Deflate

2.0

Generated by Doxygen 1.9.7

Chapter 1

README

Fundamentals of Computer Programming S1 project, ZiP (deflate Algorithm)

This project is an implementation of Deflate (i.e., LZ77 and Huffman Coding) Encoding Algorithm for text input files. The executable file after compiling the project is named deflate.

You can compress and decompress text input files with Deflate with this project. This algorithm uses LZ77 and Huffman Coding algorithms respectively and produces a smaller size output file which can be recovered later.

The commands for these operations are:

./deflate -c <input file> <output file>

and

./deflate -d <input file> <output file>

and the contents of input.txt and out_decompress.txt files would be the same.

(as you can see I tried to build using Cmake but i did not have time to learn fully how it works to be able to implement it, i left the files to be able to work on them in the future)

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 4

Class Documentation

4.1 CommandHandler Class Reference

command handler header file to call the commandHandler.cpp from main.cpp file

`#include <CommandHandler.h>`

Public Member Functions

- CommandHandler (string inputFile, string outputFile, string command)
- void execute ()

  checks the command, outputs error if not the right command and runs the functions related to either compressing or decompressing the file also writes the new output file

4.1.1 Detailed Description

command handler header file to call the commandHandler.cpp from main.cpp file

Definition at line 10 of file CommandHandler.h.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 CommandHandler()

```
CommandHandler::CommandHandler (
            string inputFile,
            string outputFile,
            string command )  [inline]
```

Definition at line 12 of file CommandHandler.h.

```
00012 : inputFile(inputFile), outputFile(outputFile), command(command) {}
```

4.1.3 Member Function Documentation

4.1.3.1 execute()

```
void CommandHandler::execute ( )
```

checks the command, outputs error if not the right command and runs the functions related to either compressing or decompressing the file also writes the new output file

compressing: read input, lz77 compression, make new tree, huffman compression, write output in new file decompression: read input, huffman decompression, lz77 decoding, write output in .txt file

Definition at line 16 of file CommandHandler.cpp.

```
00016                                {
00017     vector<string> result;
00018     string inputFileContent;
00019     FileIO::read(inputFile, inputFileContent);
00020
00021     if (command == "-c" ) {
00022         string lzCompressed;
00023         LZ77Codec::compress(inputFileContent, &lzCompressed);
00024
00025         Tree *tree = new Tree();
00026         tree->makeTree(lzCompressed);
00027
00028         string firstLine, huffmanCompressed;
00029         HuffmanCodec::compress(tree, &firstLine, &huffmanCompressed, lzCompressed);
00030
00031
00032         result.push_back(firstLine);
00033         result.push_back(huffmanCompressed);
00034         FileIO::write(outputFile, result);
00035     }
00036
00037     else if(command == "-d") {
00038         size_t lineBreak = inputFileContent.find('\n');
00039         string keys = inputFileContent.substr(0, lineBreak);
00040         string compressed = inputFileContent.substr(lineBreak + 1);
00041
00042         string huffmanDecompressed;
00043         HuffmanCodec::decompress(keys, compressed, &huffmanDecompressed);
00044
00045         string lzDecompressed;
00046         LZ77Codec::decompress(huffmanDecompressed, &lzDecompressed);
00047
00048         result.push_back(lzDecompressed);
00049         FileIO::write(outputFile, result);
00050     }
00051
00052     else {
00053         cout « "Wrong Command" « endl;
00054     }
00055 }
```

The documentation for this class was generated from the following files:

- CommandHandler.h
- CommandHandler.cpp

Chapter 5

File Documentation

5.1 CommandHandler.cpp

```
00001 #include "CommandHandler.h"
00002 #include "tree/Tree.h"
00003 #include "operations/Huffman.h"
00004 #include "operations/FileIO.h"
00005 #include "operations/lz77.h"
00006
00016 void CommandHandler::execute() {
00017     vector<string> result;
00018     string inputFileContent;
00019     FileIO::read(inputFile, inputFileContent);
00020
00021     if (command == "-c" ) {
00022         string lzCompressed;
00023         LZ77Codec::compress(inputFileContent, &lzCompressed);
00024
00025         Tree *tree = new Tree();
00026         tree->makeTree(lzCompressed);
00027
00028         string firstLine, huffmanCompressed;
00029         HuffmanCodec::compress(tree, &firstLine, &huffmanCompressed, lzCompressed);
00030
00031
00032         result.push_back(firstLine);
00033         result.push_back(huffmanCompressed);
00034         FileIO::write(outputFile, result);
00035     }
00036
00037     else if(command == "-d") {
00038         size_t lineBreak = inputFileContent.find('\n');
00039         string keys = inputFileContent.substr(0, lineBreak);
00040         string compressed = inputFileContent.substr(lineBreak + 1);
00041
00042         string huffmanDecompressed;
00043         HuffmanCodec::decompress(keys, compressed, &huffmanDecompressed);
00044
00045         string lzDecompressed;
00046         LZ77Codec::decompress(huffmanDecompressed, &lzDecompressed);
00047
00048         result.push_back(lzDecompressed);
00049         FileIO::write(outputFile, result);
00050     }
00051
00052     else {
00053         cout « "Wrong Command" « endl;
00054     }
00055 }
```

5.2 CommandHandler.h

```
00001 #ifndef COMPRESSOR_COMMANDHANDLER_H
00002 #define COMPRESSOR_COMMANDHANDLER_H
00003
00004 #include "Common.h"
00005
```

```
00010 class CommandHandler {
00011 public:
00012     CommandHandler(string inputFile, string outputFile, string command): inputFile(inputFile),
      outputFile(outputFile), command(command) {}
00013     void execute();
00014 private:
00015     string inputFile;
00016     string outputFile;
00017     string command;
00018 };
00019
00020 #endif //COMPRESSOR_COMMANDHANDLER_H
```

5.3 Common.h File Reference

Deflate project for FoCP uni subject.

```
#include <iostream>
#include <vector>
#include <string>
#include <map>
#include <queue>
#include <fstream>
#include <bitset>
#include <sstream>
```

5.3.1 Detailed Description

Deflate project for FoCP uni subject.

Author

Yassine Bendimerad (yb308985@student.polsl.pl)

Version

2.0

Date

2023-02-07

Copyright

Copyright (c) 2023

Definition in file Common.h.

5.4 Common.h

Go to the documentation of this file.

```
00001
00011 #ifndef COMPRESSOR_COMMON_H
00012 #define COMPRESSOR_COMMON_H
00013
00014 #include <iostream>
00015 #include <vector>
00016 #include <string>
00017 #include <map>
00018 #include <queue>
00019 #include <fstream>
00020 #include <bitset>
00021 #include <sstream>
00022
00023 using namespace std;
00024
00030 #endif //COMPRESSOR_COMMON_H
```

5.5 main.cpp

```
00001 #include "CommandHandler.h"
00002
00003 using namespace std;
00004
00010 int main(int argc, char *argv[]) {
00011     if (argc != 4) {
00012         cout « "Wrong number of arguments" « endl;
00013     }
00014     else {
00015         string command = argv[1];
00016         string inputFilename = argv[2];
00017         string outputFilename = argv[3];
00018
00019         CommandHandler *commandHandler = new CommandHandler(inputFilename, outputFilename, command);
00020         commandHandler->execute();
00021     }
00022     return 0;
00023 }
00024
```