C# test task

Implement a command line tool using C# for block-by-block compressing and decompressing of files using class System.IO.Compression.GzipStream.
During compression source file should be split by blocks of the same size, for example, block of 1MB. Each block then should be compressed and written to the output file independently of others blocks. Application should effectively parallel and synchronize blocks processing in multicore environment and should be able to process files, that are larger than available RAM size.
Program code must be safe and robust in terms of exceptions. In case of exceptional situations user should be informed by user friendly message, that allows user to fix occurred issue, for example, in case of OS limitations.
Only basic classes and synchronization objects should be used for multithreading (Thread, Manual/AutoResetEvent, Monitor, Semaphore, Mutex), it is not allowed to use async/await, ThreadPool, BackgroundWorker, TPL.
Source code should satisfy OOP and OOD principles (readability, classes separation and so on).
Use the following command line arguments:

- compressing: GZipTest.exe compress [original file name] [archive file name]
- decompressing: GZipTest.exe decompress [archive file name] [decompressing file name]

On success program should return 0, otherwise 1.
Note: format of the archive is up to solution author, and does not affects final score, for example there is no requirement for archive file to be compatible with GZIP file format.
Please send us solution source files and Visual Studio project. Briefly describe architecture and algorithms used.