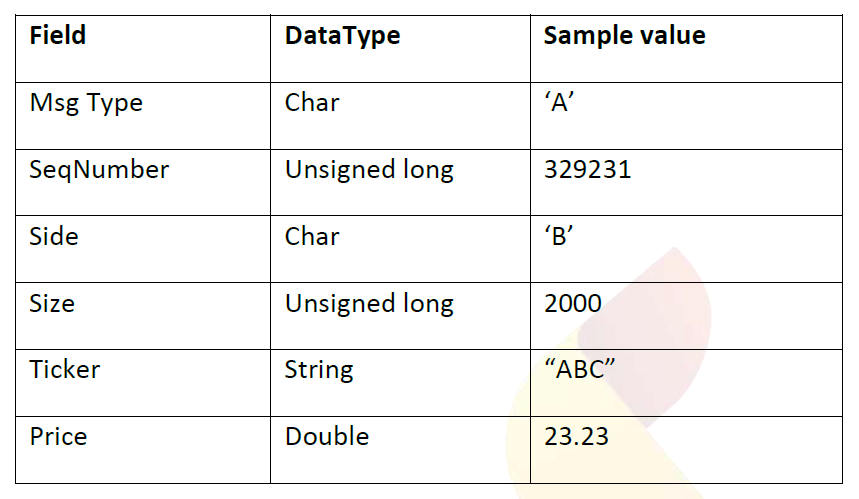
g++.exe -Wall -fexceptions -O2 -std=c++11 -std=c++14 -I..\MessageProcessor -c C:\Users\dswaroop\Documents\DeepakCoding\MessageProcessor\MessageProcessor\main.cpp -o obj\Release\main.o

g++.exe -Wall -fexceptions -O2 -std=c++11 -std=c++14 -I..\MessageProcessor -c C:\Users\dswaroop\Documents\DeepakCoding\MessageProcessor\MessageProcessor\src\MessageProcessor.cpp -o obj\Release\src\MessageProcessor.o

g++.exe -o bin\Release\MessageProcessor.exe obj\Release\main.o obj\Release\src\MessageProcessor.o -s -static-libstdc++ -static-libgcc -static

Output file is bin\Release\MessageProcessor.exe with size 838.50 KB

Message Structure:

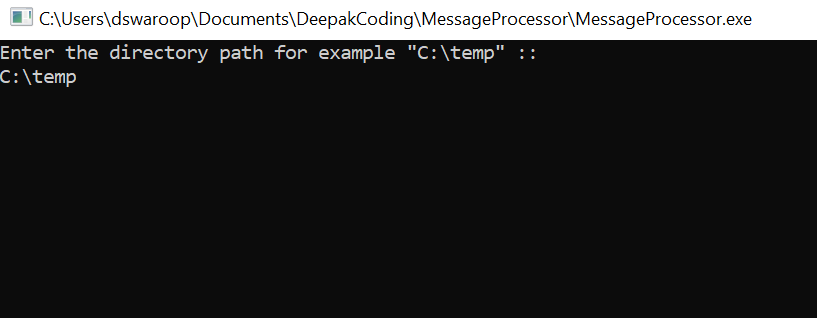


Files Included in tar file :

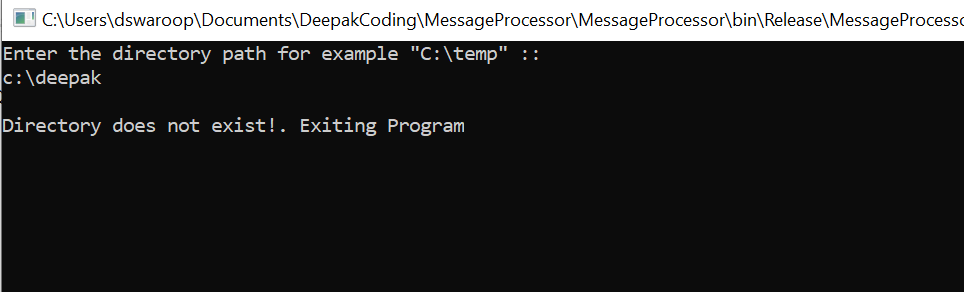
1. MessageProcessor code blocks project file including source code.
2. MessageProcessorInputData folder containing test files as input data. Just copy them to your mentioned directory.
3. MessageProcessor.exe standalone executable to execute code

Usage :

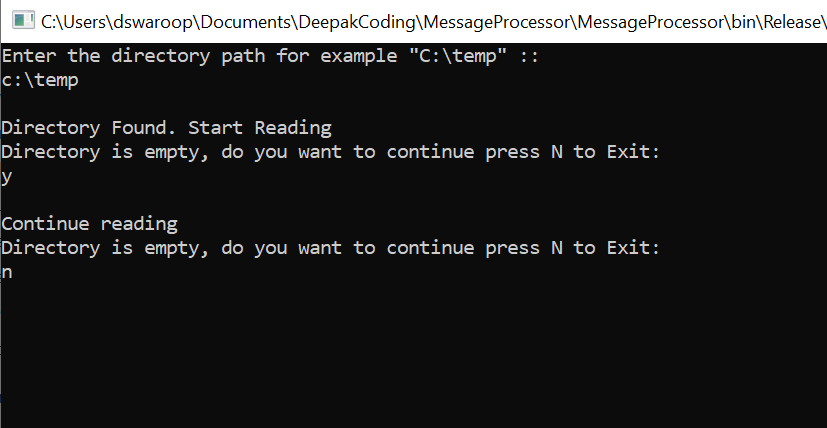
1. Execute MessageProcessor.exe it open like below and enter the directory from which you want to read data.



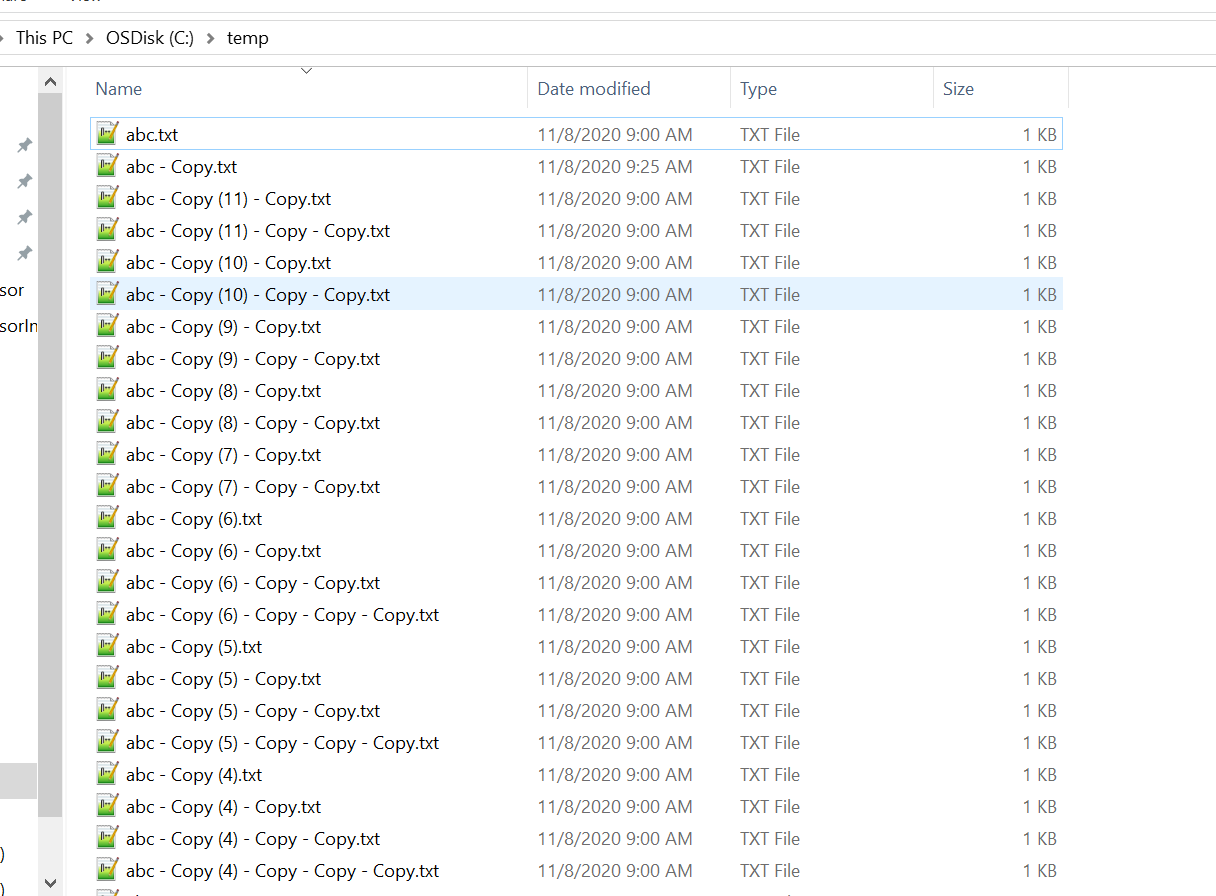
1. After you enter directory, it will first check whether directory exists or not. If not it will give exit

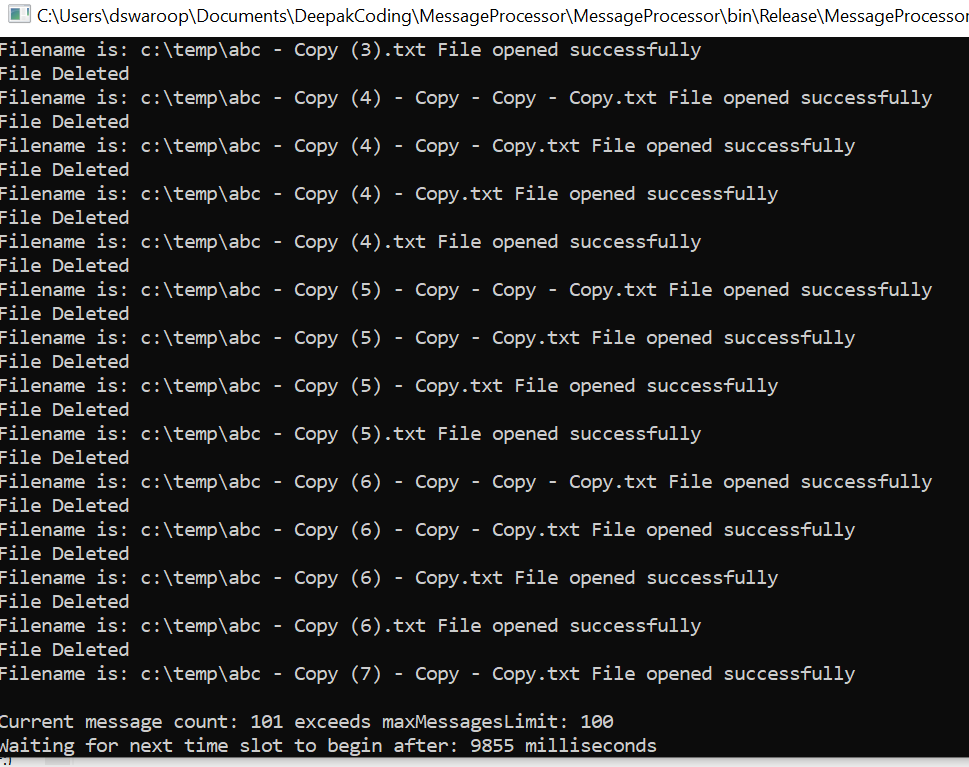


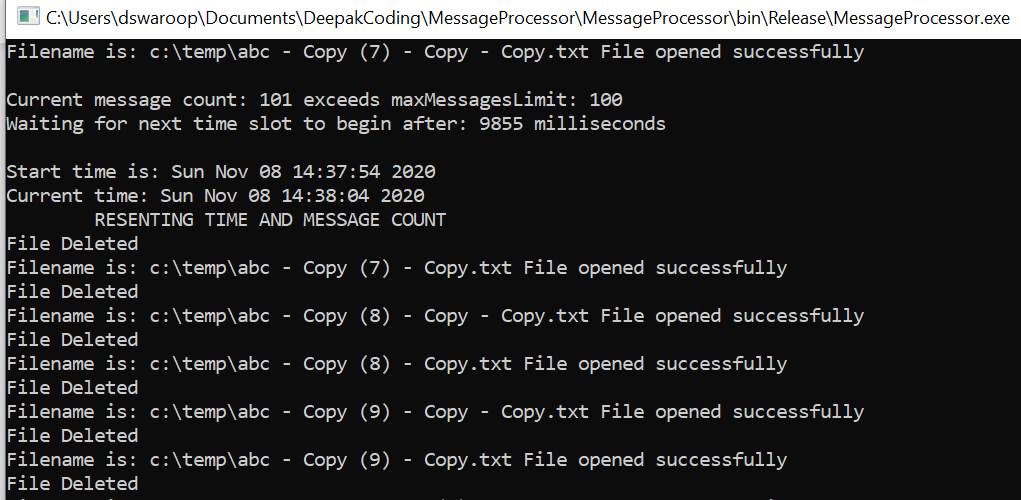
1. Also it will check for whether there are any files in directory or not. If not it will wait for files and ask you to continue. As below

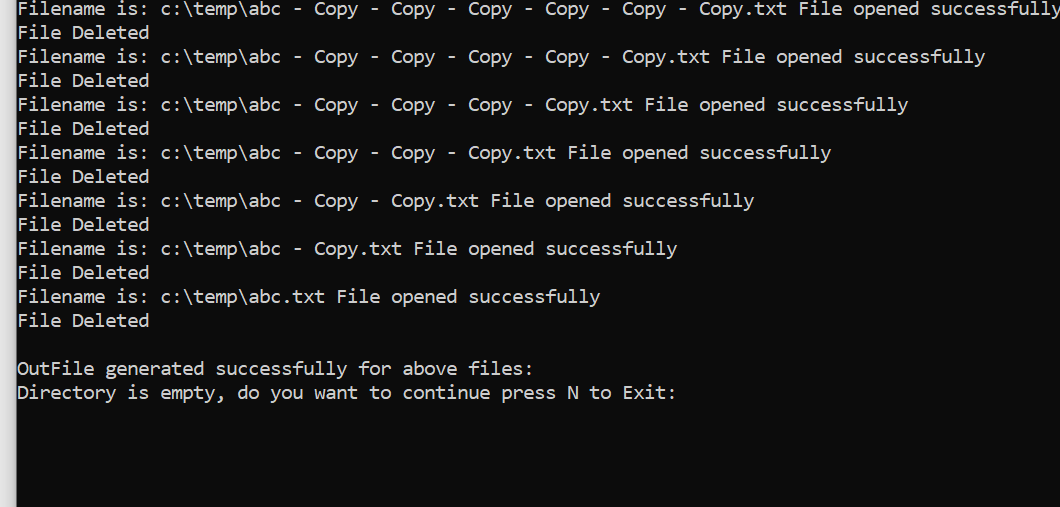


1. Now if files are present it will read data as per the rate limit and wait as well as per below images. For test purpose you can use input files from folder MessageProcessorInputData and copy them to C:\temp.

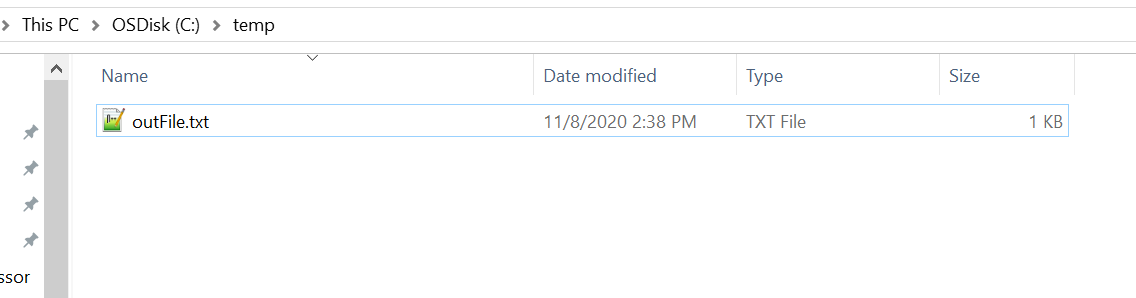




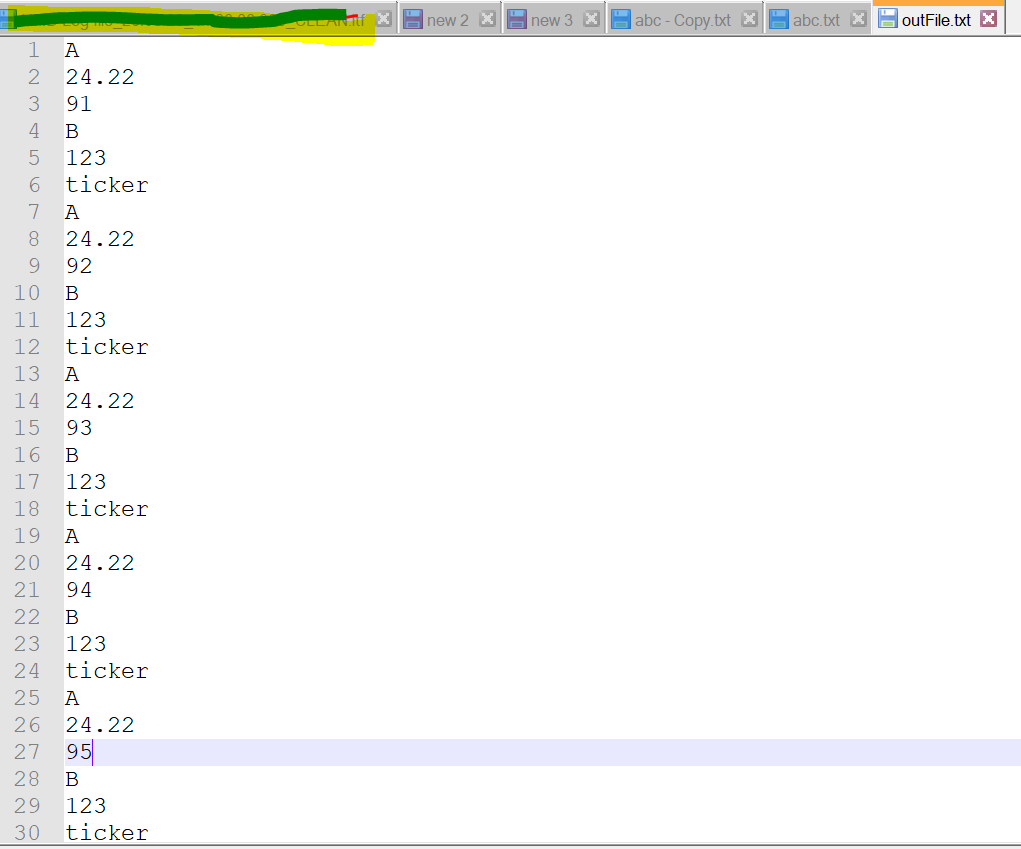




1. It will also delete the files once data has been read from that file. As show below directory is not empty and only outFile.txt is there.



1. Message in outfile is stored in increasing order of sequence number :



1. Important Messages in input file is stored in below order :

newFile<<message1.msgType;

newFile<<message1.price;

newFile<<message1.seq\_number;

newFile<<message1.side;

newFile<<message1.size;

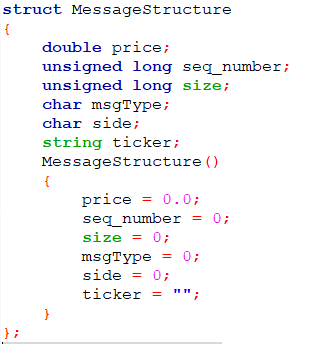
newFile<<message1.ticker;

**CODE WALKTHROUGH:**

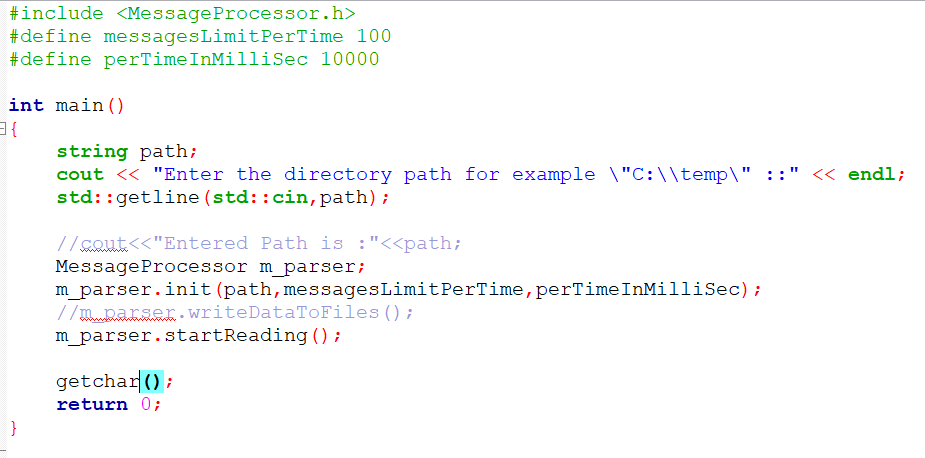
1. Class :



1. Message Structure :



1. Main.cpp :



THANK YOU!